MOHEX WINS HEX TOURNAMENT

Ryan Hayward[1]

Edmonton, Canada

1. THE TOURNAMENT

The 2011 Hex competition started on Sunday November 20 and finished on Monday November 21. Three programs competed: PANORAMEX by Fabien Teytaud, Tristan Cazenave, and Nicolas Jouandeau from France; WOLVE by Broderick Arneson, Ryan Hayward, and Philip Henderson from Canada; and MOHEX by Philip Henderson, Broderick Arneson, and Ryan Hayward. This year Aja Huang from Taiwan and Jakub Pawlewicz from Poland helped improve WOLVE and MOHEX.

PANORAMEX — named after Panoramix, the druid character from the Asterix and Obelix comic strip — uses the RAVE UCT formula (Gelly and Silver, 2007) with UCB exploration constant 0 and the save-bridge pattern in simulations. PANORAMEX ran on an 18-node cluster of 4-core machines, using root parallelization and majority vote to select each move. This yielded about 6×10^5 simulations per second.

WOLVE, the 2010 silver medallist (Arneson, Hayward, and Henderson, 2010)[2], uses truncated-width alpha-beta search, a Shannon-style electric circuit evaluation function with cell adjacencies augmented by virtual connections, and pruning of inferior cells. To save time, WOLVE uses a book built by caching 6-ply moves. This year Broderick Arneson added pondering and changed the search algorithm from fixed-ply to variable-ply with time-managed iterative deepening. WOLVE used 2 threads (one to select moves, one to solve) on a 4-core machine, reaching 6-ply on most moves.

MOHEX, the 2010 gold medallist (Arneson *et al.*, 2010), is a Monte Carlo tree search program built on the code base of FUEGO, the Go program developed by Martin Müller, Markus Enzenberger and others at the University of Alberta. FUEGO uses lock-free parallelization (Enzenberger and Müller, 2009), and backs up virtual losses for better parallelization. MOHEX computes virtual connections and inferior cells in UCT tree nodes visited at least 400 times. This year Arneson added pondering, Huang helped with tuning, and Pawlewicz improved the transposition table. On day 1 MOHEX ran on 24 cores, yielding about 1.4×10^5 simulations per second. On day 2, with the help of Timo Ewalds from Canada, MOHEX ran on 16 cores in the Amazon cloud.

MOHEX and WOLVE share an inferior cell engine, a virtual connection engine (modified this year by Pawlewicz) and a depth-first proof number search solver that runs on its own thread. SOLVER produces perfect play whenever it solves the position within the time allocated for a move.

	MOHEX	WOLVE	PANORAMEX	total	result
MOHEX		3-0	4-0	7-0	gold
WOLVE	0-3		4-0	4-3	silver
PANORAMEX	0-4	0-4		0-8	bronze

2. THE GAMES

For analysis, for each game we found the earliest position that SOLVER could solve in a short period of time (from a few minutes to an hour, depending on our interest). At each node in its search tree, SOLVER performs

[1] Department of Computing Science, University of Alberta, Canada. Email:hayward@ualberta.ca

[2] See also previous games competition reports (Melis and Hayward, 2003; Willemson and Björnsson, 2004; Hayward, 2006; Arneson, Hayward, and Henderson, 2008; Arneson, Hayward, and Henderson, 2009a; Arneson, Hayward, and Henderson, 2009b).

virtual-connection and inferior cell analysis. The latter allows many inferior moves to be pruned from the search; the former allows the computation of a *mustplay* region: any move outside this region is a loss for the player to move next. SOLVER has the same virtual-connection and inferior cell engines as MOHEX and WOLVE.

Round 1. Double round robin with 11 games — the scheduled 12th game was not needed.

Game 1. MOHEX-PANORAMEX 1-0. 1.B[a2] 2.W[f6] ... PANORAMEX plays solidly and MOHEX is losing by move 27. MOHEX sees the loss and plays to prolong the game, PANORAMEX blunders with 56.W[h10] — g10 and g9 both win, and MOHEX escapes.

Game 2. PANORAMEX-MOHEX 0-1. 1.B[a3] 2.W[swap] ... PANORAMEX is losing by move 17. By move 20, MOHEX knows that all but 17 cells are inferior.

Game 3. PANORAMEX-WOLVE 0-1. 1.B[a3] 2.W[swap] ... PANORAMEX is losing by move 29.

Game 4. WOLVE-PANORAMEX 1-0. 1.B[a2] 2.W[f6] ... PANORAMEX is losing by move 16.

Game 5. MOHEX-WOLVE 1-0. 1.B[a2] 2.W[swap] ... WOLVE is losing by move 38.

Game 6. WOLVE-MOHEX 0-1. 1.B[e2] 2.W[swap] ... WOLVE is losing by move 27.

Game 7. PANORAMEX-MOHEX 0-1. 1.B[a2] 2.W[e7] ... MOHEX's pre-search virtual-connection computation recognizes 35.B[e2] as a loss, and immediately replies with a winning move.

Game 8. MOHEX-PANORAMEX 1-0. 1.B[f2] 2.W[swap] ... Move 18.B[e11] might be a blunder: it is inferior to f10, and both MOHEX and WOLVE would play one of e5,j2,i3,h4. PANORAMEX is losing by move 22.

Game 9. WOLVE-PANORAMEX 1-0. 1.B[c1] 2.W[f6] ... PANORAMEX is losing by move 22. WOLVE knows it is winning by move 25.

Game 10. PANORAMEX-WOLVE 0-1. 1.B[c1] 2.W[e7] ... WOLVE's first three moves are from its cache-book. Black is losing by move 25.

Game 11. MOHEX-WOLVE 1-0. 1.B[a2] 2.W[swap] ... MOHEX is winning by move 18. The players know the outcome by move 22.

3. CONCLUSIONS

In Hex as in many other games, this year's champion uses Monte Carlo tree search. However, WOLVE uses only 2 threads (one to select moves, one to solve). In recent single-thread fixed-time tests (60 s/game and 300 s/game), WOLVE won 57% of the games against MOHEX. So parallelizing WOLVE's move selection might help.

PANORAMEX's debut performance was stronger than its record indicates. It played some strong openings and was unlucky not to win Game 1. Using virtual connections and/or inferior cells might help.

MOHEX's performance was weaker than its record indicates. Its opening play was sporadic. Adding prior knowledge before simulations might help.

Acknowledgements. We thank NSERC, iCORE, Martin Müller, Jonathan Schaeffer, and the UofA GAMES group for financial support; and the many people who have worked on previous versions of WOLVE, MOHEX, and SOLVER.

4. REFERENCES

Arneson, B., Hayward, R. B., and Henderson, P. (2008). Wolve Wins Hex Tournament. *ICGA*, Vol. 31, No. 4.

Arneson, B., Hayward, R. B., and Henderson, P. (2009a). MoHex Wins Hex Tournament. *ICGA*, Vol. 32, No. 2, pp. 114–116.

Arneson, B., Hayward, R. B., and Henderson, P. (2009b). Wolve 2008 Wins Hex Tournament. *ICGA*, Vol. 32, No. 1, pp. 49–53.

Arneson, B., Hayward, R. B., and Henderson, P. (2010). MoHex Wins Hex Tournament. *ICGA*, Vol. 33, No. 3, pp. 181–186.

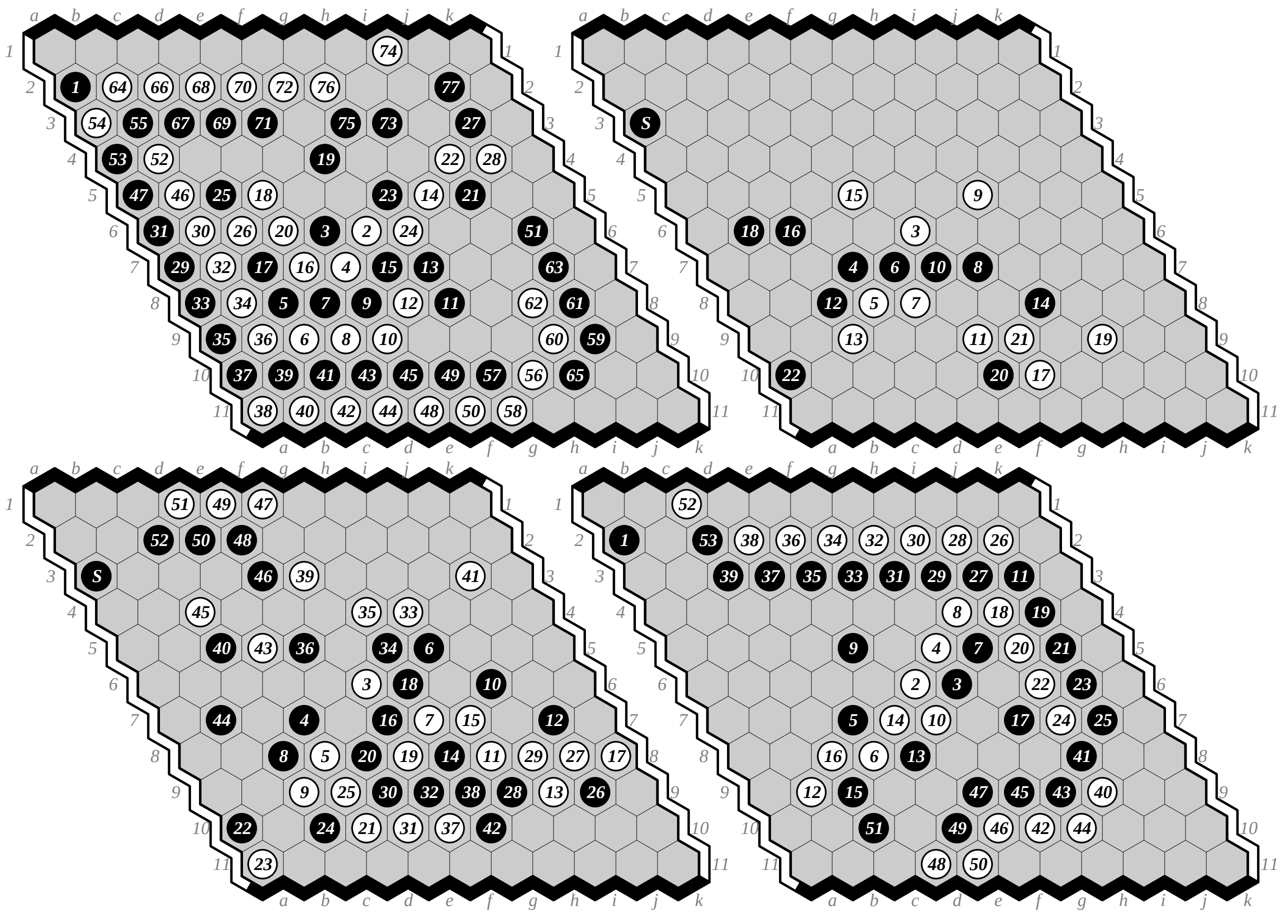
Enzenberger, M. and Müller, M. (2009). A Lock-free Multithreaded Monte-Carlo Tree Search. *Advances in Computer Games 12* (eds. H. J. van den Herik and P. Spronck), Vol. 666 of *Lecture Notes in Computer Science*, pp. 1–12. Springer.

Gelly, S. and Silver, D. (2007). Combining online and offline knowledge in UCT. *ICML '07: Proc. 24th international conference on machine learning*, pp. 273–280.

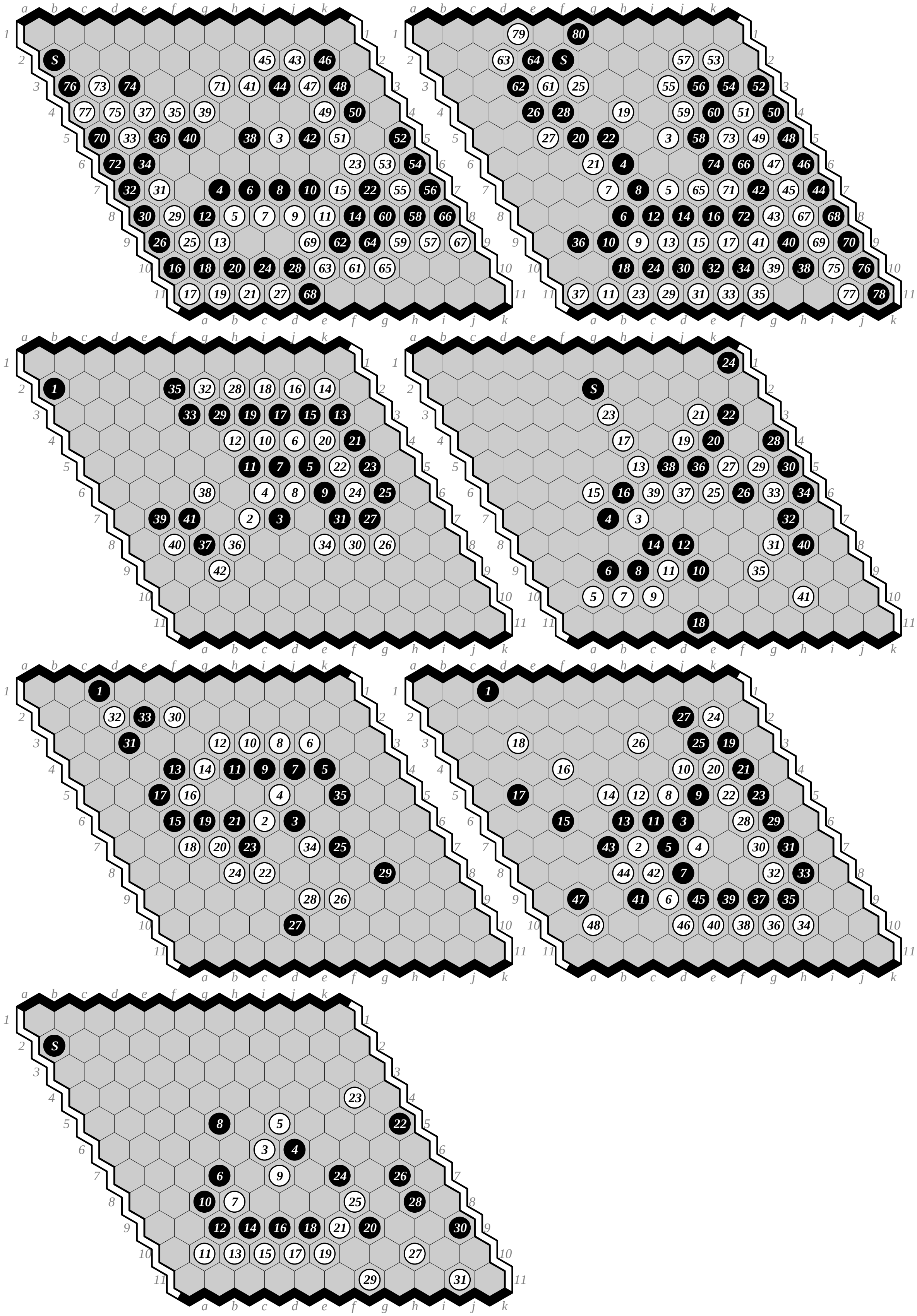
Hayward, R. B. (2006). Six Wins Hex Tournament. *ICGA*, Vol. 29, No. 3, pp. 163–165.

Melis, G. and Hayward, R. (2003). Six Wins Hex Tournament. *ICGA*, Vol. 26, No. 4, pp. 277–280.

Willemson, J. and Björnsson, Y. (2004). Six Wins Hex Tournament. *ICGA*, Vol. 27, No. 3, p. 180.



Games 1-4 (top to bottom). MoHex-Panoramex 1-0, Panoramex-MoHex 0-1, Panoramex-Wolve 0-1, Wolve-Panoramex 1-0. In some games the operator of the losing program resigned.



Games 5-11 (top to bottom). M-W 1-0, W-M 0-1, P-M 0-1, M-P 1-0, W-P 1-0, P-W 0-1, M-W 1-0.