1. THE TOURNAMENT

The 2013 Hex competition started on Monday August 12 and finished on Tuesday August 13. Three programs competed: \textsc{Jhinenox} by Jinno Masatoshi from Japan; \textsc{Ezo} by Kei Takada, Masaya Honjo, Makoto Mitsuhashi, and Masahito Yamamoto from Japan; \textsc{MoHex} 2.0 by Broderick Arneson, Aja Huang, Philip Henderson, Jakub Pawlewicz, and Ryan Hayward from Canada. A fourth program by Bruno Bouzy of France was unable to participate.

\textsc{Jhinenox} uses MCTS. \textsc{Jhinenox} ran on a 1.9GHz Core i7 laptop with 32 Gb memory. After Round 1 a bug was fixed and 17 dead- and captured-cell patterns were added to the program to prune moves in the tree.

\textsc{Ezo} uses 2-ply alpha-beta search with an evaluation function based on the theory of complex networks. The function combines a betweenness measure of the player’s two sides with a shortest path metric that accounts for the degree of important nodes. \textsc{Ezo} computes virtual connections but no edge templates. \textsc{Ezo} takes about 2 min./move in the early game, and switches to a Hexy-style evaluation when remaining time becomes short. \textsc{Ezo} ran 64 bit Ubuntu on a 1.9GHz Core i7 laptop with 128Gb SSD and 4Gb memory.

\textsc{MoHex} 2.0 (Huang et al., to appear 2014) is a revised version of \textsc{MoHex} (Arneson, Hayward, and Henderson, 2010b), the previous Olympiad gold medal winner (Hayward, 2012; Arneson, Hayward, and Henderson, 2010a; Arneson, Hayward, and Henderson, 2009). \textsc{MoHex} 2.0 is built on the code base of \textsc{Fuego} (Enzenberger et al., 2007–2012), the MCTS Go program developed by Martin Müller, Markus Enzenberger and others at the University of Alberta. \textsc{MoHex} 2.0 uses MCTS and size-12 patterns with learned weights to guide child selection in the tree, and for probabilistic simulations; it ran on a hyper-threaded 12-core machine (except for 2 games, when it ran on a 2-thread laptop). It uses at least one thread to run a solver, which produces perfect play if a position is solved within the allotted time. It used a book built by Broderick Arneson using progressive widening and Thomas Lincke’s method (Lincke, 2000). For each expanded node, one million simulations were run to find the best move; then that move was pruned from the list of candidates, and one million simulations were run to find the next best move, etc. The default number of children expanded was 4.

<table>
<thead>
<tr>
<th></th>
<th>MoHex</th>
<th>Ezo</th>
<th>Jhinenox</th>
<th>total</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoHex</td>
<td>4-0</td>
<td>4-0</td>
<td>8-0</td>
<td>gold</td>
<td></td>
</tr>
<tr>
<td>Ezo</td>
<td>0-4</td>
<td>3-1</td>
<td>3-5</td>
<td>silver</td>
<td></td>
</tr>
<tr>
<td>Jhinenox</td>
<td>0-4</td>
<td>1-3</td>
<td>1-7</td>
<td>bronze</td>
<td></td>
</tr>
</tbody>
</table>

2. THE GAMES

For post-match analysis, for each game we ran a parallel solver on 16 threads for a few minutes.

**Round 1.** Double round robin.

**Game 1.** \textsc{Jhinenox-MoHex} 0-1. 1.B[i1] 2.W[f6] ... Black play is hampered by a bug. Notice how a Hex game can end quickly if a player fails to block the opponent’s most direct threats. White is winning by move 8.

**Game 2.** \textsc{Jhinenox-Ezo} 0-1. 1.B[h9] 2.W[swap] ... Black is winning by move 8.

**Game 4.** MoHEX-JHINENOX 1-0. 1.B[k2] 2.W[swap] . . . White (after swap) is losing by move 8.

**Game 5.** EZO-JHINENOX 1-0. 1.B[i8] 2.W[g5] . . . White is losing by move 8.


**Round 2.** Double round robin.


**Game 9.** MoHEX-EZO 1-0. 1.B[a2] 2.W[swap] . . . White (after swap) is winning by move 15.

**Game 10.** MoHEX-JHINENOX 1-0. 1.B[a2] 2.W[e7] . . . The MoHEX operator could not connect with the 12-core machine in Canada, and so ran a 2-thread laptop version with no opening book. 3.B[f7] — one of three moves likely MoHEX moves under these conditions — leads to a game where, after each White move, Black has only one strong response. The result is a curious game in which each move is close the opponent's previous move. White is losing by move 36. The white moves in row 1 (k1 j1 . . .) may not be blunders, but the resulting ladder (k1 j2 j1 i2 . . .) makes it easier to solve the position. The solver indicates white is probably losing by move 34, but this position is much harder to prove.


3. **CONCLUSIONS**

EZO and JHINENOX are algorithmically different — connectivity-measuring alpha-beta search versus MCTS — but seemed evenly matched by Round 2. Neither put MoHEX in a troubling position, but each played on a single thread, a disadvantage to the 12 threads used by MoHEX.

**Acknowledgements.** We thank NSERC for financial support and Martin Müller for loaning his 12-core machine.

4. **REFERENCES**


In some games the operator of the losing program resigned.

**Games 1-6.** JHINENOX-MoHEX 0-1, JHINENOX-EZO 0-1, MoHEX-EZO 1-0, MoHEX-JHINENOX 1-0, EZO-JHINENOX 1-0, EZO-MoHEX 0-1. In some games the operator of the losing program resigned.
Games 7-12. JHINEX-MoHEX 0-1, JHINEX-EZO 1-0, MoHEX-EZO 1-0, MoHEX-JHINEX 1-0, EZO-JHINEX 1-0, EZO-MoHEX 0-1.