

2018 New Taipei City Computer Olympiad Report

Hex 2018: MoHex3HNN over DeepEzo

Chao Gao, Kei Takada and Ryan Hayward*

Department of Computing Science, University of Alberta, Canada

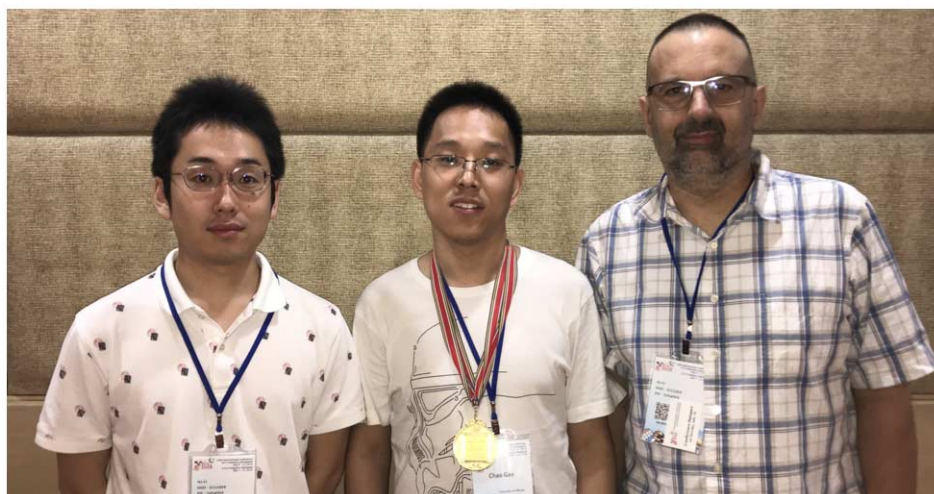


Fig. 1. From left, Kei Takada, Chao Gao, Ryan Hayward.

1. THE TOURNAMENTS

There were two Hex tournaments at the 2018 Computer Olympiad in New Taipei City: 11×11 – proposed by Piet Hein in 1942 – and 13×13 .¹

Visa problems unfortunately prevented Chinese teams from competing, leaving two contestants for each tournament: DEEPEZO from Japan, by Kei Takada, supervised by Masahito Yamamoto; and MOHEX-3HNN from Canada, by Chao Gao, supervised by Ryan Hayward and Martin Müller and assisted by Jakub Pawlewicz, Ashley Herman, Joseph Meleshko, Jiahao Li, Paul Banse, Siqi Yan, Wai Yi Low and Xutong Zhao. Gao and Pawlewicz developed a self-play game procedure, used to select opening moves.

*Corresponding author. E-mail: hayward@ualberta.ca.

¹Hayward et al. (2018) has .sgf game records and source files for this report. Arneson (2014) has an SGF (Smart Game Format) viewer. SGF was developed by Kierulf et al. (1987).

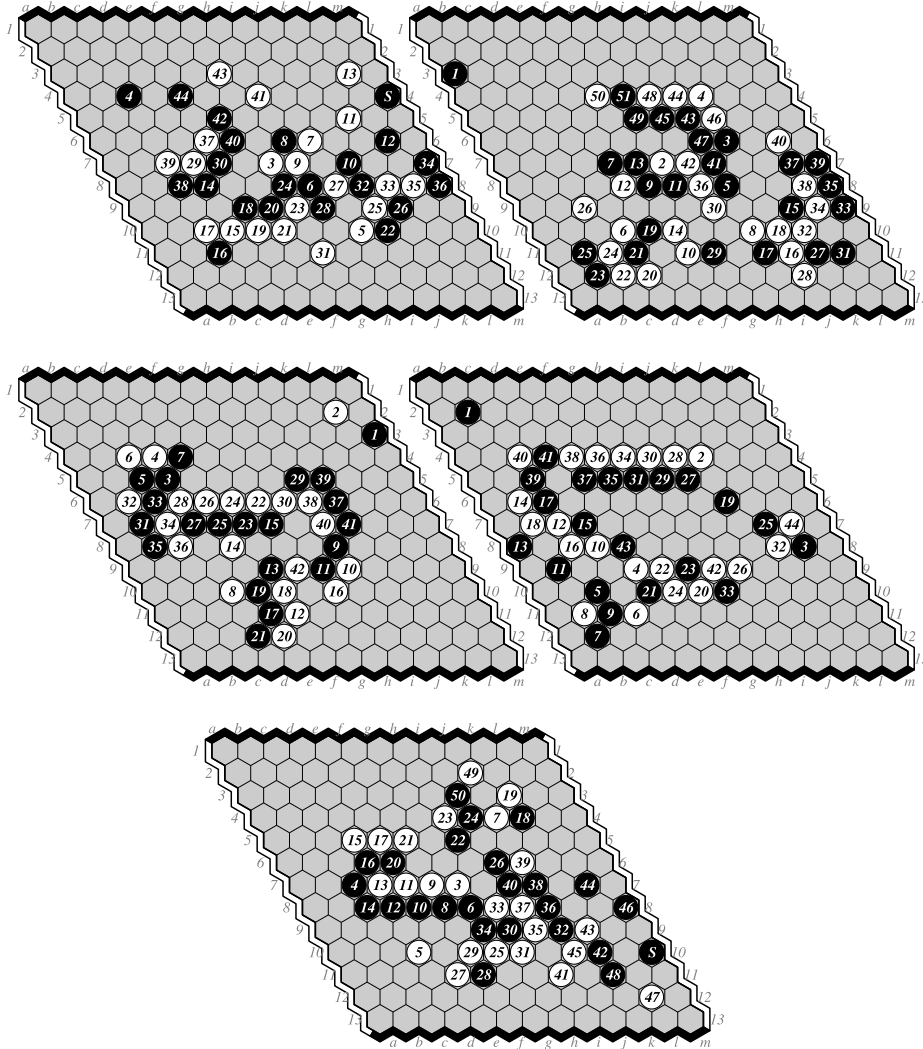


Fig. 2. 13×13 Games 1–5. E–M 0–1, M–E 1–0, E–M 0–1, M–E 1–0, E–M 0–1. S indicates that move 2 was swap.

MOHEX-3HNN is an AlphaGo-style neural net program that shares code with MoHex2.0 by Broderick Arneson, Philip Henderson, Aja Huang, Jakub Pawlewicz, Noah Weninger, Kenny Young and Ryan Hayward (2013). MoHex versions have won the previous eight Olympiad Hex competitions (Hayward and Weninger, 2017). MoHex2.0 is an MCTS program that uses the Benzene Hex framework built on the code base of FUEGO (Enzenberger et al., 2007–2012).

MOHEX-3HNN, the successor of MOHEX-CNN, uses a three-head convolutional neural net (CNN) with 128 filters per layer (Gao et al., 2018) trained on 400 000 self-play games. At each new node of the Monte Carlo search tree, the three-headed neural net is called and returns policy, state, and action values. MOHEX-3HNN ran remotely on a machine with four CPU cores and one GPU. Benzene’s solver was not used: all threads were used for tree computation.

DEEPEZO ran remotely on a machine with two CPUs (one thread for search, one for Benzene’s solver) and one GPU. In some games the internet connection dropped and computation restarted on a laptop.

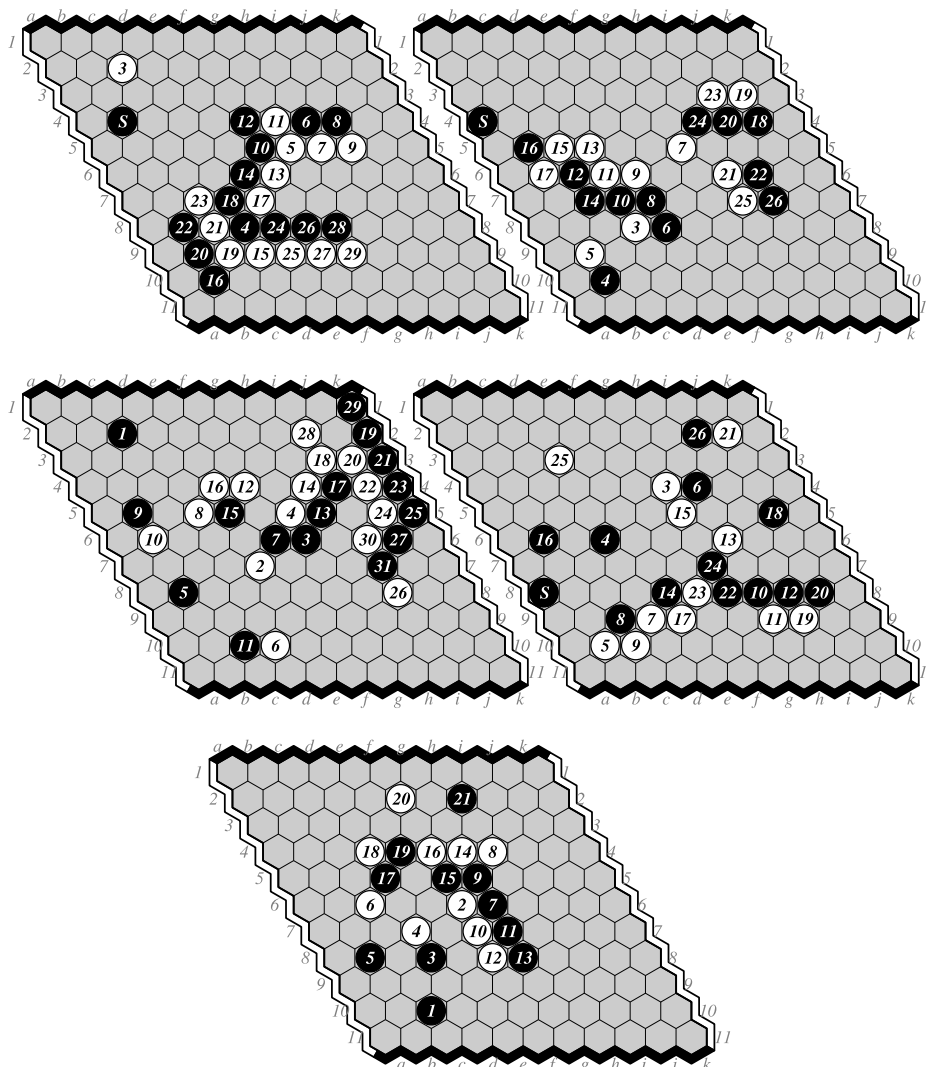


Fig. 3. 11×11 Games 1–5. M–E 1–0, E–M 0–1, M–E 1–0, E–M 0–1, M–E 1–0.

DEEPEZO, written on the Benzene framework, uses iterative deepening depth-first search and two CNN functions. One is the evaluation function, which evaluates the current position. The other is the policy function, which decides which moves to search next. Moves whose value exceed a threshold are explored; the rest are pruned. Starting with random weights, the two functions are built with reinforcement learning. Learning games are generated by self-play with depth-one minimax search.

On a 13×13 board, in a 676 game test (all 1-move openings, 2 games as 1st player, 2 games as 2nd player), with DEEPEZO and MoHex2.0 allowed 30 s/move and EZO-CNN (2017) a 4-ply search, DEEPEZO wins about .80 and .85 of its games respectively against MoHex2.0 and EZO-CNN (2017).

The DEEPEZO model for this competition had more learning than the tested model, so might be even stronger. 2.6 million learning games were played.

Tournament results. Each tournament was best of 8 games, with 30 min/game per player. The 13×13 and 11×11 tournaments were played on July 8 and 9 respectively. Each game ended by resignation as soon as an opponent win was detected. MOHEX-3HNN won both tournaments 5–0. See Figs 2 and 3.

2. CONCLUSIONS

Using machine learning ideas inspired by AlphaGo, both programs improved significantly this year. MOHEX-3HNN is especially strong on 11×11 : Move 5 in Game 5 reminds us of Move 37 in Game 2 of the AlphaGo-Lee Sedol match: we have never seen this move before, but by the end of the game its utility is clear. Hopefully we will see similar moves next year on the 13×13 and 19×19 boards.

ACKNOWLEDGEMENT

We thank the NSERC Discovery Grant Program for research funding.

REFERENCES

- Arneson, B. (2014). Hexgui: An sgf hex viewer. <http://github.com/ryanbhayward/hexgui>.
- Enzenberger, M., Müller, M., Arneson, B., Segal, R., Xie, F. & Huang, A. (2007–2012). Fuego. <https://sourceforge.net/p/fuego/code/HEAD/tree/>.
- Gao, C., Müller, M. & Hayward, R. (2018). Three-head neural network architecture for Monte Carlo tree search. In *Proc. 27th IJCAI*. <https://www.ijcai.org/proceedings/2018/0523.pdf>.
- Hayward, R., Gao, C. & Takada, K. (2018). Source files for this report. <http://github.com/ryanbhayward/icga-olympiad-hex>.
- Hayward, R. & Weninger, N. (2017). Hex 2017: MoHex wins the 11×11 and 13×13 tournaments. *ICGA*, 39(3), 222–227. <https://webdocs.cs.ualberta.ca/~hayward/papers/leiden17.pdf>.
- Huang, S., Arneson, B., Hayward, R.B., Müller, M. & Pawlewicz, J. (2013). Mohex 2.0: A pattern-based MCTS hex player. In H.J. van den Herik, H. Iida and A. Plaat (Eds.), *Computers and Games – 8th International Conference, CG 2013*. Lecture Notes in Computer Science Yokohama, Japan, August 13–15, 2013 (Vol. 8427, pp. 60–71). Springer. Revised Selected Papers.
- Kierulf, A., Müller, M. & Hollosi, A. (1987). Smart game format. https://en.wikipedia.org/wiki/Smart_Game_Format.