ICGA Journal 43 (2021) 132–134
DOI 10.3233/ICG-210192
IOS Press

DESCENT wins five gold medals at the Computer Olympiad

Quentin Cohen-Solal * and Tristan Cazenave
LAMSADE, University Paris-Dauphine, PSL, CNRS, France

Descent (Cohen-Solal, 2020; Cohen-Solal and Cazenave, 2021) is a zero knowledge Deep Reinforcement Learning algorithm that has learned to play many games. It won five gold medals at the 2020 Computer Olympiad.

Unlike AlphaZero-like algorithms (Silver et al., 2018), the Descent framework uses a variant of Unbounded Minimax (Korf and Chickering, 1996), instead of Monte Carlo Tree Search, to construct the partial game tree used to determine the best action to play and to collect data for learning. During training, at each move, the best sequences of moves are iteratively extended until terminal states. During evaluations, the safest action is chosen (after that the best sequences of moves are iteratively extended each until a leaf state is reached). Moreover, it also does not use a policy network, only a value network. The actions therefore do not need to be encoded. Unlike the AlphaZero paradigm, with Descent all data generated during the searches to determine the best actions to play is used for learning. As a result, much more data is generated per game, and thus the training is done more quickly and does not require a (massive) parallelization to give good results (contrary to AlphaZero). It can use end-of-game heuristic evaluation to improve its level of play faster, such as game score or game length (in order to win quickly and lose slowly).

Five gold medals were won by our programs based on these algorithms for the following games: Othello 10 × 10, Breakthrough, Surakarta, Amazons, and Clobber. A silver medal was won by our programs at Othello 8 × 8.

The other competitors for each game were:

- Breakthrough: DaSoJai (author: Wei-Lin Wu and Shun-Shii Lin) and Polygames (Facebook NDHU). Polygames (Cazenave et al., 2020) is a reimplementation of AlphaZero.
- Amazons: 8QP (Johan de Koning) and SherlockGo (Liang Shuang, Liang Tailin, Wang Jilong, Li Xiaorui, and Zhou Ke).
- Othello 10 × 10: Polygames (Facebook NDHU) and Persona (Surag Nair, Nai-Yuan Chang, Shun-Shii Lin).
- Othello 8 × 8: Polygames (Facebook NDHU) and Maverick (Yen-Chi Chen and Shun-Shii Lin).
- Clobber: Pan.exe (Johan de Koning), Klopper (Johannes Schwagereit), and Calpurnia (Christian Jans). Calpurnia uses an AlphaZero-like approach and an endgame solver based on the Combinatorial Game Theory.
- Surakarta: CZF_Surakarta (Liang-Fu Liu), FuChou (Jia-Fong Yeh, Yen-Chi Chen, Shun-Shii Lin), and VSSurakarta (Zhang Yunpeng, Li Wei, Zhang Yuxuan, Zhang Pei, and Zhou Ke). CZF_Surakarta is trained based on AlphaZero.

Some details of the matches performed by the programs created by the Descent framework are described in Table 1.

*Corresponding author. E-mail: quentin.cohen-solal@dauphine.psl.eu.

Table 1

Details of matches played during the first phase et second phase (playoffs) of the 2020 Computer Olympiad on Ludii: Ludii game ID, Ludii version, game name, first player program, second player program, and the winner (in bold: program based on the Descent framework)

ID	Version	Game	First player	Second player	Winner
303	1.1.5	Havannah 8	Polygames	**Doombot-8**	Polygames
310	1.1.5	Havannah 8	**Doombot-8**	Polygames	Polygames
333	1.1.5	Breakthrough	DaSoJai	**R2D2**	**R2D2**
336	1.1.5	Breakthrough	**R2D2**	DaSoJai	**R2D2**
348	1.1.5	Clobber	**Hercule**	Klopper	**Hercule**
350	1.1.5	Clobber	Klopper	**Hercule**	**Hercule**
360	1.1.5	Clobber	Pan.exe	**Hercule**	Pan.exe
361	1.1.5	Clobber	**Hercule**	Pan.exe	Pan.exe
391	1.1.5	Othello 8	Maverick	**Réplicateur #8**	Maverick
392	1.1.5	Othello 8	**Réplicateur #8**	Maverick	Maverick
414	1.1.5	Breakthrough	Polygames	**R2D2**	**R2D2**
417	1.1.5	Breakthrough	**R2D2**	Polygames	**R2D2**
430	1.1.7	Othello 8	Polygames	**Réplicateur #8**	**Réplicateur #8**
435	1.1.7	Othello 8	**Réplicateur #8**	Polygames	**Réplicateur #8**
436	1.1.7	Othello 10	Polygames	**Réplicateur #10**	**Réplicateur #10**
439	1.1.7	Othello 10	**Réplicateur #10**	Polygames	**Réplicateur #10**
443	1.1.7	Surakarta	FuChou	**Athénan**	**Athénan**
445	1.1.7	Surakarta	**Athénan**	FuChou	**Athénan**
447	1.1.7	Othello 10	Persona	**Réplicateur #10**	**Réplicateur #10**
448	1.1.7	Othello 10	**Réplicateur #10**	Persona	**Réplicateur #10**
452	1.1.5	Amazons	8QP	**Thésée**	**Thésée**
453	1.1.5	Amazons	**Thésée**	8QP	**Thésée**
462	1.1.7	Surakarta	**Athénan**	VSSurakarta	**Athénan**
500	1.1.7	Surakarta	VSSurakarta	**Athénan**	**Athénan**
475	1.1.7	Surakarta	CZF_Surakarta	**Athénan**	CZF_Surakarta
477	1.1.7	Surakarta	**Athénan**	CZF_Surakarta	**Athénan**
480	1.1.7	Surakarta	FuChou	**Athénan**	draw
487	1.1.7	Clobber	Calpurnia	**Hercule**	**Hercule**
488	1.1.7	Clobber	**Hercule**	Calpurnia	**Hercule**
507	1.1.7	Amazons	SherlockGo	**Thésée**	**Thésée**
508	1.1.7	Amazons	**Thésée**	SherlockGo	**Thésée**
517	1.1.8	Hex 13	**Ultron-13**	Polygames	Polygames
529	1.1.8	Hex 13	Polygames	**Ultron-13**	Polygames
532	1.1.8	Hex 19	**Ultron-19**	Polygames	**Ultron-19**
538	1.1.8	Hex 19	Polygames	**Ultron-19**	Polygames
541	1.1.9	Clobber	Klopper	**Hercule**	**Hercule**
542	1.1.9	Clobber	**Hercule**	Klopper	**Hercule**
548	1.1.9	Clobber	Pan.exe	**Hercule**	**Hercule**
549	1.1.9	Clobber	**Hercule**	Pan.exe	Pan.exe
554	1.1.9	Clobber	Pan.exe	**Hercule**	**Hercule**
560	1.1.9	Hex 19	Polygames	**Ultron-19**	Polygames
562	1.1.9	Hex 19	**Ultron-19**	Polygames	Polygames

The Descent framework could struggle with connection and alignment games. It notably lost against Polygames at Hex and Havannah. However, Polygames used much more computing power for their training (100 GPU against 1 GPU) and uses very deep networks, so it is possible that this is the cause of this difference.

Descent thus constitutes an alternative to AlphaZero, in particular under resource constraints.

REFERENCES

Cazenave, T., Chen, Y.-C., Chen, G.-W., Chen, S.-Y., Chiu, X.-D., Dehos, J., Elsa, M., Gong, Q., Hu, H., Khalidov, V., Cheng-Ling, L., Lin, H.-I., Lin, Y.-J., Martinet, X., Mella, V., Rapin, J., Roziere, B., Synnaeve, G., Teytaud, F., Teytaud, O., Ye, S.-C., Ye, Y.-J., Yen, S.-J. & Zagoruyko, S. (2020). Polygames: Improved zero learning. *ICGA Journal*, *42*(4), 244–256. doi:10.3233/ICG-200157.

Cohen-Solal, Q. (2020). Learning to Play Two-Player Perfect-Information Games without Knowledge. arXiv preprint. 2008.01188.

Cohen-Solal, Q. & Cazenave, T. (2021). Minimax strikes back. In *Reinforcement Learning in Games at AAAI*.

Korf, R.E. & Chickering, D.M. (1996). Best-first minimax search. *Artificial intelligence*, *84*(1–2), 299–337. doi:10.1016/0004-3702(95)00096-8.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, *362*(6419), 1140–1144. doi:10.1126/science.aar6404.