# Guest Editorial: Special Issue on Evolutionary Computation for Games

THE application of evolutionary computation (EC) [1] is widespread because the core method is very general. If you have a way to represent candidate solutions, modify candidate solutions and evaluate candidate solutions, then you can apply EC. No complex formalism or theoretical framework is needed, which makes EC easy to apply, and lowers the barrier for entry by new researchers. This generality is particularly useful for games, because there are many different aspects of games that can benefit from searching for better solutions.

What is more, EC is also a powerful method with many benefits over traditional optimization methods. Because evolution is population based, multiple solutions to a problem can be produced rather than just one. Having a population of solutions is extremely useful in multimodal problems, multiobjective problems, problems with nonstationary fitness landscapes, and situations where a diversity of different solutions is desired. All of these situations are common in games: good games typically allow agents to solve problems in a variety of different ways (multimodal), different players/agents may have preferences for different objectives (multiobjective), the value of a given evolved artifact or agent strategy may change in response to player actions in an online game (non-stationary), and many games focus on collecting a wide range of different novel artifacts (diversity).

Studies exploring various aspects of games with EC are abundant. There have been numerous examples of evolution applied to the discovery of agent behavior in games, from board games, such as Checkers [2] and Chess [3], to classic arcade games, such as *Ms. Pac-Man* [4], *Super Mario Bros*. [5], and various *Atari* games [6], all the way up to modern games including first-person shooters [7] and car racing [8]. Evolution has also proven effective as a form of procedural content generation (PCG) [9], having been used to generate maps and levels [10], artistic content [11], and weapons [12]. Evolution also blends well with many other techniques that are relevant in the playing and designing of games [13], [14].

Machine learning, particularly with deep neural networks, has dominated many areas of AI research in recent years. Generative adversarial networks and variational autoencoders provide new ways to induce a latent space of possibilities that give rise to novel artistic designs, and these vector spaces can be searched by EC. In the realm of agent control, many successful evolutionary approaches are actually a combination of evolution with some other techniques, such as game-tree search or neural networks. The flexibility of evolution keeps it relevant as other techniques continue to advance, since there will always be better ways to parameterize some aspect of the process that drives any other successful technique.

This Special Issue on applications of EC to games demonstrates several ways in which evolution can push boundaries and explore new areas of what is possible in the realm of games research, with a focus on game-playing, automatic agent parameter tuning, automatic game testing, and procedural content generation. A total of eight papers were accepted for this Special Issue of IEEE TRANSACTIONS ON GAMES, each of which underwent a rigorous process of peer review and revision before final acceptance for publication. The papers can be organized under the following broad topics.

## I. Rolling Horizon Evolutionary Algorithm

Two papers apply variants of the rolling horizon evolutionary algorithm (RHEA) to play games. The RHEA evolves a population of action sequences that can be performed in a finite number of future time steps and applies the rolling horizon strategy widely used in optimal control. It has shown to be competitive in playing single- and two-player games [15].

When applying RHEA to two-player games, an individual is often evaluated by playing against a search-based or random agent. In [A1], Tang et al. evaluate the evolved action sequences by playing against a well-trained learning-based opponent model. Their enhanced RHEA agent combined with a policy-gradient-based opponent model won first place in the 2020 IEEE Conference on Games edition of the Fighting Game AI Competition.[1]

In [A2], Bravi and Lucas compare variants of RHEA to other approaches. Specifically, a new method of handling games with sparse reward signals is proposed, which assigns values to events occurring in the game. This approach differs from and is demonstrated to be superior to heuristics and state-value functions.

## II. Agent Control

While RHEA can evolve action sequences when a forward model is available, EC is also frequently applied to the configuration of agents responding directly to the environment. In [A3],

Digital Object Identifier 10.1109/TG.2022.3225730

[1][Online]. Available: https://www.ice.ci.ritsumei.ac.jp/~ftgaic/

Dockhorn et al. optimize robust controllers for agents navigating a variety of different environments. They propose a real-valued encoding for parameters in a context-steering agent and search for optimal parameters using an evolutionary algorithm. Three fitness functions are designed to represent three different goals of the agents, and direct the search for desired configurations. The proposed approach is shown to be able to find robust configuration across different scenes of the same environment.

## III. Procedural-Level Generation

Two papers deal with PCG for level creation. Use of procedurally generated levels for commercial games has a long history, going back to the game *Rogue* and extending into the present with games such as *Diablo* and *No Man's Sky*. In academic games research, two recent approaches to PCG that have received much focus are Wave Function Collapse (WFC) and generative adversarial networks (GANs). The two papers in this Special Issue extend work in these two areas by combining these techniques with evolution.

EC techniques can direct the search in level space and generate levels of particular features or with particular aims. However, they do not guarantee the feasibility or playability of generated levels. In [A4], Bailly and Levieux generate 3-D levels with a genetic algorithm in which WFC is used as a repair operator to generate playable levels. This work combines WFC's ability to generate playable levels with evolution's ability to discover better levels.

In [A5], Schrum et al. evolve compositional pattern producing networks (CPPNs) to generate latent vectors for pre-trained GANs that produce level segments that are combined into cohesive levels for *Super Mario Bros.* and *The Legend of Zelda*. The added benefit from CPPNs is their ability to encode large-scale patterns that incorporate symmetry and repetition into the global-level design. However, to encourage more low-level variation in the levels, the indirect encoding provided by CPPNs can transition into a direct encoding that evolves the latent vectors individually. By evolving this representation using the quality diversity algorithm MAP-Elites [16], a wide variety of levels is discovered.

## IV. Automated Game Design

Many aspects of the game design process can benefit from automation via EC. Games are complex systems full of many fine-tuned parameters that need to be balanced to suit the needs of various players, and evolution can help in many ways.

In [A6], Rogers et al. performed a user study to assess evolution's ability to craft complexity graphs modeling in-game economies. The paper contributes methods for defining the complexity of a game economy in a city-building game, and explores how human users perceive complexity in such games.

In [A7], Gong et al. showed a deep neural network that provides the fitness function for an evolutionary algorithm creating teams for the game *Romance of the Three Kingdoms: Strategy Edition*. The method could be used to test similar games before launch, or to guide players with recommendations when creating their own teams.

Finally, Albaghajati and Ahmed [A8] present a survey of various automated testing approaches for video games. Their analysis of the literature ends up comparing several approaches, including those based on evolution.

## V. Conclusion

Different families of evolutionary algorithms have been broadly used in playing games and designing games. Papers in this Special Issue show how EC techniques can be used (alone or with other techniques) to play games and design games, including procedural-level generation, game balancing, and game testing. Although EC is capable of evolving better and more diverse solutions (e.g., action sequences, content, and latent vectors), there are still many challenges worthy of further study. Because evolution depends on the use of a population, many evaluations must be performed, which can be quite time consuming. In addition, if the wrong encoding or fitness function is chosen, even a simple problem can become practically unsolvable. Despite the chance for missteps, researchers are already addressing these challenges. Evolution can easily scale to take advantage of parallel evaluation, and surrogate fitness models can reduce reliance on expensive evaluation functions, as done by a paper in this very issue [A7]. Similarly, many novel solution encodings are being applied by the EC community, such as the CPPNs used in this issue [A5]. These examples demonstrate the bright future that EC has, particularly in the area of games research.

Jacob Schrum, *Guest Editor*
Department of Mathematics and Computer Science
Southwestern University
Georgetown, TX 78626 USA

Jialin Liu, *Guest Editor*
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen 518055, China

Cameron Browne, *Guest Editor*
Department of Data Science and Knowledge Engineering
Maastricht University
6229 EN, Maastricht, The Netherlands

Anikó Ekárt, *Guest Editor*
Department of Computer Science
Aston University
B4 7ET Birmingham, U.K.

Marcus Gallagher, *Guest Editor*
School of Information Technology and Electrical
Engineering
University of Queensland
Brisbane, Queensland, 4072, Australia

## Appendix: Related Articles

[A1] Z. Tang, Y. Zhu, D. Zhao, and S. M. Lucas, "Enhanced rolling horizon evolution algorithm with opponent model learning: Results for the fighting game AI competition," *IEEE Trans. Games*, vol. 15, no. 1, pp. 5–15, Mar. 2023, doi: 10.1109/TG.2020.3022698.

[A2] I. Bravi and S. Lucas, "Rinascimento: Playing *Splendor*-like games with event-value functions," *IEEE Trans. Games*, vol. 15, no. 1, pp. 16–25, Mar. 2023, doi: 10.1109/TG.2022.3171908.

[A3] A. Dockhorn, M. Kirst, S. Mostaghim, M. Wieczorek, and H. Zille, "Evolutionary algorithm for parameter optimization of context steering agents," *IEEE Trans. Games*, vol. 15, no. 1, pp. 26–35, Mar. 2023, doi: 10.1109/TG.2022.3157247.

[A4] R. Bailly and G. Levieux, "Genetic-WFC: Extending wave function collapse with genetic search," *IEEE Trans. Games*, vol. 15, no. 1, pp. 36–45, Mar. 2023, doi: 10.1109/TG.2022.3192930.

[A5] J. Schrum, B. Capps, K. Steckel, V. Volz, and S. Risi, "Hybrid encoding for generating large scale game level patterns with local variations," *IEEE Trans. Games*, vol. 15, no. 1, pp. 46–55, Mar. 2023, doi: 10.1109/TG.2022.3170730.

[A6] K. Rogers, V. Le Claire, J. Frommel, R. Mandryk, and L. E. Nacke, "Using evolutionary algorithms to target complexity levels in game economies," *IEEE Trans. Games*, vol. 15, no. 1, pp. 56–66, Mar. 2023, doi: 10.1109/TG.2023.3238163.

[A7] Y.-J. Gong et al., "Automated team assembly in mobile games: A data-driven evolutionary approach using a deep learning surrogate," *IEEE Trans. Games*, vol. 15, no. 1, pp. 67–80, Mar. 2023, doi: 10.1109/TG.2022.3145886.

[A8] A. M. Albaghajati and M. Ahmed, "Video game automated testing approaches: An assessment framework," *IEEE Trans. Games*, vol. 15, no. 1, pp. 81–94, Mar. 2023, doi: 10.1109/TG.2020.3032796.

## References


[1] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, 1st ed. Bristol, U.K.: IOP Publishing Ltd., 1997.

[2] D. B. Fogel, *Blondie24: Playing at the Edge of AI*. Burlington, MA, USA: Morgan Kaufmann, 2002.

[3] O. E. David, H. J. van den Herik, M. Koppel, and N. S. Netanyahu, "Genetic algorithms for evolving computer chess programs," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 779–789, Oct. 2014.

[4] J. Schrum and R. Miikkulainen, "Discovering multimodal behavior in Ms. Pac-Man through evolution of modular neural networks," *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 1, pp. 67–81, Mar. 2016.

[5] J. Togelius, S. Karakovskiy, J. Koutnik, and J. Schmidhuber, "Super Mario evolution," in *Proc. IEEE Symp. Comput. Intell. Games*, 2009, pp. 156–161.

[6] D. G. Wilson, S. Cussat-Blanc, H. Luga, and J. F. Miller, "Evolving simple programs for playing Atari games," in *Proc. Genet. Evol. Comput. Conf.*, 2018, pp. 229–236.

[7] A. Friedman and J. Schrum, "Desirable behaviors for companion bots in first-person shooters," in *Proc. IEEE Conf. Games*, 2019, pp. 1–8.

[8] S. Nallaperuma, F. Neumann, M. R. Bonyadi, and Z. Michalewicz, "EVOR: An online evolutionary algorithm for car racing games," in *Proc. Gen. Evol. Comput.*, 2014, pp. 317–324.

[9] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 3, pp. 172–186, Sep. 2011.

[10] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. M. Smith, and S. Risi, "Evolving Mario levels in the latent space of a deep convolutional generative adversarial network," in *Proc. Genet. Evol. Comput. Conf.*, 2018, pp. 221–228.

[11] S. Risi, J. Lehman, D. B. D'Ambrosio, R. Hall, and K. O. Stanley, "Petalz: Search-based procedural content generation for the casual gamer," *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 3, pp. 244–255, Sep. 2016.

[12] E. J. Hastings, R. K. Guha, and K. O. Stanley, "Automatic content generation in the galactic arms race video game," *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 4, pp. 245–263, Dec. 2009.

[13] A. Summerville et al., "Procedural content generation via machine learning (PCGML)," *IEEE Trans. Games*, vol. 10, no. 3, pp. 257–270, Sep. 2018.

[14] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius, "Deep learning for procedural content generation," *Neural Comput. Appl.*, vol. 33, no. 1, pp. 19–37, 2021.

[15] D. Perez-Liebana, J. Liu, A. Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, "General video game AI: A multitrack framework for evaluating agents, games, and content generation algorithms," *IEEE Trans. Games*, vol. 11, no. 3, pp. 195–214, Sep. 2019.

[16] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," 2015, *arXiv:1504.04909*.


**Jacob Schrum** (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from The University of Texas at Austin, Austin, TX, USA, in 2009 and 2014, respectively.

He is currently an Associate Professor of computer science with Southwestern University, Georgetown, TX, USA, and the Current Chair of the Department of Mathematics and Computer Science. His research interests include evolutionary computation, quality diversity, deep neural networks, neuroevolution, and procedural content generation.

Dr. Schrum is an Associate Editor for IEEE Transactions on Games and a Member of the Games Technical Committee of the IEEE Computational Intelligence Society. He was the recipient of the 2K BotPrize Competition to design Human-Like Bots in 2012 with collaborators from the University of Texas.

**Jialin Liu** (Senior Member, IEEE) received the B.Sc. degree in optical & electronic information from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2010, the Diplôme d'Ingénieur degree in computer science from the Polytech'Paris-Sud, Gif-sur-Yvette, France, in 2012, the M.Sc. degree in bioinformatics and biostatistics from the École Polytechnique and Université Paris-Sud, Gif-sur-Yvette, France, in 2013, and the Ph.D. degree in computer science from Université Paris-Saclay, Gif-sur-Yvette, France, in 2016.

She is currently a Tenure-Track Assistant Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Her research interests include AI in games, fair machine learning, robust optimization, and evolutionary computation and its applications.

Dr. Liu is an Associate Editor for IEEE TRANSACTIONS ON GAMES and was the Chair of the Games Technical Committee of the IEEE Computational Intelligence Society (2020–2021).

**Cameron Browne** received the Ph.D. degree in artificial intelligence from the Queensland University of Technology, Brisbane, QLD, Australia, in 2009, producing the world's first published computer-generated games.

He is currently an Associate Professor with the Department of Data Science and Knowledge Engineering, Maastricht University, Maastricht, The Netherlands, where he is currently running the European Research Council funded Digital Ludeme Project. He has authored or coauthored books, including *Hex Strategy, Connection Games*, and *Evolutionary Game Design*.

Dr. Browne is the Section Editor of IEEE TRANSACTIONS ON GAMES and *International Computer Games Association* (ICGA) journal, and was the Founder and Editor-in-Chief of the *Game & Puzzle Design* journal. He was the recipient of 2012 GECCO "Humies" Award for human-competitive results in evolutionary computation.

**Anikó Ekárt** received the Ph.D. degree in informatics from Eötvös Loránd University, Budapest, Hungary, in 2001.

She is currently a Professor of artificial intelligence with Aston University, Birmingham, U.K. Her research interests include evolutionary computation, genetic programming, and neuroevolution and their application to art, health, engineering, and transport problems.

Dr. Ekárt is a Member of the editorial board of the *Evolutionary Computation* and *Genetic Programming and Evolvable Machines* journals. She was the recipient of the Evo$*$ Award for Outstanding Contribution to Evolutionary Computation in Europe in 2022.

**Marcus Gallagher** (Senior Member, IEEE) received the BCompSc and GradDipSc degrees in computer science from The University of New England, Armidale, NSW, Australia, in 1994 and 1995, respectively, and the Ph.D. degree in computer science and engineering from The University of Queensland, Brisbane, QLD, Australia, in 2000.

He is currently an Associate Professor with the School of Information Technology and Electrical Engineering, The University of Queensland. His research interests include evolutionary computation, algorithm benchmarking, games, optimization, and machine learning.

Dr. Gallagher is an Associate Editor for the *Swarm and Evolutionary Computation* journal and Current Chair (2021–2022) of the Queensland Chapter of the IEEE Computational Intelligence Society.