Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 8, No. 2, May, Pp. 599-606

Evolutionary Algorithm in Game – A Systematic Review

Hendrawan Armanto[1,2], Harits Ar Rosyid*[1], Muladi[1], Gunawan[2]
Universitas Negeri Malang, Malang, Indonesia[1]
Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia[2]

Article Info
Keywords:
Evolutionary Algorithm, Game, Systematic Review

Article history:
Received: April 14, 2023
Accepted: May 30, 2023
Published: May 31, 2023

Cite:
H. Armanto, H. A. . Rosyid, Muladi, and Gunawan, "Evolutionary Algorithm in Game – A Systematic Review", KINETIK, vol. 8, no. 2, May 2023. https://doi.org/10.22219/kinetik.v8i2.1714

*Corresponding author.
Harits Ar Rosyid
E-mail address:
harits.ar.ft@um.ac.id

Abstract
Research in the game field is increasingly numerous and challenging. The high interest in research on games is influenced by public awareness of the importance of games in developing ways of thinking, although it is undeniable that many people only pursue pleasure in playing games. In the past, not much games research has influenced into topics such as artificial intelligence, education, or other computer topics. But now games are having a tremendous impact on these topics. In fact, not infrequently games are used in various areas of life. Right now, artificial intelligence is an integral part of the game. If before, it was only used for creating an enemy. Right now artificial intelligence can affect various things, starting from assets, game difficulty levels, non-player characters (NPC), and even bots (AI agents) to run player characters. The complexity of artificial intelligence which is getting higher and higher requires a good optimization algorithm. The evolutionary algorithm is one of the optimization algorithms, even though it cannot find the best one, with the high speed it can find a good solution. Through this paper review, good conclusions are drawn regarding the use of evolutionary algorithms, representations made, fitness functions used, ways to prove a success, to what topics should be studied further.

1. Introduction

Research in games has now become an interesting and hot topic among researchers. Not inferior to other research topics in computer science such as computer vision, data science, or other popular topics. Now Games have arised as one of the research topics that are felt to be useful. For example, according to bestcolleges[1], game-related research can be directed to its use to solve current world problems. In fact, not infrequently, these studies enter the area of computer science topics. This happens, because research in the game world is not limited to one or two directions of research but can cross into various topics. Some research that can be done in the game world is research related to game theory [1]–[3], game design [4]–[6], artificial intelligence [7], user experience [8], game balancing [9], and various other types of research. Although there are various types of research on games, this literature review is focused on research that uses evolutionary algorithms in its development. This is done considering that the evolutionary algorithm is a good and fast optimization algorithm, but it is not widely applied to the game world, even though the evolutionary algorithm itself continues to grow and get faster. This is a loss in game research. Because for certain cases the evolutionary algorithm is the same or even better than other algorithms. It is hoped that in the future with this literature review, the use of evolutionary algorithms will become more intent and can be utilized by other game research.

2. Evolutionary Algorithm

Evolutionary algorithms[2] [10], [11] are optimization algorithms obtained from the behavior of living organisms such as the behavior of birds in traveling [12], cats in searching for prey [13], or penguins in warming their bodies [14]. Apart from being based on the behavior of living things, the concept of evolutionary algorithms is also derived from natural phenomena such as the evolution of living things, water droplets, or symbiotic relationships between living

[1]https://www.bestcolleges.com/bootcamps/guides/top-technology-research-projects/
[2] Often also called Metaheuristic algorithms or Swarm Intelligence algorithms (however, the term swarm intelligence can only be used for evolutionary algorithms that use the concept of a certain group of living things). However, the term evolutionary algorithm itself can mean two things, first is a collection of algorithms that use evolutionary concepts or second is the EA algorithm which was discovered before the genetic algorithm.

things. Although the workings of these algorithms vary[3], the basic concepts used are similar. Which comes from its predecessor, namely genetic algorithm, and particle swarm optimization.

The genetic algorithm [15]–[17] is the starting point of all existing evolutionary algorithms, where the concept was originally obtained from Darwin's Theory of Evolution. According to Darwin, living things are creatures that reproduce, but not all offspring are good in each generation. Offspring that cannot adapt will be exposed to natural selection while offspring that can adapt will evolve and pass their genes on to the next generations. It is hoped that after several generations, the remaining offspring will be the superior and the best. Based on this theory, the genetic algorithm represents a candidate solution to the problem as an individual living being. These solutions will continue to grow and get better, and up to a certain generation, the best candidate solutions will be used as results to solve the problem. Table 1 is the use of genetic algorithms both in research and in community applications. Although until now there have been various new evolutionary algorithms. From Table 1, it can be seen that the genetic algorithm is still the favorite algorithm. This is because many assume that this algorithm is still good and is the origin of the evolutionary algorithm.

Table 1. Genetic Algorithm Implementation [15]

Category	Case Problem
Operation Management	Facility layout Scheduling Inventory Control Forecasting and Network Design Sensor-Based Robot Path Planning
Multimedia	Information Security Image Processing Video Processing Medical Imaging Precision Agriculture Gaming
Wireless Networking	Load Balancing Localization Bandwidth and Channel Allocation Network Routing Protocol

In addition to genetic algorithms, one of the predecessors of evolutionary algorithms is Particle Swarm Optimization (PSO) [12], [18]. PSO is the first evolutionary algorithm that uses number representation instead of permutations (such as genetic representation). The use of number representation had a big impact on the world of evolutionary algorithms and has been followed by other algorithms since it can even be said that there are almost no algorithms after PSO that still use permutation representations, although it is undeniable that in some cases the use of permutation representations will make the process much easier than numerical representation. In addition to introducing number representation, PSO is also the first algorithm to implicitly introduce the concept of exploration and exploitation. This concept is the starting point for various modifications offered by subsequent evolutionary algorithms. Not inferior to genetic algorithms, PSO is also widely used for various research and applications in society. Some of the popular studies are multi-objective applications [19], Neural Network Weight Modification [20], and Biometric Security Systems [21]. Even the most recent research using PSO touches on issues such as Airfoil Design [22] and Disease Detection [23]. This proves that just like the genetic algorithm, even today PSO is still one of the best-trusted evolutionary algorithms.

3. Evolutionary Algorithm in Games

Research on games continues to grow and increase, including research using evolutionary algorithms. Currently, various types of evolutionary algorithms are studied to handle cases in games, where each type of case has its own complexity and target solution. The following sub-chapters will explain in more detail regarding each type of case that has been studied, the evolutionary algorithm used, to the results achieved by the research.

[3] Several evolutionary algorithms that have been developed can be viewed on the Wikipedia website via the link: https://en.wikipedia.org/wiki/Table_of_metaheuristics (however this list is not complete because Wikipedia is not an official research website but a knowledge website that filled by the general public)

3.1 Content Generation

The creation of content[4] in games is one case that can be solved by evolutionary algorithms. This is because determining content is generally the preference of game makers so there are no specific rules in it. The most important is the content is in line with the game and the players like it. In general, game creators will use this automatic content generation when the game is large or dynamic. it easier for game makers considering that content creation requires a lot of funds. Table 2 contains data from several recent studies related to content generation using an evolutionary algorithm.

Table 2. Content Generation using Evolutionary Algorithm

No	Research Title	Content Generation Type	Game Genre/Name	Evolutionary Algorithm
1	Automatic Game World Generation for Platformer Games using Genetic Algorithm [24]	Level / Map	2D Platformer	Genetic
2	Evolutionary Generation of Game Levels [25]	Level / Map	Top-Down Shooter	Genetic
3	Levels for Hotline Miami 2: Wrong Number using Procedural Content Generations [26]	Level / Map	Hotline Miami 2	EA
4	Game Level Layout Generation using Evolved Cellular Automata [27]	Level / Map	Adventure	Genetic
5	Transposition of Location-based Games: using Procedural Content Generation to Deploy Balanced Game Maps to Multiple Locations [28]	Level / Map	Faith Quest	Genetic
6	Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning [29]	Quest	2D RPG	Genetic

There are 2 types of content generation in Table 2, where both types represent the representation of the evolutionary algorithm. In the content generator for level/map generation, the researchers used a 2-dimensional map matrix as a representation. The best map results produced by the evolutionary algorithm are used to form user-facing levels/maps in either 2D or 3D games. Whereas in the content generator for quests, researchers use the quest order as a representation. The best-resulting quest sequence is given to players to play. The main difference in these studies is the fitness function used. Due to the different types of games used, the fitness function will adjust to the game. Whether good or not, the fitness function of content generation is always determined by the preference of the game maker or researcher. The higher the fitness value produced, the more content that is formed is closer to the preference.

Based on the results of using the evolutionary algorithm from the research in table 2, it can be concluded that the evolutionary algorithm is good for content generation. The content generated by the genetic algorithm is said to meet the target of the respective research. Even the complexity of the generated content can also be adjusted based on the given parameters. However, it is very unfortunate that existing research does not dare to try or use evolutionary algorithms other than genetic algorithms. Only 1 out of 6 studies used another evolutionary algorithm (EA Algorithm). But unfortunately, this algorithm was a predecessor of Genetic Algorithm. Apart from the lack of variation in the evolutionary algorithm used, the research in Table 2 shows that Genetic Algorithm is can be used today and produces quality solutions that meet the needs of modern research.

3.2 Difficulty Adjustment

Another application of the evolutionary algorithm is adjusting the difficulty[5] level of a game. Generally, the difficulty level of a game is static, this is done to make it easier for developers to determine the balance of the games they make. However, this does not apply to players, everyone has different ways and abilities to play[6]. This difference has led to various studies related to the level of difficulty of the game, where the level of difficulty is expected to be adjusted according to the player's abilities[7]. Apart from being researched, these studies have also begun to be implemented by various game developers so that the game is perceived as more interesting by most players. Table 3 contains data from several recent studies related to difficulty adjustment using an evolutionary algorithm.

[4] The term content in the discussion above generally in existing research only means 2 things, namely maps (levels) or mission flow (quests)

[5] The level of difficulty is based on the ability of the AI Agent. The higher level of difficulty, the better ability of the AI Agent and the harder it is to beat.

[6] In facing game's challenge, there are players who can play well but there are also players who have difficulty.

[7] The better the player plays it, the more difficult the game that must be faced by the player.

Table 3. Difficulty Adjustment using Evolutionary Algorithm

No	Research Title	Game Genre/Name	Evolutionary Algorithm
1	Reinforced Evolutionary Algorithms for Game Difficulty Control [30]	Pong dan Ghost Story	EA
2	Solving the Balance Problem of Massively Multiplayer Online Role-playing Games using Coevolutionary Programming [31]	MMORPG	Cooperative Coevolutionary Algorithm, Genetic, Stimulated Annealing, Hybrid Discrete Particle Swarm Optimization
3	Dynamic Difficulty Adjustment with a simplification ability using Neuroevolution [32]	Fighting	Genetic
4	Dynamic Difficulty Adjustment in Digital Games Using Genetic Algorithms [33]	Empire Game	Genetic
5	Genetic Algorithm in Survival Shooter Games NPCs [34]	Survival Shooter	Genetic

Based on the studies from Table 3, the AI Agent is modified at any time according to the player's abilities. There are two types of evolutionary algorithms implementations. First, implementing evolutionary algorithms without the help of other algorithms, or second combining evolutionary algorithms (EA) with decision-making algorithms such as reinforcement learning or neural networks. The construct of the first EA requires tuning of parameters or properties of the AI Agent. Meanwhile, the second type of EA requires optimization of the decision-making algorithm's parameters. In addition, there are techniques to indicate successful results from the EA:

- Based on the fitness value generated by the evolutionary algorithm. Each study uses a different fitness function, depending on the type and way of playing the game used. However, the fitness function is expected to be able to describe the success of research in measuring game difficulty adjustment. Some of the fitness functions that are used include Total Difficulty Difference (in percent), the percentage of player wins, scores obtained by players, or several other parameters such as vitality, HP increase or decrease, to agent lifetime.
- Based on the results of the players' questionnaire. Questionnaires are a valid way of proving research related to adjusting difficulty levels. This is because players have a subjective understanding of the current difficulty level. No matter how good the fitness calculation is, it does not represent the player's experience. Thus, difficulty level should be considered as a subjective case.

(a) (b)

Figure 1. Questionnaire Result Example [34]
(a) Based on Enemy Difficulty
(b) Based on Changes in the Level of Difficulty

Figure 1 is the result of a questionnaire in one of the studies. From Figure 1 (a), it can be concluded there were 61.1% of players felt playing with a normal level of difficulty and only 5.6% thought the game was too easy. Figure 1 (b) states that 88.9% of players experience progressive difficulty levels based on their skills.

- Finally, based on expert[8] observations. Apart from questionnaires to players, the second way to better[9] validate the level of difficulty is to record the game and carry out observations and analysis by an expert. In contrast to the questionnaire, expert observations can be made with a small number of players, but the players who play it should understand the game being played so that changes in difficulty levels are visible to experts.

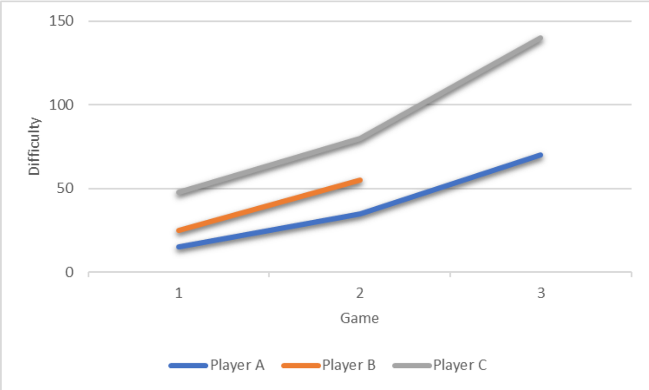

Figure 2. Game Difficulty Change Graphics

Figure 2 is the result of observing the change in difficulty level between three players. The graph shows that the difficulty level increases, but vary for each player. Player A slightly improved but another player drastically improved (Player C). Meanwhile, player B failed to complete the game.

3.3 Non-Player Character (NPC) Behaviour

The NPCs in this study are computer-controlled characters. They can be a player's companion, an enemy, or a supporting character who directs the storyline. Based on such roles, it is necessary to regulate good behavior so that it can optimize the gaming experience. For example, an accompanying NPC helps a player in achieving a goal. The more complex of a role for an NPC, the more difficult and lengthier the process of determining the behavior of the NPC. Hence, this large search space is a problem to be solved by an evolutionary algorithm. Table 4 contains data from several recent studies related to NPC Behaviour using an evolutionary algorithm.

Table 4. NPC Behaviour using Evolutionary Algorithm

No	Research Title	Game Genre/Name	Evolutionary Algorithm
1	Using Genetic Algorithms to Evolve Character Behaviours in Modern Video Games [35]	Unreal Tournament 2004	Genetic
2	Viral Infection Genetic Algorithm with Dynamic Infectability for Pathfinding in A Tower Defense Game [36]	Tower Defense	Genetic
3	Swarm Intelligence Scheme for Pathfinding and Action Planning of Non-player Characters on a Last-Generation Video Game [37]	First Person Shooting	Particle Swarm Optimization
4	Evolutionary Design of Cooperative Predation Strategies [38]	Strategy	EA
5	Evolved Communication Strategies and Emergent Behaviour of Multi-Agents in Pursuit Domains [39]	Strategy	Genetic
6	Advanced Gameplay Strategy Based on Grey Wolf Optimization [40]	Shooter	Grey Wolf Optimization

Based on the studies in Table 4, NPC behavior can be further sub-categorized into its level of difficulty. Some of the sub-categories are the decision-making, the movement of an NPC, and the preparation of strategies for a group of NPCs. This is the open topic of NPC research.

[8] An expert here can be a player who has been playing for a long time or is proficient at playing. It can also be a game developer whose focus is on the same or equivalent game model.
[9] Compared with the fitness value which is the preference of researchers or game makers.

The representation used in this study is adjusted to the sub-category of the research. The representation of decisions made by the NPC depends on the type of game. This includes parameters or properties of the NPC such as health, weapons, or other parameters. Meanwhile, the representation of an NPC's movement depends on the position of the player, enemy, or other objects. Finally, the representation of the NPC group, apart from depending on the type of game, is also based on the parameters/conditions of the team. A good team can support one another. While fitness functions generally depend on the type of game played, different ways of playing will have different fitness functions.

There are methods of measuring NPC behavior including fitness, questionnaires, expert analysis, and competition between NPCs. This research opts for NPCs competition. Figure 3 shows the research results from a battle between NPC. The graph shows the difference in fitness value and other NPCs configurations. The more positive the value of the difference, the better the NPC.

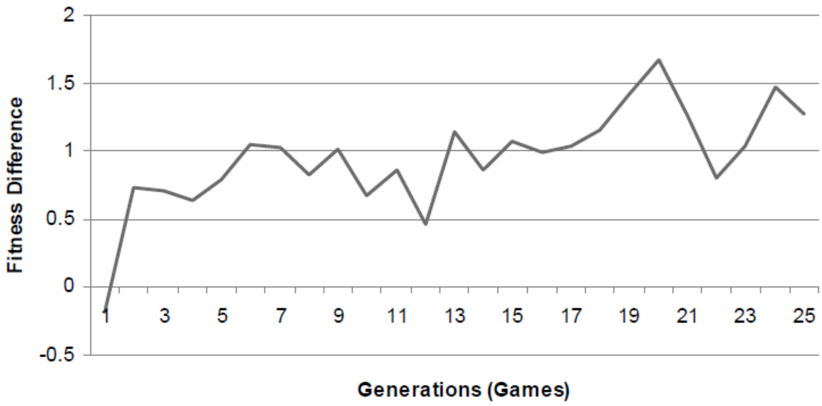

Figure 3. Fitness Difference Between Research Results and other NPC [35]

3.4. AI Agent to Replace Player

The approach of this NPC behavior research is more directed to artificial intelligence that mimics player behavior (often also called bots). This is in contrast to the previous sub-chapter which analyses the impact of NPCs on players. Table 5 shows AI in games and their algorithms applied. Table 5 contains data from several recent studies related to AI agents using an evolutionary algorithm.

Table 5. AI Agent to Replace Player using Evolutionary Algorithm

No	Research Title	Game Genre/Name	Evolutionary Algorithm
1	Evolutionary Algorithms for Controllers in Games [41]	Super Mario	Genetic
2	EVOR: An Online Evolutionary Algorithm for Car Racing Games [42]	Car Racing	EA
3	Optimizing Hearthstone Agents using an Evolutionary Algorithm [43]	Hearthstone	EA
4	Rolling Horizon Evolutionary Algorithms for General Video Game Playing [44]	GVGAI Games	Genetic
5	Optimizing player behavior in a real-time strategy game using evolutionary algorithms [45]	Realtime Strategy	Genetic
6	There Can Be Only One: Evolving RTS Bots via Joust Selection [46]	Realtime Strategy	Genetic

AI agents in research from Table 5 use the same representations with NPC behavior: 1) the parameter/property of the AI Agent and 2) the weights of the decision-making algorithm. In NPC behavior, the fitness function is based on its condition and the probable impact on the players. However, in this study, the fitness function is based on the agent’s parameters (e.g. health, mana points, etc.) and the results of the game (such as the number of wins, winning conditions, etc.). The better the fitness value, the better the AI Agent's ability to complete challenges in the game. Experiments were carried out by calculating fitness values, game simulation results (for single-player games), match results with other AI (for multi-player games), and game analysis by experts. According to these, the evolutionary algorithm should fit to make AI Agents human-like players.

4. Conclusion

Based on the studies from some relevant articles, some conclusions can be made. First, even though the development of the evolutionary algorithm is relatively rapid (minimum one algorithm per year), applications in games are lacking. Most researchers prefer genetic algorithms due to the easier implementation of genetic algorithms compared to newer evolutionary algorithms or lack of confidence when applying new algorithms in games. Second, if we talk about game problem representation, there are two types of representation in evolutionary algorithms for games: the parameters or properties of the problem and the inclusion of the machine learning algorithm's parameters. Third, in terms of research experiments, the existing studies used the preferences of the researcher or developer as the basis for their fitness function. In addition, the ability targets of AI Agents are also included, such as the impact on player health points, the number of wins, the number of losses, or something else. Fourth, there are four methods to measure experiments for this field of research: Measure success based on the fitness value where the higher the number the more human the AI; Questionnaires to players, concerning their knowledge of the game they played. However, this requires a vast number and types of participants to ensure the objectiveness of the responses; Analysis by an expert or a player who understands the problem being researched. However, subjectivity could hinder the expected result if not anticipated; Finally, an alternative approach via competition between NPCs. The NPC's winning rate will later be used as proof of success whether the NPC under study works well or not. Last, research on the NPC's behavior and AI agents for a game is extensive, deep, and growing research until recently. This is because both NPCs and AI Agents are highly dependent on the development of the game.

References

[1] Z. Hellman and M. Pintér, "Charges and bets: a general characterisation of common priors," *Int. J. Game Theory*, vol. 51, no. 3, pp. 567–587, 2022. https://doi.org/10.1007/s00182-022-00805-4

[2] E. C. Ramón and E. Gutiérrez-López, "The equal collective gains value in cooperative games," *Int. J. Game Theory*, vol. 51, no. 1, pp. 249–278, 2022. https://doi.org/10.1007/s00182-021-00791-z

[3] D. Christopher Opolot, "On the relationship between p-dominance and stochastic stability in network games," *Int. J. Game Theory*, vol. 51, no. 2, pp. 307–351, 2022. https://doi.org/10.1007/s00182-021-00794-w

[4] M. S. O. Almeida and F. S. C. da Silva, "A Systematic Review of Game Design Methods and Tools BT - Entertainment Computing – ICEC 2013," 2013, pp. 17–29. https://doi.org/10.1007/978-3-642-41106-9_3

[5] P. Lankoski, "Game Design Research: An Introduction to Theory & Practice," 2017.

[6] D. Bulut, Y. Samur, and Z. Cömert, "The effect of educational game design process on students' creativity," *Smart Learn. Environ.*, vol. 9, no. 1, p. 8, 2022. https://doi.org/10.1186/s40561-022-00188-9

[7] B. Xia, X. Ye, and A. Abuassba, *Recent Research on AI in Games*. 2020.

[8] A. Persada, "User experience on games development trends," *J. Phys. Conf. Ser.*, vol. 1341, p. 42010, Oct. 2019. https://doi.org/10.1088/1742-6596/1341/4/042010

[9] A. Becker and D. Görlich, *Game Balancing – A Semantical Analysis*. 2019.

[10] A. N. Sloss and S. Gustafson, "2019 Evolutionary Algorithms Review," Jun. 2019.

[11] P. A. Vikhar, "Evolutionary algorithms: A critical review and its future prospects," in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, 2016, pp. 261–265. https://doi.org/10.1109/ICGTSPICC.2016.7955308

[12] M.-P. Song and G.-C. Gu, "Research on particle swarm optimization: a review," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, 2004, vol. 4, pp. 2236–2241 vol.4. https://doi.org/10.1109/ICMLC.2004.1382171

[13] A. Ghaedi, A. K. Bardsiri, and M. J. Shahbazzadeh, "Cat hunting optimization algorithm: a novel optimization algorithm," *Evol. Intell.*, vol. 16, no. 2, pp. 417–438, 2023. https://doi.org/10.1007/s12065-021-00668-w

[14] O. W. Khalid, N. A. M. Isa, and H. A. Mat Sakim, "Emperor penguin optimizer: A comprehensive review based on state-of-the-art meta-heuristic algorithms," *Alexandria Eng. J.*, vol. 63, pp. 487–526, 2023. https://doi.org/10.1016/j.aej.2022.08.013

[15] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, 2021. https://doi.org/10.1007/s11042-020-10139-6

[16] A. Lambora, K. Gupta, and K. Chopra, "Genetic Algorithm- A Literature Review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 380–384. https://doi.org/10.1109/COMITCon.2019.8862255

[17] M. Kumar, M. Husain, N. Upreti, and D. Gupta, "Genetic algorithm: Review and application," *J. Inf. Knowl. Manag.*, Jul. 2010.

[18] D. Freitas, L. G. Lopes, and F. Morgado-Dias, "Particle Swarm Optimisation: A Historical Review Up to the Current Developments," *Entropy*, vol. 22, no. 3. 2020. https://doi.org/10.3390/e22030362

[19] Y. Yang, Q. Liao, J. Wang, and Y. Wang, "Application of multi-objective particle swarm optimization based on short-term memory and K-means clustering in multi-modal multi-objective optimization," *Eng. Appl. Artif. Intell.*, vol. 112, p. 104866, 2022. https://doi.org/10.1016/j.engappai.2022.104866

[20] C. Zhang, H. Shao, and Y. Li, "Particle swarm optimisation for evolving artificial neural network," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. "cybernetics evolving to systems, humans, organizations, and their complex interactions" (cat. no.0*, 2000, vol. 4, pp. 2487–2490 vol.4. https://doi.org/10.1109/ICSMC.2000.884366

[21] Q. Xiong, X. Zhang, X. Xu, and S. He, "A Modified Chaotic Binary Particle Swarm Optimization Scheme and Its Application in Face-Iris Multimodal Biometric Identification," *Electronics*, vol. 10, no. 2. 2021. https://doi.org/10.3390/electronics10020217

[22] S. Mirjalili, J. Song Dong, A. Lewis, and A. S. Sadiq, "Particle Swarm Optimization: Theory, Literature Review, and Application in Airfoil Design BT - Nature-Inspired Optimizers: Theories, Literature Reviews and Applications," S. Mirjalili, J. Song Dong, and A. Lewis, Eds. Cham: Springer International Publishing, 2020, pp. 167–184. https://doi.org/10.1007/978-3-030-12127-3_10

[23] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, "A Systematic Literature Review on Particle Swarm Optimization Techniques for Medical Diseases Detection," *Comput. Math. Methods Med.*, vol. 2021, p. 5990999, 2021. https://doi.org/10.1155/2021/5990999

[24] A. S. Kholimi, A. Hamdani, and L. Husniah, “Automatic Game World Generation for Platformer Games Using Genetic Algorithm,” in *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2018, pp. 495–498. https://doi.org/10.1109/EECSI.2018.8752741
[25] A. M. Connor, T. J. Greig, and J. Kruse, “Evolutionary generation of game levels,” *EAI Endorsed Trans. Creat. Technol.*, vol. 5, no. 15, 2018. http://dx.doi.org/10.4108/eai.10-4-2018.155857
[26] J. A. Brown, B. Lutfullin, P. Oreshin, and I. Pyatkin, “Levels for Hotline Miami 2: Wrong Number Using Procedural Content Generations,” *Comput.*, vol. 7, p. 22, 2018.
[27] A. Pech, M. Masek, C. Lam, and P. Hingston, “Game level layout generation using evolved cellular automata,” *Conn. Sci.*, vol. 28, pp. 1–20, Feb. 2016. https://doi.org/10.1080/09540091.2015.1130020
[28] L. Maia, W. Viana, and F. Trinta, “Transposition of Location-based Games: Using Procedural Content Generation to deploy balanced game maps to multiple locations,” *Pervasive Mob. Comput.*, vol. 70, p. 101302, Jan. 2021. https://doi.org/10.1016/j.pmcj.2020.101302
[29] E. Soares de Lima, B. Feijó, and A. Furtado, *Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning*. 2019.
[30] G. Cui *et al.*, “Reinforced Evolutionary Algorithms for Game Difficulty Control,” 2021. https://doi.org/10.1145/3446132.3446165
[31] H. Chen, Y. Mori, and I. Matsuba, “Solving the balance problem of massively multiplayer online role-playing games using coevolutionary programming,” *Appl. Soft Comput. J.*, vol. 18, pp. 1–11, May 2014. https://doi.org/10.1016/j.asoc.2014.01.011
[32] M. Shakhova and A. Zagarskikh, “Dynamic Difficulty Adjustment with a simplification ability using neuroevolution,” *Procedia Comput. Sci.*, vol. 156, pp. 395–403, 2019. https://doi.org/10.1016/j.procs.2019.08.219
[33] M. Weber and P. Notargiacomo, “Dynamic Difficulty Adjustment in Digital Games Using Genetic Algorithms,” in *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2020, pp. 62–70, doi: 10.1109/SBGames51465.2020.00019.
[34] A. Henrique, R. Soares, R. Dazzi, and R. Lyra, *Genetic Algorithm in Survival Shooter Games NPCs*. 2020.
[35] T. Bullen and M. Katchabaw, “Using Genetic Algorithms To Evolve Character Behaviours In Modern Video Games,” 2008.
[36] G. T. Galam, T. P. Remedio, and M. A. Dias, “Viral Infection Genetic Algorithm with Dynamic Infectability for Pathfinding in a Tower Defense Game,” in *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2019, pp. 198–207. https://doi.org/10.1109/SBGames.2019.00034
[37] G. Díaz and A. Iglesias, “Swarm Intelligence Scheme for Pathfinding and Action Planning of Non-player Characters on a Last-Generation Video Game BT - Harmony Search Algorithm,” 2017, pp. 343–353.
[38] A. Von Moll, P. Androulakakis, Z. Fuchs, and D. Vanderelst, “Evolutionary Design of Cooperative Predation Strategies,” in *2020 IEEE Conference on Games (CoG)*, 2020, pp. 176–183. https://doi.org/10.1109/CoG47356.2020.9231945
[39] G. Grossi and B. Ross, “Evolved communication strategies and emergent behaviour of multi-agents in pursuit domains,” in *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, 2017, pp. 110–117. https://doi.org/10.1109/CIG.2017.8080423
[40] K. Singh, A. V Singh, and S. K. Khatri, “Advanced Gameplay Strategy Based on Grey Wolf Optimization,” in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 2019, pp. 183–185. https://doi.org/10.1109/ISCON47742.2019.9036295
[41] O. Mautschke, “Evolutionary Algorithms for Controllers in Games,” 2019.
[42] S. Nallaperuma, F. Neumann, M. reza Bonyadi, and Z. Michalewicz, *EVOR : An Online Evolutionary Algorithm for Car Racing Games*. 2014. https://doi.org/10.1145/2576768.2598298
[43] P. García-Sánchez, A. Tonda, A. J. Fernández-Leiva, and C. Cotta, “Optimizing Hearthstone agents using an evolutionary algorithm,” *Knowledge-Based Syst.*, vol. 188, Jan. 2020. https://doi.org/10.1016/j.knosys.2019.105032
[44] R. D. Gaina, S. Devlin, S. M. Lucas, and D. Perez-Liebana, “Rolling horizon evolutionary algorithms for general video game playing,” *IEEE Trans. Games*, vol. 14, no. 2, pp. 232–242, 2021. https://doi.org/10.48550/arXiv.2003.12331
[45] A. Fernández-Ares, A. M. Mora, J. J. Merelo, P. García-Sánchez, and C. Fernandes, “Optimizing player behavior in a real-time strategy game using evolutionary algorithms,” in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 2017–2024. https://doi.org/10.1109/CEC.2011.5949863
[46] A. Fernández-Ares, P. García-Sánchez, A. M. Mora, P. A. Castillo, and J. J. Merelo, “There Can Be only One: Evolving RTS Bots via Joust Selection BT - Applications of Evolutionary Computation,” 2016, pp. 541–557. https://doi.org/10.1007/978-3-319-31204-0_35