



APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNIQUES IN MS. PAC-MAN GAME: A REVIEW

TSE GUAN TAN *

JASON TEO **

Abstrak

Teknik Kecerdasan Buatan (AI) berjaya digunakan dan diaplikasikan dalam pelbagai bidang, termasuk pembuatan, kejuruteraan, ekonomi, perubatan dan ketenteraan. Kebelakangan ini, terdapat minat yang semakin meningkat dalam Permainan Kecerdasan Buatan atau permainan AI. Permainan AI merujuk kepada teknik yang diaplikasikan dalam permainan komputer dan video seperti pembelajaran, *pathfinding*, perancangan, dan lain-lain bagi mewujudkan tingkah laku pintar dan autonomi kepada karakter dalam permainan. Objektif utama kajian ini adalah untuk mengemukakan beberapa teknik yang biasa digunakan dalam merekabentuk dan mengawal karakter berasaskan komputer untuk permainan *Ms Pac-Man* antara tahun 2005-2012. *Ms Pac-Man* adalah salah satu permainan yang digunakan dalam siri pertandingan permainan diperingkat antarabangsa sebagai penanda aras untuk perbandingan pengawal autonomi. Kaedah analisis kandungan yang menyeluruh dijalankan secara ulasan dan sorotan literatur secara kritikal. Dapatan kajian menunjukkan bahawa, walaupun terdapat berbagai teknik, limitasi utama dalam kajian terdahulu untuk mewujudkan karakter permainan Pac Man adalah kekurangan *Generalization Capability* dalam kepelbagaian karakter permainan. Hasil kajian ini akan dapat digunakan oleh penyelidik untuk meningkatkan keupayaan *Generalization AI* karakter permainan dalam Pasaran Permainan Kecerdasan Buatan.

* Senior Lecturer at Faculty of Creative Technology and Heritage, Universiti Malaysia Kelantan, Malaysia.

** Senior Lecturer at Faculty of Computing and Informatics, University Malaysia Sabah, Malaysia.





Abstract

Artificial Intelligence (AI) techniques are successfully used and applied in a wide range of areas, including manufacturing, engineering, economics, medicine and military. In recent years, there has been an increasing interest in Game Artificial Intelligence or Game AI. Game AI refers to techniques applied in computer and video games such as learning, pathfinding, planning, and many others for creating intelligent and autonomous behaviour to the characters in games. The main objective of this paper is to highlight several most common of the AI techniques for designing and controlling the computer-based characters to play Ms. Pac-Man game between years 2005-2012. The Ms. Pac-Man is one of the games that used as benchmark for comparison of autonomous controllers in a series of international Game AI competitions. An extensive content analysis method was conducted through critical review on previous literature related to the field. Findings highlight, although there was various and unique techniques available, the major limitation of previous studies for creating the Ms. Pac-Man game characters is a lack of generalization capability across different game characters. The findings could provide the future direction for researchers to improve the Generalization A.I capability of game characters in the Game Artificial Intelligence market.

Keyword: Artificial Intelligence Techniques, Game Artificial Intelligence, Ms. Pac-Man

* Senior Lecturer at Faculty of Creative Technology and Heritage, Universiti Malaysia Kelantan, Malaysia.

** Senior Lecturer at Faculty of Computing and Informatics, University Malaysia Sabah, Malaysia.





1.0 Introduction

Games have emerged as powerful platforms for testing Artificial Intelligence (AI) techniques and ideas. Some major international Game AI competitions are held each year, such as the IEEE Conference on Computational Intelligence and Games (CIG), the IEEE World Congress on Computational Intelligence (WCCI) and the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE) for researchers and practitioners from a wide range of areas including game design, graphics, interface design, scripting and sound to discuss and share recent advances and developments in game industry and also to explore future directions in the research and practice of Game AI. The Ms. Pac-Man game is an interesting, non-deterministic and challenging test-bed for evaluating machine as well as human intelligence (Lucas, 2005). Therefore, it is an ideal benchmark to test and analyse existing AI techniques whether the generated computer-based controllers can play the game in an intelligent manner similar to that of a human playing the game. Figure 1 shows the Ms. Pac-Man game.

The paper has been organised in the following way. Section 2 presents the review of the application of AI techniques in Ms. Pac-Man game between years 2005-2012, followed by the discussions in Section 3. Finally, a conclusion of the paper is provided in Section 4.

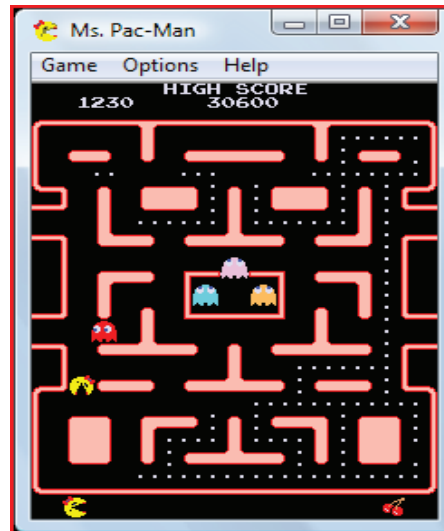


Figure 1: Ms. Pac-Man Game





2.0 Artificial Intelligence Techniques Applied In Ms. Pac-Man Game

In this section, a review of the application of AI techniques in Ms. Pac-Man game between years 2005-2012 is presented. Table 1 shows a summary of a literature survey on Ms. Pac-Man agent and ghosts' AI controllers. As can be seen from the table, the techniques can be classified under general or specific methods. The controllers are created by general methods, which do not involve specific rules and conditions, so that they are applicable to a variety of games or domains. However, specific methods are designed under particular rules and conditions.

In 2005, Lucas (2005) proposed an approach, which evolves Artificial Neural Network (ANN) by using Evolutionary Strategy (ES) to play Ms. Pac-Man. The input of the network is a handcrafted feature vector that consists of the distance to each normal ghost, distance to each edible ghost, location of current node, distance to nearest pill, the distance to nearest power pill and distance to nearest junction, whereas the calculated output is a score for every possible next location given the agent's current location. ES is applied to evolve ANN connection weights. The best evolved agent with (10+10)-ES had an average score of 4781 and a maximum of 9200 over 100 runs of the non-deterministic game.

In 2007, Szita and Lorincz (2007) proposed a controller based on Reinforcement Learning (RL) and Cross-Entropy (CE) method for generating low-complexity Rule-Based (RB) policies to control the movement of Ms. Pac-Man agent. The RL is used to learn to combine the hand-coded high-level actions and observations into a good policy. Furthermore, the CE method (de Boer et al., 2005) is used for optimizing RB policies from a large pool of rules that involves the following three iterative phases: (1) generate random policies based on current parameter set, (2) evaluate the policies on the game, (3) update the set of parameter variables using the CE method update rules to produce better policies in the next iteration. The average score achieved by the optimized agent was 8186 points with an interquartile range of 2687, while the human players had an average score of 8064 points with an interquartile range of 4965 in 10 parallel learning runs.

In 2008, Handa and Isozaki (2008), and Handa (2008) proposed evolutionary fuzzy systems for creating autonomous Ms. Pac-Man controllers to play the game. It is used to assist the controller to select an appropriate action at each time step, which reacts to changes in the game environment. The ES is applied for optimizing the parameters of the fuzzy rules. Additionally, Handa studied the use of RL in the proposed system. Q-learning with Cerebellar Model Articulation Controller (CMAC) (Albus, 1975) is utilized to detect and respond to critical situations for agent. The best score in the experiments was approximately 5700 points. Wirth and Gallagher (2008) created a computer-controlled game agent for playing Ms. Pac-Man according to the Influence Map Model (IMM) by capturing the essentials of game. Three main types of parameter: eat dots, avoid ghosts and eat edible ghosts are encoded into the model, in order to measure the fitness of the model for agent decision making. The results in their study found that the interactions between model parameters are fairly simple, and also the parameters to be used in the model are optimized reasonably straightforward. It achieved a high score of 19490 and an average of 6848.





In 2009, Fitzgerald and Congdon (2009) developed a RB agent for playing Ms. Pac-Man called RAMP. Generally, each rule consists of two main components: conditions and actions. The conditions are guidelines for selecting appropriate action among a group of actions. Fitzgerald and Congdon defines nine condition elements based on the game state and four action events involving eat closest pills, eat edible ghosts, avoid the normal ghosts and run to closest power pill to help in better decision making. The agent was able to score a mean of 10364.6 with a maximum of 18120 points in 100 tries. Robles and Lucas (2009) used the Simple Tree Search (STS) method to find possible paths for the movement of Ms. Pac-Man. The computer-based player will evaluate the entire tree down to 40 depths. The algorithm consists of three strategies that are used for path selection in the game: the Rand-Safe-Path, Most-Pills-Safe-Path and Hand-Coded-Select. As a result, the Hand-Coded-Select is the best pathfinder compared to other strategies for more intelligent path finding. With this controller, the agent was able to attain average scores of 9630 points on the screen-capture mode and over 14757 points on Ms. Pac-Man simulator. Thawonmas and Matsumoto (2009) created an intelligent Ms. Pac-Man agent called the ICE Pambush 2. The algorithm utilizes two variants of the A* Algorithm (AA) to compute the path with lowest cost such as the distance cost, the ghost cost and the node cost between that Ms. Pac-Man and the target location. The AA the uses Manhattan distance to push the search toward the goal. Specifically, there are seven decision making rules for controlling operations of the agent. The agent was able to attain an average score of 13059 and a maximum score of 24640 in 10 tries.

In addition, Burrow and Lucas (2009) investigated the differential impact of evolution and Temporal Difference (TD) learning methods in game with two function approximators: interpolated table and Multi-Layer Perceptron (MLP). The ES and TD(0) algorithms were used to optimize learning parameters in an effort to attain rapid learning for playing Ms. Pac-Man. The model requires two input variables for function approximators, the distance from the candidate node to the closest escape node, and the distance from the candidate node to the closest pill along the shortest maze path to estimate the movement rate for candidate node. The experimental results revealed that evolution of MLP has a better performance compared to TD learning. Overall, the average score of evolved controller was around 5500 points. DeLooze and Viner (2009) reported an autonomous agent that has been trained to control Ms. Pac-Man in a non-deterministic environment by using the Fuzzy State Aggregation (FSA) and Q-Learning (QL). This combination system is known as the Fuzzy Q-Learning (FQL). The agent has three selections in response to any given situation, including eats the closest pill, eats the closest power pill or escapes from closest ghost, and it made the selection based upon its knowledge of current and past situations. The agent's knowledge base is stored in the three tables that contain 27 situations with a set of Mu values. As a result, this FQL was able to obtain scores between 3000 to 5000 points.

In 2010, Galván-López *et al.* (2010b) published a paper in which they compared the effects of the Position Independent Grammatical Evolution (π GE) and standard Grammatical Evolution (GE) (Galván-López *et al.*, 2010a). In particular, this study examines the application of GE to evolve and optimize a series of rules for automatically generating Ms. Pac-Man controllers in a dynamic environment. Each rule occurs in





sequences and has the form: IF condition THEN action. It can be seen from the results that both approaches have similar performance for maximizing the score generally. However, π GE is a useful approach toward combining high-level functions. Bell *et al.* (2010) used the Dijkstra's Algorithm (DA) and Benefit-Influenced Tree Search (BITS) algorithm with six handcrafted rules to decide on the best path between two nodes of the map to control their Ms. Pac-Man. The former algorithm is implemented in specific situations and applied for eating pills and edible ghosts while the latter algorithm is helping the agent avoids dangerous situations, such as when ghosts are nearby. In addition, the authors proposed an approach called the Ghost Iris Detection for detecting the direction of ghosts to find safe paths for the agent, in order to escape from the ghosts. The performance of the optimized agent averaged a score of approximately 18000 points and the best individual score achieved was 24640 in 10 runs. Thawonmas and Ashida (2010) published a revised, improved version of its ICE Pambush 3 (Matsumoto *et al.*, 2009) for enhancing the performance of game playing in Ms. Pac-Man. The main differences of the improved ICE Pambush 3 in comparison to the original version of ICE Pambush 3 are as follows. Firstly, the authors use evolution strategy to optimize parameters such as distance and costs parameters in their controller design. Secondly, the Depth-First Search (DFS) is applied instead of the AA for path finding, because of its low computational time. Based on the Min-Max results, the improved ICE Pambush 3 achieved a high score of 9480 while the original ICE Pambush 3 achieved a maximum score of 8120. The improved ICE Pambush 3 raised the score by around 17%.

In addition, Oh and Cho (2010) integrated the DA and evolutionary neural network for improving agent's learning and decision making in Ms. Pac-Man game. The agent follows a set of rules, utilising the DA to find a safe direction. The distance between two points is measured by the Euclidean distance. If the rules are unable to cover the situation, the NeuroEvolution of Augmenting Topologies (NEAT) is used to train the ANN with Genetic Algorithm (GA) in order to determine the direction of the agent with the highest ANN output value. The average score in the experiments was 55785.5 points, with a maximum score of 111610. Tong and Sung (2010) designed Ms. Pac-Man agent based on the Monte Carlo (MC) simulation to help it avoid being caught by ghosts and able to survive for longer periods of time in the game. The DA is used as the search algorithm for finding the shortest path between two points in the maze. Essentially, the game is divided into two situations, safe situation and critical situation. In the safe situation, the agent eats pills, edible ghosts and power pill according to the circumstances. On the other hand, the MC simulation module is applied when faced with critical situation such as ghosts are nearby. The agent was able to attain an average score of 12872 and a maximum score of 22670 in 25 games. Emilio *et al.* (2010) integrated the Ant Colony Optimization (ACO) with GA to assist the Ms. Pac-Man agent in searching the appropriate paths for different circumstances in the game. The result is called the Pac-mAnt. The Pac-mAnt uses two different kinds of ants namely, collector ants and explorers. The former one is to search the paths with points, while the latter one is to search the safe paths for avoiding dangerous circumstances. Additionally, the GA is used to evolve the parameters of the artificial ants to enhance ACO performance. According to the results, the highest score achieved in a single game was 20850 and average score was 10120 points on the original game using a screen-capture interface.





In 2011, Samothrakis *et al.* (2011) studied the effects of the Monte Carlo Tree Search (MCTS) in Ms Pac-man agent and ghosts by using the 5-player maxⁿ tree representation of the game (including agent and the 4 ghosts). The search is only performed to a limited depth in the game tree and significantly different sets of payoff rules were used in the tree search for the agent compared to the ghosts. According to fundamental principle of these algorithms, the objective of each player is to maximize its payoffs independently in response to the strategies of other players. Overall, in the experiment of MCTS Ms. Pac-Man versus MCTS ghost team, the agent was able to achieve a maximum score of 6000 points. Foderaro *et al.* (2011) described a model-based approximate λ -policy iteration approach using temporal differences to optimize the paths for Ms. Pac-Man agent in dynamic environment. In their study, the Cell Decomposition (CD) approach is used for path planning from the specified start location to the desired goal destination. The workspace is decomposed into cells using the line-sweeping method. Then, the decision tree for the decomposed workspace is constructed. Each of its branches represents an alternative solution path, and also corresponds to a finite sequence of actions. The hybrid intelligent system combined the decision tree and temporal difference-based approximate λ -policy iteration for selecting the optimal path. The agent achieved a high score of 7710 with an average score of 4708 over 20 partial game simulations. Tong *et al.* (2011) designed Ms. Pac-Man agent based on the Monte Carlo Endgame Module (EM) to help it avoid being caught by the ghosts and able to survive for longer periods of time in the game. The system is basically an extension of Tong and Sung (2010) work. The EM has two main functions, called the path generation and path testing. The first is to generate a set of paths from agent node to remaining pill nodes, while the second is to test and verify whether each of the possible paths is safe by using MC simulation if agent selects to travel along that path. However, the quality of the simulation depends strongly on the ghosts' behaviour. The DA is used as the search algorithm for finding the shortest path between two points in the maze. Overall, in the 70 games, they achieved a high score of 23130 and an average of 14269. Ikehata and Ito (2011) investigated the effectiveness of the MCTS algorithm Upper Confidence bounds applied to Trees (UCT) to evade attacks from ghosts in the game of Ms. Pac-Man. The tasks for the autonomous agent are to stay alive as long as possible and to gain the highest score during that period. A set of game rules has been defined and modified by the authors to reduce the game search spaces and the computational cost. At the same times, the rules also increased the survival rates of Ms. Pac-Man by avoiding the pincer move of the ghosts. While applying the rules, the game tree will be formed and each node will be given an Upper Confidence Bounds (UCB) evaluation value. The paths or the movements of agent will be determined according to this evaluation value of the node in the game tree. The mean score was 31105.60 points and the best individual score achieved in a single game was 58990.

In addition, Alhejali and Lucas (2011) compared the efficacy of two different types of Genetic Programming (GP): standard GP and Training Camp (TC) GP by using revised version of the function set in Alhejali and Lucas (2010) work to create Ms. Pac-Man agent. In the case of large scale problems, the idea of a TC is to decompose the problem into constituent sub-tasks, then a set of training scenarios under some conditions are designed from the domain in order to satisfy the requirements of each task. The authors divided the game domain into three sub-tasks such as clearing the maze, evading the ghosts and chasing the edible





ghosts, also finding the optimal sub-agents on each of the sub-tasks by using the GP method. Finally, the model combined all of the optimized sub-agents into an optimum agent. The TC GP performed better than the standard GP with the best average and maximum scores were over 22000 and 31000 points in 10 runs, respectively. Dai *et al.* (2011) created a ghost controller dubbed NRed, which uses a fully-connected, feed-forward ANN comprised of one input layer, one hidden layer and one output layer trained by GA to generate the moves of red ghost (Blinky) to chase the Ms. Pac-Man agent. The aim was to reduce the agent playing time and score. Essentially, the architecture can be described as 5-5-4, with five input neurons, five hidden neurons and four output neurons. The parameters of the inputs are the agent's state, agent's correction value, and Blinky's coordinate and horizontal whereas the outputs are related to the directions (up, down, left and right) of the Blinky. In their approach, they evolved the connection weights for improving the performance of the ANN. Nguyen and Thawonmas (2011) created an intelligent ghost team by using the UCT, a combination model of the MCTS algorithm and UCB to collaboratively control the ghosts for catching the Ms. Pac-Man agent. They named their team as the ICE gUCT. The MCTS selects the most efficient path with the biggest UCB value from the root node until a leaf node is reached. Subsequently, expands the leaf and randomly select one of its child nodes. After that, performs the MC simulation. The ghosts such as Pinky, Sue and Inky are controlled by the MCTS whereas Blinky selects the direction according to a set of rules. Then, the UCB values of all affected branches on the path from the leaf to the root are updated. The MCTS repeats these steps until the next direction to move is selected. Furthermore, the ICE gUCT system is used to predict the movement of the Ms. Pac-Man agent by using the K-Nearest-Neighbor (KNN) algorithm. Sarjantet *al.* (2011) proposed a policy-search algorithm called the Cross-Entropy Relational Reinforcement Learning Agent (CERRLA) to enhance the capability of agent in learning the appropriate decisions for different situations in uncertain game environments. They used the Ms. Pac-Man as one of the benchmark games to evaluate their Ms. Pac-Man agent's performance. The CERRLA combines a modified online CE method and incremental rule refinements to create effective relational RB policies by a set of rules, where each rule infers an action based on some conditions. Additionally, it is applied to guide the specialisation of rules, in order for discovering and maintaining interesting rules during learning. The average score in 10 runs was approximately 3800 points, with a maximum score of approximately 7000.

In 2012, Gagne and Congdon (2012) created a game controller called the Flexible Rule-Based Intelligent Ghost Team (FRIGHT) by using the ES to evolve the rule sets for controlling the behaviors of the ghost team. The results showed that the evolved rules are better than the hand-coded rules. Pepels and Winands (2012) designed an improved version of MCTS for controlling the Ms. Pac-Man agent, which incorporates (1) a variable depth tree, (2) playout strategies for the ghost-team and Ms. Pac-Man agent, (3) including long-term goals in scoring, (4) endgame tactics and (5) a Last-Good-Reply policy for memorizing rewarding moves during playouts. As a result, the proposed controller was able to achieve better performance. Svensson and Johansson (2012) used the influence map model to create the game controllers for the Ms. Pac-Man agent and ghosts respectively. Five main types of parameter for the Ms. Pac-Man agent controller: (1) power pill influence, (2) pill influence (3) edible ghost influence (4)



ghost influence and (5) freedom of choice influence are encoded into the model. However, the influence map had to be significantly changed to the new parameters for the ghost controller, which involved three parameters: (1) ghost weight, (2) power pill distance factor and (3) Ms. Pac-Man influence.

Table 1: A Summary of a Literature Survey on Ms. Pac-Man Game

Year	Reference	Controller	Technique Used	Method
2005	Lucas	Agent	ES + ANN	General
2007	Szita and Lorincz	Agent	RL + CE + RB	Specific
2008	Handa and Isozaki	Agent	ES + FLS + RB	Specific
	Handa	Agent	QL with CMAC + ES + FLS + RB	Specific
	Wirth and Gallagher	Agent	IMM	Specific
2009	Fitzgerald and Congdon	Agent	RB	Specific
	Robles and Lucas	Agent	STS + RB	Specific
	Thawonmas and Matsumoto	Agent	AA + RB	Specific
	Burrow and Lucas	Agent	(1) ES + MLP, (2) TD	General
	DeLooze and Viner	Agent	FSA + QL	Specific
2010	¹ Galván-López <i>et al.</i>	Agent	GE + RB	Specific
	² Galván-López <i>et al.</i>	Agent	π GE + RB	Specific
	Bell <i>et al.</i>	Agent	DA + BITS + RB	Specific
	Thawonmas and Ashida	Agent	ES + DFS + RB	Specific
	Oh and Cho	Agent	NEAT + DA + RB	Specific
	Tong and Sung	Agent	MC simulation	Specific
	Emilio <i>et al.</i>	Agent	ACO + GA + hand-coded stopping conditions	Specific
	Alhejali and Lucas	Agent	GP + safety terminals	Specific
2011	Samothrakiset <i>al.</i>	Agent, Ghosts	5-player Max ⁿ MCTS	Specific
	Foderaroet <i>al.</i>	Agent	CD + TD-based approximate λ -policy iteration	Specific
	Tong <i>et al.</i>	Agent	MC simulation + EM	Specific
	Ikehata and Ito	Agent	MCTS with UCT	Specific
	Alhejali and Lucas	Agent	TC + GP	Specific
	Dai <i>et al.</i>	Ghosts	GA + ANN	General
	Nguyen and Thawonmas	Ghosts	MCTS with UCT	Specific
	Sarjantet <i>al.</i>	Agent	CERRLA	Specific





2012	Gagne and Congdon	Ghosts	ES + RB	Specific
	Pepels and Winands	Agent	MCTS	Specific
	Svensson and Johansson	Agent, Ghosts	IMM	Specific

Note: ¹Galván-López *et al.* (2010a), ²Galván-López *et al.* (2010b)

3.0 Discussions

This review highlights the major developments in research into Ms. Pac-Man game. However, it should be emphasized that the very large majority of studies reported in the literature use the AI techniques that are not general, where researches have hand-coded specific rules or techniques that can only be utilized to play well in the Ms. Pac-Man game. Hence, these systems cannot be easily modified to play other games well, which excludes them from being general systems of the AI. Most studies in the game of Ms. Pac-Man have only focussed on (1) hand-coded rule-based approaches (or search algorithms with rule engine), (2) single objective optimization and (3) single neural network models. However, there are several drawbacks to these approaches.

- 1 There are several basic weaknesses of the RB approaches. Firstly, game domains contain highly complex solution spaces that require a large numbers of rules in order to represent a set of all possible situations and corresponding actions in game environments. For instance, Szita and Lorincz (2007) lists 42 rules of a very basic hand-coded RB agent used in the Ms. Pac-Man game from the lists of actions modules and observations to control the behaviour of agent. Secondly, the computation time required to exhaustively explore the search space is very expensive indeed if large sets of rules are used by the search strategies. Third, there is a lack of generalization across different game domains or platforms because they would only apply in that particular game or genre of game.
- 2 There is a large volume of published studies describing the role of optimization techniques to solve single objective problems for finding an optimal solution. However, most of the game problems usually involve more than one objective and often conflicting objective functions, which should be considered simultaneously when designing an autonomous controller to play the game. Thus, single objective optimization techniques have many limitations for tackling complex multiobjective optimization problems. For instance, maximize the game score, minimize the computational time (including minimize the structure of ANNs) or maximize the game playing time may be considered as objectives to be optimized in Ms. Pac-Man game, with the aim is to maximize the performance of the controllers for complex game environments.





- 3 Although single neural networks are well established, but there are some common disadvantages to these approaches including over-fitting or over-training (required longer training times) leads to poor generalization performance, with a lack of robustness in the ANN models. In addition, the algorithm can suffer from a slow convergence, because training process trapped in undesirable local minima. Another problem with these methods is the limitation of training data. Normally, the training data is generated and collected from human players.

4.0 Conclusion

The main goal of this paper is to provide an overview of the existing works on the Game AI. Specifically, focus on the work around the Ms. Pac-Man game. This paper identified a number of AI techniques for designing effective computer-based controllers to play game autonomously where Ms. Pac-Man or other game genres would be useful as a test-bed for evaluating and comparing the techniques performance. As reviewed, the major limitation of previous studies for creating the Ms. Pac-Man game characters is a lack of generalization capability across different game characters. For instance, the techniques that were used to create the Ms. Pac-Man agent were not suitable for designing the ghost team.





References

- Lucas, S. M. 2005. Evolving a Neural Network Location Evaluator to Play Ms. Pac-Man. Proceedings of the IEEE Symposium on Computational Intelligence and Games, Essex, UK. pp. 203-210.
- Szita, I. and Lorincz, A. 2007. Learning to Play Using Low-Complexity Rule-Based Policies: Illustrations through Ms. Pac-Man. Journal of Artificial Intelligence Research. 30:659-684.
- de Boer, P. T., Kroese, D. P., Mannor, S. and Rubinstein, R. Y. 2005. A Tutorial on the Cross-Entropy Method. Annals of Operations Research. 134(1):19-67.
- Handa, H. and Isozaki, M. 2008. Evolutionary Fuzzy Systems for Generating Better Ms. Pacman Players. Proceedings of the IEEE International Conference on Fuzzy Systems, Hong Kong, China. pp. 2182-2185.
- Handa, H. 2008. Constitution of Ms. Pacman Player with Critical-Situation Learning Mechanism. Proceedings of the Fourth International Workshop on Computational Intelligence and Applications, Hiroshima, Japan. pp. 48-53.
- Albus, J. S. 1975. A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC). Journal of Dynamic Systems, Measurement, and Control Transactions of ASME. 97(3):220-227.
- Wirth, N. and Gallagher, M. 2008. An Influence Map Model for Playing Ms. Pac-Man. Proceedings of the IEEE Symposium on Computational Intelligence and Games, Perth, Australia. 228-233.
- Fitzgerald, A. and Congdon, C. B. 2009. RAMP: A Rule-Based Agent for Ms. Pac-Man. Proceedings of the IEEE Congress on Evolutionary Computation, Trondheim, Norway. pp. 2646-2653.
- Robles, D. and Lucas, S. M. 2009. A Simple Tree Search Method for Playing Ms. Pac-Man. Proceedings of the IEEE Symposium on Computational Intelligence and Games, Milano, Italy. pp. 249-255.
- Thawonmas, R. and Matsumoto, H. 2009. Automatic Controller of Ms. Pac-Man and Its Performance: Winner of the IEEE CEC 2009 Software Agent Ms. Pac-Man Competition. Proceedings of the Asia Simulation Conference, Shiga, Japan.
- Burrow, P. and Lucas, S. M. 2009. Evolution Versus Temporal Difference Learning for Learning to Play Ms. Pac-Man. Proceedings of the IEEE Symposium on Computational Intelligence and Games, Milano, Italy. pp. 53-60.
- DeLooze, L. L. and Viner, W. R. 2009. Fuzzy Q-Learning in a Nondeterministic Environment: Developing an Intelligent Ms. Pac-Man Agent. Proceedings of the IEEE Symposium on Computational Intelligence and Games, Milano, Italy. pp. 162-169.





Galván-López, E., Fagan, D., Murphy, E., Swafford, J. M., Agapitos, A., O'Neill, M. and Brabazon, A. 2010b. Comparing the Performance of the Evolvable π Grammatical Evolution Genotype-Phenotype Map to Grammatical Evolution in the Dynamic Ms. Pac-Man Environment. Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain.

Galván-López, E., Swafford, J. M., O'Neill, M. and Brabazon, A. 2010a. Evolving a Ms. Pacman Controller Using Grammatical Evolution. Proceedings of the Applications of Evolutionary Computation, EvoApplicatons: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey. pp. 161-170.

Bell, N., Fang, X., Hughes, R., Kendall, G., O'Reilly, E. and Qiu, S. 2010. Ghost Direction Detection and Other Innovations for Ms. Pac-Man. Proceedings of the IEEE Conference on Computational Intelligence and Games, Copenhagen, Denmark. pp. 465-472.

Thawonmas, R. and Ashida, T. 2010. Evolution Strategy for Optimizing Parameters in Ms Pac-Man Controller ICE Pambush 3. Proceedings of the IEEE Conference on Computational Intelligence and Games, Copenhagen, Denmark. pp. 235-240.

Matsumoto, H., Ashida, T., Ozasa, Y., Maruyama, T. and Thawonmas, R. 2009. ICE Pambush 3, <http://cswwww.essex.ac.uk/staff/sml/pacman/cig2009/ICEPambush3/ICE%20Pambush%203.pdf>. Retrieved 20 July 2012.

Oh, K. and Cho, S. B. 2010. A Hybrid Method of Dijkstra Algorithm and Evolutionary Neural Network for Optimal Ms. Pac-Man Agent. Proceedings of the World Congress on Nature and Biologically Inspired Computing, Kitakyushu, Japan. pp. 239-243.

Tong, B. K. B. and Sung, C. W. 2010. A Monte-Carlo Approach for Ghost Avoidance in the Ms. Pac-Man Game. Proceedings of the International IEEE Consumer Electronic Society Games Innovation Conference, Hong Kong, China.

Emilio, M., Moises, M., Gustavo, R. and Yago, S. 2010. Pac-mAnt: Optimization Based on Ant Colonies Applied to Developing an Agent for Ms. Pac-Man. Proceedings of the IEEE Conference on Computational Intelligence and Games, Copenhagen, Denmark. pp. 458-464.

Samothrakis, S., Robles, D. and Lucas, S. 2011. Fast Approximate Max-n Monte Carlo Tree Search for Ms Pac-Man. IEEE Transactions on Computational Intelligence and AI in Games. 3(2):142-154.

Foderaro, G., Raju, V. and Ferrari, S. 2011. A Model-Based Approximate λ -Policy Iteration Approach to Online Evasive Path Planning and the Video Game Ms. Pac-Man. Journal of Control Theory and Applications. 9(3):391-399.





Tong, B. K. B., Ma, C. M. and Sung, C. W. 2011. A Monte-Carlo Approach for the Endgame of Ms. Pac-Man. Proceedings of the IEEE Conference on Computational Intelligence and Games, Seoul, South Korea. pp. 9-15.

Ikehata, N. and Ito, T. 2011. Monte-Carlo Tree Search in Ms. Pac-Man. Proceedings of the IEEE Conference on Computational Intelligence and Games, Seoul, South Korea. pp. 39-46.

Alhejali, A. M. and Lucas, S. M. 2010. Evolving Diverse Ms. Pac-Man Playing Agents Using Genetic Programming. Proceedings of the UK Workshop on Computational Intelligence, Colchester, UK.

Alhejali, A. M. and Lucas, S. M. 2011. Using a Training Camp with Genetic Programming to Evolve Ms Pac-Man Agents. Proceedings of the IEEE Conference on Computational Intelligence and Games, Seoul, South Korea. pp. 118-125.

Dai, J. Y., Li, Y., Chen, J. F. and Zhang, F. 2011. Evolutionary Neural Network for Ghost in Ms. Pac-Man. Proceedings of the International Conference on Machine Learning and Cybernetics, Guilin, China. pp. 732-736.

Nguyen, K. Q. and Thawonmas, R. 2011. Applying Monte-Carlo Tree Search to Collaboratively Controlling of a Ghost Team in Ms Pac-Man. Proceedings of the IEEE International Games Innovation Conference, California, USA. pp. 8-11.

Sarjant, S., Pfahringer, B., Driessens, K. and Smith, T. 2011. Using the Online Cross-Entropy Method to Learn Relational Policies for Playing Different Games. Proceedings of the IEEE Conference on Computational Intelligence and Games, South Korea. pp. 182-189.

Gagne, D. J. and Congdon, C. B. 2012. FRIGHT: A Flexible Rule-Based Intelligent Ghost Team for Ms. Pac-Man. Proceedings of the IEEE Conference on Computational Intelligence and Games, Granada, Spain. pp. 273-280.

Pepels, T. and Winands, M. H. M. 2012. Enhancements for Monte-Carlo Tree Search in Ms Pac-Man. Proceedings of the IEEE Conference on Computational Intelligence and Games, Granada, Spain. pp. 265-272.

Svensson, J. and Johansson, S. J. 2012. Influence Map-Based Controllers for Ms. Pacman and the Ghosts. Proceedings of the IEEE Conference on Computational Intelligence and Games, Granada, Spain. pp. 257-264.

