Hindawi Publishing Corporation International Journal of Computer Games Technology Volume 2013, Article ID 170914, 7 pages http://dx.doi.org/10.1155/2013/170914



## Research Article

# Single- versus Multiobjective Optimization for Evolution of Neural Controllers in Ms. Pac-Man

## Tse Guan Tan, Jason Teo, and Kim On Chin

Evolutionary Computing Laboratory, School of Engineering and Information Technology, Universiti Malaysia, Jalan (UMS), 88400 Kota Kinabalu, Sabah, Malaysia

Correspondence should be addressed to Tse Guan Tan; tseguantan@gmail.com

Received 19 September 2012; Revised 14 December 2012; Accepted 4 January 2013

Academic Editor: Abdennour El Rhalibi

Copyright © 2013 Tse Guan Tan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The objective of this study is to focus on the automatic generation of game artificial intelligence (AI) controllers for Ms. Pac-Man agent by using artificial neural network (ANN) and multiobjective artificial evolution. The Pareto Archived Evolution Strategy (PAES) is used to generate a Pareto optimal set of ANNs that optimize the conflicting objectives of maximizing Ms. Pac-Man scores (screen-capture mode) and minimizing neural network complexity. This proposed algorithm is called Pareto Archived Evolution Strategy Neural Network or PAESNet. Three different architectures of PAESNet were investigated, namely, PAESNet with fixed number of hidden neurons (PAESNet\_F), PAESNet with varied number of hidden neurons (PAESNet\_V), and the PAESNet with multiobjective techniques (PAESNet\_M). A comparison between the single- versus multiobjective optimization is conducted in both training and testing processes. In general, therefore, it seems that PAESNet\_F yielded better results in training phase. But the PAESNet\_M successfully reduces the runtime operation and complexity of ANN by minimizing the number of hidden neurons needed in hidden layer and also it provides better generalization capability for controlling the game agent in a nondeterministic and dynamic environment.

#### 1. Introduction

A number of optimization solution techniques have been introduced for solving Multi-Objective Problems (MOPs) [1]. An MOP has a set of conflicting objective functions subject to certain constraints which are to be minimized or maximized [2]. Among these techniques, Evolutionary Algorithms (EAs) are particularly suited for handling MOPs [3, 4] because of its population approach that can help in finding a set of trade-off solutions in single simulation run, instead of having to perform a series of separate runs such as in the case of traditional optimization techniques. Moreover, EAs have been successfully used in solving complex problems such as discontinuities, multimodality, disjoint feasible spaces, and noisy function evaluations [5]. A large range of practical applications of Multi-Objective Evolutionary Algorithms (MOEAs) to real-life problems across a host of different disciplines can be found in the reference texts by Deb [6] and Coello et al. [3]. There are several types of effective MOEAs such as Pareto Archived Evolution Strategy

(PAES) [7], Strength Pareto Evolutionary Algorithm 2 [8], Nondominated Sorting Genetic Algorithm II [9], and Pareto-frontier Differential Evolution [10].

Generally, MOEAs are able to solve separate distinct varied dimensional optimization problems. In other words, MOEAs outperformed single-objective EAs without combining and resorting those multiple problems into single weighted-sum objective. Such weighted-sum methods are disadvantageous in obtaining suitable mode for combining different objectives into a single-objective function, which caused high-cost effective [6]. Furthermore, each evolutionary run generates single solution; the second solution will only be generated after the weights are changed; these processes will be repeated for obtaining other solutions [11, 12]. Another distinct advantage of MOEAs is its capability in generating a complete set of Pareto optimal solutions in a single run with provides users a choice of solutions for trade-off between different objectives.

On the other hand, the disadvantages of MOEAs [6, 11, 12] are (i) as the number of objectives increases, the coverage

of the Pareto front sparser and become unable to provide a comprehensive set of solutions over multiple dimensions, (ii) problematic in maintaining good spread diverse solutions along the Pareto front, and (iii) the difficulty of fitness sharing decision making in MOEAs which utilize multiple populations. This open research question provides the motivation for the work in this paper. In other words, "Is a single-objective optimization technique better than multiobjective optimization in real-life problems?"

Games are one of the common used platforms for answering research question by allowing the testing and comparison of new and experimental approaches on a challenging but well-defined problem [13–17]. In this research, Ms. Pacman has been chosen as the test-bed due to its ease of use in comparing the performances of the single-objective optimization and multiobjective optimization techniques.

In this study, a feed-forward artificial neural network (FFNN) is used and later evolved with PAES, a well-known and simple MOEA, for computer-based players to learn and optimally play Ms. Pac-man game. There are two distinct objectives to be optimized: (i) maximize the Ms. Pac-man's game scores and (ii) minimize the number of hidden neurons used in the FFNN architecture.

A comparative empirical experiment will be conducted in order to verify the performances for the methods used.

- (1) Single-objective optimization: the first experiment uses fixed number of hidden neurons in the FFNN and only maximizes Ms. Pac-man game scores, namely, PAESNet\_F.
- (2) Single-objective optimization: the second experiment is using variable number of hidden neurons in the FFNN and only maximizes Ms. Pac-man game scores, namely, PAESNet\_V.
- (3) Multiobjective optimization: the third experiment maximizes the game scores as well as minimizes the hidden neurons in the FFNN, namely, PAESNet\_M.

The main contribution of these proposed algorithms is to create computer-based agent that not only is able to make intelligent decisions like human players in the dynamic game environments, but also is highly beneficial to the real-world problems with the successful application of these techniques, such as in the application of robotics and other complex systems.

#### 2. Other Related Researches

Basically, most studies in the game of Ms. Pac-man have only focussed on hand-coded rule-based (RB) approaches or other specific methods [18–21]. Although these methods can achieve quite high scores, they are associated with some limitations. Firstly, game domains contain highly complex solution spaces that require a large number of rules in order to represent a set of all possible situations and corresponding actions in game environments. For instance, Szita and Lorincz [21] list 42 rules of a very basic hand-coded RB agent used in Ms. Pac-man game from the lists of action 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. Thirdly, 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.

The intention of this research is to create game controllers capable of general intelligent action without requiring any domain-dependent solution and also trying to be proficient in other games by just changing the input and output values of ANN. Thus, the experimental results will be compared to an appropriate reference system created by Lucas [22]. Lucas used general methods in designing the game controller which evolves ANN by using evolutionary strategy to play Ms. Pacman. 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 over 100 runs of the nondeterministic game.

## 3. Methods and Parameter Setting

This investigation has two modes of operation: training and testing as shown in Figure 1. In the training mode, the FFNNs are trained using evolution-based algorithm. The agents will learn to play many games in order to optimize weights, biases, and number of hidden neurons in FFNN architecture, as an effective mode for training. After the training process, the neural network is tested for generalization using the optimized networks.

- 3.1. Pareto Archived Evolution Strategy. The (1+1)-PAES for a two-membered PAES has been applied for simultaneously optimizing network parameters and architecture to solve single, and multiobjective optimization problems. The resulting algorithm is referred to as the PAESNet. Figure 2 shows the flowchart of PAESNet and fitness evaluation process. The strengths of PAES are listed as follows:
  - (1) simple structure;
  - (2) easy to implement;
  - (3) (1 + 1)-PAES and  $(1 + \lambda)$ -PAES are based on local search method with lower computational effort required compared to population-based MOEAs.
  - (4) a small number of parameters are needed;
  - (5) the simplest possible nontrivial algorithm capable of generating diverse solutions in the Pareto optimal set [23].

3.2. Single Objective: PAESNet with Fixed Number of Hidden Neurons and PAESNet with Varied Number of Hidden Neurons. Two systems are discussed in this section, which are the PAESNet with fixed number of hidden neuron (PAESNet\_F) and PAESNet with varied number of hidden neurons

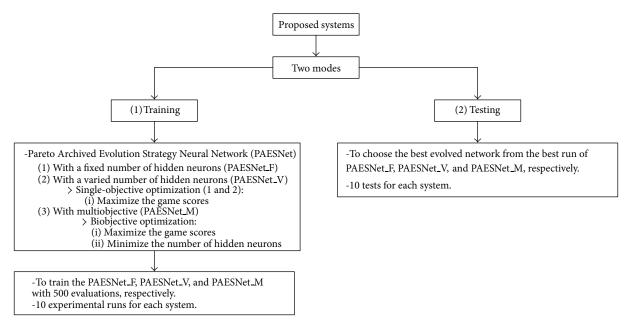


FIGURE 1: The overview of the study.

(PAESNet\_V). The default number of hidden neurons is set to 20. In the initialization phase, the ANN weights and biases are encoded into a chromosome from uniform distribution with range [-1, 1] to act as parent and its fitness is evaluated. Subsequently, polynomial mutation operator is used with distribution index = 20.0 to create an offspring from the parent and the fitness is evaluated. After that, the fitnesses of the offspring and parent are compared. If the offspring performs better than the parent, then the parent is replaced by the offspring as a new parent for the next evaluation. Otherwise the offspring is eliminated and a new mutated offspring is generated. If the parent and the offspring are incomparable, the offspring is compared with a set of previously nondominated individuals in the archive. Below is the description of the archiving process in PAESNet.

There are three possible situations that can occur between the comparison of the offspring and archive [7, 24, 25]. First, if the offspring is dominated by a member of the archive, then the offspring is discarded and a new mutated offspring is created from the parent. Second, if the offspring dominates some members of the archive, then the set of dominated members is removed from the archive. The offspring will then be added to the archive and it also becomes the parent of the next generation. Third, if the offspring and the archive members do not dominate each other, then the archive will be maintained depending on the archive size. If the archive is not full, the offspring will be directly copied to the archive. Otherwise, in the scenario that the archive is full, a neighborhood density measure is used to ensure that a well-spread distribution is maintained in the archive. If the offspring has succeeded to increase the archive diversity, it will replace the archive member in the most crowded grid location in order to maintain the maximum archive size. Note that in this third situation, the offspring and the parent are the nondominated members of the archive. The neighborhood

density measure is also applied for parent selection of the next generation from both of them. If the offspring resides in the less crowded area than the parent, then the offspring is selected.

3.3. Multiobjective: PAESNet with Multiobjective (PAESNet\_M). The structure of this proposed algorithm is similar to the algorithms in Section 3.2 except for the architecture of the ANN. In this proposed algorithms, two objectives are involved. The first objective is to maximize the game scores while the second objective is to minimize the number of neurons in the FFNN. The initial value of hidden neurons is set to 20.

3.4. Feed-Forward ANN. The typical FFNN is composed of three layers: input, hidden, and output layers [26]. The following is used to describe the feed-forward ANN architecture:

- (1) *I*, *H*, and *O* are the numbers of input neurons, hidden neurons, and output neurons, respectively;
- (2)  $\omega_{ih}$  and  $\omega_{ho}$  are the weights connecting input unit i, i = 1, ..., I, to hidden unit h, h = 1, ..., H, and from hidden unit to output unit o, o = 1, ..., O;
- (3)  $x_i$  is the input signal.

The net input of a neuron is calculated using the weighted sum of inputs from all neurons in the previous layer, as follows:

$$\sum_{i=0}^{I} \omega_{ih} x_i. \tag{1}$$

Log-sigmoid (logsig) is used as the activation function in the hidden and output layers. Based on [27], logsig has been identified as suitable activation function in ANN for creating neural-based Ms. Pac-man agent.

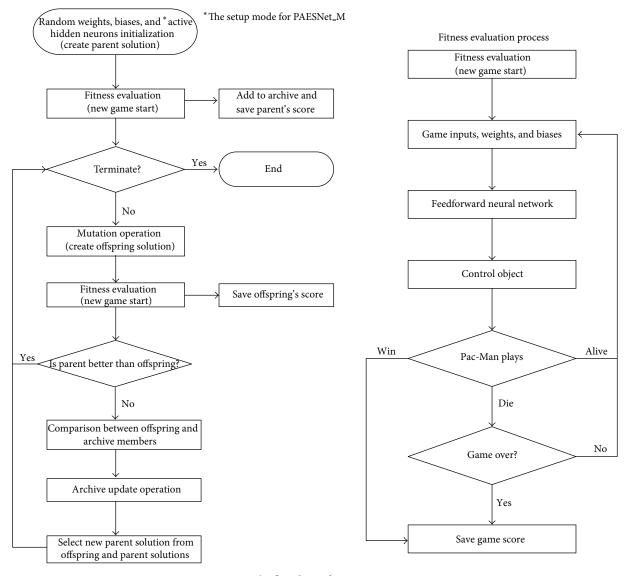


FIGURE 2: The flowchart of PAESNet optimizer.

- 3.5. Experimental Setting. The FFNN architecture of this model has a 5-20-1 structure, which consists of 5 inputs and 1 output together with one hidden layer of 20 neurons. This number of hidden neurons was suggested by Lucas [22]. The Euclidean distance is applied to calculate the distance in the maze as the inputs of the network were obtained based on the following information:
  - (1) the closest distance from agent to a pill;
  - (2) the closest distance from agent to a power pill;
  - (3) the closest distance from agent to a ghost;
  - (4) the closest distance from agent to an edible ghost;
  - (5) the closest distance from agent to a fruit.

### 4. Experimental Results and Discussions

The results obtained from the analysis of training and testing performances can be compared in the tables and figures below.

4.1. Training Results. Table 1 presents the training results of mean, standard deviation (SD), minimum (Min) and maximum (Max) values obtained from the best game scores, and number of hidden neurons in each run. The statistics (mean of scores, mean of hidden neurons) for PAESNet\_F, PAESNet\_V, and PAESNet\_M were (7161, 20), (5935, 9.7), and (5734, 8), respectively. According to the mean values of scores, the results showed that PAESNet\_F has the highest average score. However, the best scores are comparable across all three approaches (7430 in PAESNet\_F, 7190 in PAESNet\_V, and 7170 in PAESNet\_M). On the other hand, taking the mean

TABLE 1: The training results.

Run	PAESNet_F		PAESNet_V		PAESNet_M	
Kuii	Score	Neurons	Score	Neurons	Score	Neurons
1	6710	20	7190	13	7170	7
2	7050	20	5810	11	5690	10
3	7290	20	6260	11	6400	8
4	7190	20	5450	6	4940	5
5	7430	20	5190	8	4670	9
6	6930	20	5520	6	5990	6
7	7080	20	6650	11	5280	9
8	7280	20	5980	9	5260	9
9	7320	20	5500	12	6120	8
10	7330	20	5800	10	5820	9
Min	6710	20	5190	6	4670	5
Max	7430	20	7190	13	7170	10
Mean	7161	20	5935	9.7	5734	8
SD	219.47	0	612.78	2.41	739.58	1.56

Score: best score; Neurons: number of hidden neurons.

TABLE 2: Win rates for training results across all 10 runs.

Proposed algorithm	PAESNet_F	PAESNet_V	PAESNet_M
PAESNet_F	_	90%	90%
PAESNet_V	10%	_	60%
PAESNet_M	10%	40%	_

(A versus B) + (B versus A) = 100%.

values of hidden neurons, we observed that the PAESNet\_M reduces the number of hidden neurons from 20 to 8 which is around 60% improvement. This emphasizes the advantages of MOEA approach in terms of computational complexity in FFNN.

Additionally, the average scores of all the proposed algorithms, PAESNet\_F (7161), PAESNet\_V (5935), and PAESNet\_M (5734), are relatively higher when compared to Lucas (4781) [22] for training to play the Ms. Pac-man. Hence, this is further a proof that the proposed systems with PAES are able to usefully and automatically generate Ms. Pac-Man agents that display some intelligent playing behavior.

Table 2 lists the win rates (WR) for each comparison, which is the number of runs an artificial controller wins per total number of runs as shown in (2). Firstly, for PAESNet\_F versus PAESNet\_V, WR = 90%, PAESNet\_F won 9 out of 10 runs compared to PAESNet\_V except Run 1, and the result is same for PAESNet\_F versus PAESNet\_M. Subsequently, for PAESNet\_V versus PAESNet\_M, WR = 60%, PAESNet\_V won 6 out of 10 runs compared to PAESNet\_M except Run 3, Run 6, Run 9 and Run 10. The results clearly show that PAES-Net\_F outperformed the other two competing approaches. This result may be explained by the fact that PAESNet\_F is concerned with a single-objective of maximizing the game scores, while that of the PAESNet\_M is to find the set of trade-off solutions between the scores and number of hidden neurons. The acceptance of trade-off solutions is due to convergence performance and diversity preservation in Pareto



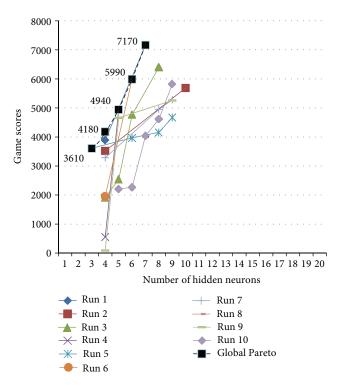


FIGURE 3: The global Pareto-frontier solutions obtained across all 10 runs using multiobjective optimization.

optimal front. Due to these two criteria, the multiobjective optimization is harder than single-objective optimization. Another possible explanation for this is that multiobjective optimization is dealing with two search spaces, which are decision variable space and objective space compared to single-objective optimization just involving one search space (decision variable space). This factor may influence the performance of PAESNet\_M:

$$WR = \frac{\text{runs won}}{\text{total of runs played}} * 100\%.$$
 (2)

The global Pareto-frontier solutions obtained with the goal of maximizing scores and minimizing hidden neurons across all 10 runs using multiobjective optimization are illustrated in Figure 3. The global Pareto solutions are shown by the dotted line. As can be seen from the figure, the PAESNet\_M reported significantly decreases the number of hidden neurons needed from 20 to the range of 3 to 7 nodes in the hidden layer as the optimized networks and the game scores achieved were 3610, 4180, 4940, 5990, and 7170.

4.2. Testing Results. After the training phase, the best evolved networks were used to test the generalization ability of the models in order to score as high as possible. The selected best numbers of neurons in the hidden layer are 20, 13, and 7 for PAESNet\_F, PAESNet\_V, and PAESNet\_M, respectively, as the optimum networks, as shown in Table 3.

TABLE 3: The best evolved networks from proposed algorithms.

Proposed algorithm	Number of hidden neurons
PAESNet_F	20
PAESNet_V	13
PAESNet_M	7

TABLE 4: The testing results.

Run	PAESNet_F	PAESNet_V	PAESNet_M
1	3090	4160	4250
2	2720	2740	2700
3	2020	2590	2890
4	4170	2100	5360
5	2500	2280	2510
6	2580	1080	2760
7	2780	4020	2860
8	1990	4820	2080
9	2090	2290	3130
10	4360	4880	3950
Min	1990	1080	2080
Max	4360	4880	5360
Mean	2830	3096	3249
SD	836.62	1286.48	983.04

TABLE 5: Win rates for testing results across all 10 runs.

Proposed algorithm	$PAESNet\_F$	$PAESNet_{-}V$	PAESNet_M
PAESNet_F	_	30%	20%
PAESNet_V	70%	_	40%
PAESNet_M	80%	60%	

(A versus B) + (B versus A) = 100%.

Table 4 presents the testing results of the three proposed algorithms. We can observe that both of the max and mean scores in PAESNet\_M (5360, 3249) were higher than PAESNet\_F (4360, 2830) and PAESNet\_V (4880, 3096). Based on mean values, PAESNet\_M was shown to have better performance compared to PAESNet\_F and PAESNet\_V.

Table 5 lists the win rates for each comparison. Firstly, for PAESNet\_F versus PAESNet\_V, WR = 70% and PAESNet\_V won 7 out of 10 runs compared to PAESNet\_F, except Run 4, Run 5, and Run 6. Next, for PAESNet\_F versus PAESNet\_M, WR = 80% PAESNet\_M won 8 out of 10 runs compared to PAESNet\_F, except Run 2 and Run 10. Lastly, for PAESNet\_V versus PAESNet\_M, WR = 60% and PAESNet\_M won 6 out of 10 runs compared to PAESNet\_V, except Run 2, Run 7, Run 8, and Run 10. From this data, we can see that PAESNet\_M resulted in the highest value of win rate compared to the two algorithms. The PAESNet\_M successfully found the appropriate network architecture and parameters by maximizing the game scores and minimizing the hidden neurons. Overall, the testing results have shown that FFNNs and PAES have strong potential for controlling game agents in the game world.

#### 5. Conclusions

In this study, the FFNN is evolved with the PAES MOEA for the computer player to automatically learn and optimally play the game of Ms. Pac-man which is called PAESNet. Three forms of PAESNet, PAESNet\_F, PAESNet\_V, and PAESNet\_M, were introduced to solve single- and multiobjective optimization problems and compared to each other in the training and testing processes. The Pareto optimal front resulted from each MOEA run provided a set of NNs which maximized the scores of Ms. Pac-man and at the same time minimized the size of the controller. In the training process, PAESNet\_F outperformed PAESNet\_V and PAESNet\_M. However, in the testing process, PAESNet\_M outperformed the other two algorithms. One of the most significant findings to emerge from this study is that the generalization performance of the neural networks could improve significantly by evolving the architecture and connection weights (including biases) synchronously via a MOEA approach as opposed to fixing the network architecture and optimizing the scoring component only using a single-objective optimization approach.

## Acknowledgment

This research is funded under the Science Fund Project SCF52-ICT-3/2008 granted by the Ministry of Science, Technology and Innovation, Malaysia.

#### References

- [1] C. Zheng and P. Wang, "Application of flow and transport optimization codes to groundwater pump-and-treat systems: Umatilla Army Depot, Oregon," Tech. Rep., University of Alabama, Tuscaloosa, Ala, USA, 2001.
- [2] C. A. C. Coello, "Evolutionary multi-objective optimization: a critical review," in *Evolutionary Optimization*, R. Sarker, M. Mohammadian, and X. Yao, Eds., pp. 117–146, Kluwer Academic, Boston, Mass, USA, 2002.
- [3] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, New York, NY, USA, 2007.
- [4] W. Q. Ying, Y. X. Li, C. Y. S. Phillip, Y. Wu, and F. H. Yu, "Geometric thermodynamical selection for evolutionary multiobjective optimization," *Chinese Journal of Computers*, vol. 33, no. 4, pp. 755–767, 2010.
- [5] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, pp. 1–16, 1995.
- [6] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, New York, NY, USA, 2001.
- [7] J. D. Knowles and D. W. Corne, "The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the IEEE Congress on Evolution*ary Computation (CEC '99), pp. 98–105, Washington, DC, USA, July, 1999.
- [8] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," Tech. Rep. 103, Computer Engineering and Network Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

- [9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [10] H. A. Abbass, R. Sarker, and C. Newton, "PDE: a Paretofrontier differential evolution approach for multi-objective optimization problems," in *Proceedings of the IEEE Conference* on Evolutionary Computation, pp. 971–978, May 2001.
- [11] C. A. C. Coello, G. T. Pulido, and E. M. Montes, "Current and future research trends in evolutionary multiobjective optimization," in *Information Processing with Evolutionary Algorithms: From Industrial Applications to Academic Speculations*, M. Grana, R. Duro, A. d'Anjou, and P. P. Wang, Eds., pp. 213–231, Springer, London, UK, 2005.
- [12] C. A. C. Coello, "Recent trends in evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds., pp. 7–32, Springer, London, UK, 2005.
- [13] K. T. Chang, K. O. Chin, J. Teo, and A. M. Jilui-Kiring, "Evolving neural controllers using GA for warcraft 3-real time strategy game," in *Proceedings of the IEEE 6th International Conference* on Bio-Inspired Computing: Theories and Applications, pp. 15–20, September 2011.
- [14] K. T. Chang, J. H. Ong, J. Teo, and K. O. Chin, "The evolution of gamebots for 3D first person shooter (FPS)," in *Proceedings of* the IEEE 6th International Conference on Bio-Inspired Computing: Theories and Applications, pp. 21–26, September 2011.
- [15] K. T. Chang, J. Teo, K. O. Chin, and B. L. Chua, "Automatic generation of real time strategy tournament units using differential evolution," in *Proceedings of the IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, pp. 101–106, October 2011.
- [16] C. H. Ng, S. H. Niew, K. O. Chin, and J. Teo, "Infinite mario bross AI using genetic algorithm," in *Proceedings of the IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, pp. 85–89, October 2011.
- [17] J. H. Ong, J. Teo, and K. O. Chin, "Interactive evolutionary programming for mobile games rules generation," in *Proceedings of the IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, pp. 95–100, Semenyih, Malaysia, October 2011.
- [18] A. Fitzgerald and C. B. Congdon, "RAMP: a rule-based agent for Ms. Pac-man," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 2646–2653, Trondheim, Norway, May 2009.
- [19] E. Galván-López, J. M. Swafford, M. O'Neill, and A. Brabazon, "Evolving a Ms. Pacman controller using grammatical evolution," in *Proceedings of the Applications of Evolutionary Computation, EvoApplications: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC*, pp. 161–170, Istanbul, Turkey, April 2010.
- [20] D. Robles and S. M. Lucas, "A simple tree search method for playing Ms. Pac-man," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG '09)*, pp. 249–255, September 2009.
- [21] I. Szita and A. Lorincz, "Learning to play using low-complexity rule-based policies: illustrations through Ms. Pac-man," *Journal of Artificial Intelligence Research*, vol. 30, pp. 659–684, 2007.
- [22] S. M. Lucas, "Evolving a neural network location evaluator to play Ms. Pac-man," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, pp. 203–210, Essex, UK, April 2005.

- [23] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [24] L. T. Bui and S. Alam, "An introduction to multi-objective optimization," in *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*, L. T. Bui and S. Alam, Eds., pp. 1–19, IGI Global, Hershey, Pa, USA, 2008.
- [25] T. Wong, P. Bigras, and K. Khayati, "Causality assignment using multi-objective evolutionary algorithms," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 36–41, October 2002.
- [26] D. Svozil, V. Kvasnička, and J. Pospíchal, "Tutorial: introduction to multi-layer feed-forward neural networks," *Chemometrics* and Intelligent Laboratory Systems, vol. 39, no. 1, pp. 43–62, 1997.
- [27] T. G. Tan, J. Teo, and P. Anthony, "Comparative investigation of non-linear activation functions in neural controllers for searchbased game AI engineering," *Artificial Intelligence Review*, 2011.

















Submit your manuscripts at http://www.hindawi.com























