Automatic Generation of Real Time Strategy Tournament Units using Differential Evolution

Chang Kee Tong, Chin Kim On, Jason Teo, Chua Bih Lii
Evolutionary Computing Laboratory
School of Engineering and Information Technology
Universiti Malaysia Sabah
Sabah, Malaysia
changkeetong@gmail.com, kimonchin@gmail.com, jtwteo@ums.edu.my, bihlii@ums.edu.my

Abstract— **This paper demonstrates the research results obtained for the application of Differential Evolution (DE) algorithm in a well known real time strategy game, namely Warcraft 3. The DE algorithm is one of the global optimizers that commonly used in solving real-time problems. In this work, the DE algorithm is combined with the conventional feed-forward artificial neural network in optimizing the solutions. The DE acts as an optimization technique used during evolution whilst the neural network operates as the controller in deciding the unit that should be spawned for mimicking the computer AI. The experimentation results show a group of mixed randomized opponent from a larger food limit can be defeated by the generated AI army using DE. Thus, it proofs that the DE used has successfully tuned the neural weights which acts as controllers in this tournament game. Furthermore, the generated controllers could decide the best units that should be spawned in defeating the opponent.**

Keywords- Artificial Intelligence (AI); Artificial Neural Networks (ANNs); Differential Algorithm (DE); Real-Time Strategy Game (RTS); Warcraft 3.

I. INTRODUCTION

Differential Evolution (DE) is proposed by Stone and Price during 1995 [1]. Since then, DE as a branch of evolutionary computing had shown promising outcome in many applications such as neural networks learning [2][3], multiprocessor synthesis [4], optimization of dynamic system [5], estimation of heat transfer [6], optimization design of heat exchangers [7], optimization and synthesis of heat integrated distillation system [8], optimize water pumping system [9], network design optimization [10], optimizing sensor [11], etc. Recently, Boryczka and Juszczuk used DE as an approximation algorithm in selecting optimal strategy in playing zero-sum game and the authors showed DE can generate the likely optimal mixed strategies for large zero-sum game [12]. Zero-sum game is a mathematical game where a player gaining is only from another player's lost [12]. In other cases, DE also presenting promising results in neural network learning. Slowik and Bialko uses DE to train artificial neural network named as differential evolution – artificial neural network training (DE-ANNT) and compared DE with evolutionary algorithm – neural network training (EA-NNT), as well as Levenberg-Marquardt algorithm (LM), and back-propagation method (EBP) [13]. The experimentation results showed DE-ANNT show promising result by comparing to LM and DE-ANNT can train a better artificial neural network (ANN) comparing to EA-NNT and EBP. Furthermore, Abdul-Kader investigates DE with three different structures of ANN: multilayer perception (MLP), radial basis function network (RBF), and feed forward artificial neural network (FFANN) in weather prediction inside some Egyptian town [14]. The outcome shows that DE with FFANN is the most accurate approach in weather forecasting. Furthermore, Li, Wang, Jiao, and Han used an Improved Déferential Evolution (IDE) with Radial Basis Function (RBF) neural network in identification of thermal process. Their simulation results showed the improved RBF neural network that was used to identify thermal complex objects will get greater identification effectiveness when IDE algorithm was used to determine the optimum parameters [15]. DE and ANN had been applied in many areas with promising results but still unsighted in gaming Artificial Intelligent (AI), especially in RTS game.

Games have long been chosen as testing platform for AI due to its competitive and dynamic environment. It is a respectable and challenging research area in computer science as the complexity of designing gaming AI is much more interesting and difficult compared to other real life problems [16]. AI is an important element for game. It makes the game more interesting by providing challenging game-play or controlling non-player character. Thus, this motivates the investigation for potential of DE in generating a neural controller for real-time strategy game, Warcraft 3 (custom map). In this research, the Warcarft 3 is used as a RTS test bed for its popularity as one of the multi-billion seller RTS games [17] [18] and it shares the common characteristic of nowadays RTS game such as several choices of races which having their own special units, building and bonus features. Warcraft 3 is also being utilized in investigating on improving micromanagement of RTS game with case-based reasoning system and in resulting the proposed approach is capable to defeat the build in game AI nevertheless, it still out smarts by human players [19].

In this study, a real time environment is provided by a custom made Warcraft 3 map. There are two forces of battle unit involved in each game: (1) randomized opponent battle unit, namely RO-AI and (2) generated neural controller battle unit, namely DE-NN-AI. The DE-NN-AI used is a combination of DE and FFANN during the optimization stage. The conventional DE is applied to tune the weights of the FFANN where FFANN works as a decision mechanism on what battle unit to spawn for competing over the opponent unit. The research objective is to generate an optimum group of battle unit that extremely superior that can defeat a huge opponent unit by having a maximum remain units after the game. The research is focused on the human units during its games play as a basic research scope.

II. METHOD

A. Feed Forward Artificial Neural Networks

FFANN has been well known for its extremely fast and accurate performance in gaming and robotics area [13][14][16][20]. The researchers have proven highly promising results could be generated with the FFANN used. It has been chosen and used in this research as it is simple, easy to understand, easy to develop and less time consuming during the optimization processes as it involved very little and simple layer of neural structure.

Throughout the literature reviewed as presented in the previous section, the combination of DE with ANN has been proven to be useful in generating optimal solutions for real time problems such as discussed in [13][14][20]. The basic structure of the ANN consists of a minimum of three layers: (1) input layer, (2) hidden layer and (3) output layer [20].

In this paper, the FFANN plays as two important roles. The FFANN is used as the neural controller for (1) deciding the type of unit and (2) number of unit to be spawned during the games play. It is important to decide the type of unit to be spawned during the games play as each different unit is having different damage level and attack skill whilst the number of unit to be spawned is the key to mimic the RO-AI. The FFANN used in this study also consists of three layers. Each layer consists of number of nodes and the nodes are connected to other layer with forwarded connectionist [20]. The input of the FFANN is the spawned RO-AI unit in each match and the output from the FFANN represents the unit to be spawned for the DE-NN-AI. The bias value of the NN used is +1. Each of the weight of the FFANN will be represent with real number which later will be the chromosomes for evolutionary stage. Fig. 1 shows the structure of the FFANN that used in this study.

B. Differential Evolution

DE is a population based optimizing method and branch of evolutionary computing presented by [1]. In this study, DE is used to evolve the weight of FFANN for solving single objective problem in generating the corresponding units to defeat a group of RO-AI unit. DE is different comparing to other evolutionary techniques because DE uses different number of parents in generating next population. The DE algorithm used is shows as below.

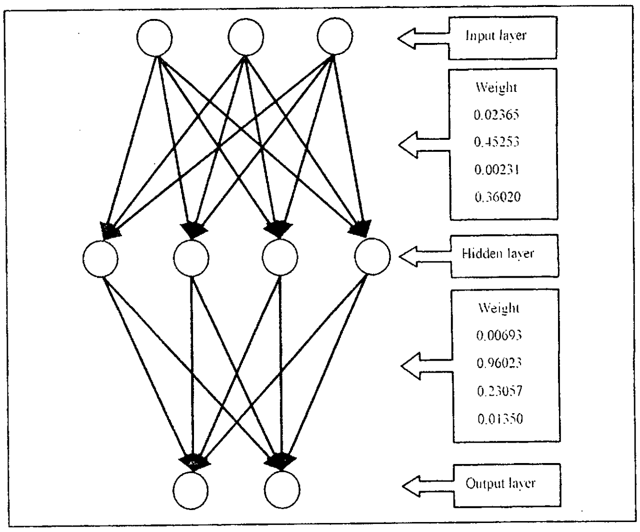


Figure 1. Structure of FFANN.

1.0 Start.

2.0 Initialization of population (10 chromosomes/FFANN weights) by generating random real value N(-1,1) which represents the weight of the FFANN.

3.0 Loop

3.1 Randomly generate enemy unit until food limit is reached which is then used as input for FFANN.

3.2 An Individual (chromosome / weights) is loaded into the FFANN and the output is computed. After that, the generated output is decoded into integer values that representing the amount of units that should be spawned for each type of unit.

3.3 Game starts after all units for both sides had been spawned by ordering all units to attack-move to the centre of the customized Warcraft 3 map.

3.4 Game ends once either one side of food reached zero (all units killed) or if the life of the game has elapsed 180 second. Then, fitness of the individual is calculated and stored in an archive. The remained units are cleared away from the ground.

3.5 Step 3.1, 3.2 and step 3.3 are repeated until all individuals of current population are evaluated.

4.0 Select three parents (P1, P2 and P3) with tournament selection.

4.1 If it is not the first generation of the optimization process, compare among all parents stored in the archive. Choose three outperformed parents for further used in the evolution phase.

4.2 Otherwise, store the selected optimal parents to the archive.

5.0 Loop

5.1 Crossover P1, P2 and P3 with some uniform (0, 1) to create a new offspring as below:

If within crossover rate,
$W_i^{child} = W_i^{P1} + N(0,1)(W_i^{P2} - W_i^{P3})$
else
$W_i^{child} = W_i^{P1}$
Endif

5.2 Mutate with some uniform (0, 1) as below: $W_i^{child} \leftarrow W_i^{P1} + N(0, mutation\_rate)$

5.3 Add offspring into next population.

5.4 Step 5.1, 5.2 and 5.3 are repeated until offspring of next generation reaches 10.

6.0 Eliminate all individual in the current population.

Step 3, 4, 5, and 6 are repeated until reached 200 generations.

III. WARCRAFT 3 (JASS)

Warcraft 3: The Frozen Throne is one of the Warcraft latest series produced by Blizzard entertainment. In this research, Warcraft 3 can provide the required environment for testing our neural controller that is dynamic, complex and imperfect information in a three dimensional (3D) environment. Other than that, Warcraft 3 also meet the feature of current RTS game by having multiple races with own special units, buildings and technology. Warcraft 3 provides four races of character: (1) human that having high armour for units and buildings (2) orc with highly strength hit points units, (3) undead that quick in unit production, and (4) nightelf with offensive unit. The map of Warcraft 3 can be customized with programming language call Jass by using Jass Editor. In this study, JassCraft is used as the editor to modify the war3map.j for a map as the test bed. As previously mentioned, this study focused in human race unit. We believed that the generated controllers could mimic any type of randomized RO-AI unit as the human unit is strengthened by the armour and hence the generated optimal units could easily defeat other type of RO-AI races. The details of the units that used in the experiment are shown in Table 1 and Table 2, respectively.

TABLE 1. HUMAN UNITS' STATS

Name	Hit Point	Att-ack	Attack Range	Attack Type
Footman	420	12.5	Melee	Normal
Rifleman	505	21	Range	Pierce
Knight	835	34	Melee	Normal
Spell Breaker	600	14	Range	Normal
Flying Machine	200	7.5	Range	Pierce
Mortar Team	360	58	Range	Siege
Gryphon Raider	825	50	Range	Magic
DragonHawk Raider	575	19	Range	Pierce

Table 1 and Table 2 clearly show the varieties of the units in human race of Warcraft 3. Knight is the strongest ground unit with equipped extremely high armor nevertheless it skill has been limited to apply for ground unit. Gryphon raider is the strongest air unit but skilled with very light of defensive characteristic. Mortar team is a special unit that having siege attack in damaging a group of units in a small effective area. However, it is limited to only target for ground unit and it is having lightly defensive skill. Unlike other game, each unit used in Warcraft 3 is having its own unique capability and limitation and hence a tournament could not easily to be defeated by huge group of army used. In such a case, a group of 15 footmen could be easily mimicked by only 5 knights. In other case, one footman could easily mimic at least two riflemen. The combination of units used during games play is the key for the winning rather than depending on number of units used.

TABLE 2. HUMAN UNITS' STATS

Name	Ar-mour	Armou r type	Unit Type	Fo-od
Footman	2	Heavy	Ground	2
Rifleman	0	Mediu m	Ground	3
Knight	5	Heavy	Ground	4
Spell Breaker	3	Mediu m	Ground	3
Flying Machine	2	Heavy	Air	1
Mortar Team	0	Heavy	Ground	3
Gryphon Raider	0	Light	Air	4
DragonHawk Raider	1	Light	Air	3

IV. EXPERIMENTAL SETUP

This experiment uses a 20x30 Warcraft 3 custom map. DE and FFANN are developed in war3map.j using JassCraft, a Jass editor. The war3map.j is then injected to the map by MPQMaster. Fig. 2 shows the map created on WorldEdit.

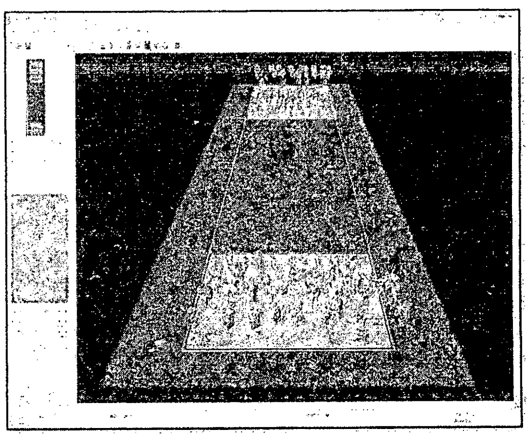


Figure 2. Map on WorldEdit.

The map generated in Fig. 2 shows there are two groups of human race units used during the games play, one group is located on the top side and another group is positioned at the bottom side of the map. The DE-NN-AI units are spawned on the bottom side of the map whilst the RO-AI units are spawned and located on the top side of the map. Furthermore, the spell casters units and heroes units are switched off and so with the research technology and upgrade bonus. As previously mentioned, the units used in this experiment were limited only for human races and the micromanagement of each unit was not included.

There are 10 set of experiments conducted with crossover rate 0.7 and mutation rate 0.02. Food limit for DE-NN-AI unit is prefixed to 100 and the food limit for RO-AI is assigned to 120. The food limit is purposely prefixed differently in order to increase the difficulty of the games play. We hope the combination of FFANN and DE could generate a group of less AI unit that able to defeat a larger group of opponent. The tournament is begun once all of the units of both sides have been spawned by ordering all units to attack-move to the centre of the map. The tournament is considered ended by two situations (1) food limit reached zero for either one sides (all units have been mimicked) or (2) the match reached 180 seconds. The winner comes from the group that having highest number of food limit remained during the games play. A timer is included during the games play as there was a limitation found during some strange games play. In a match, it might happen with remaining units for both sides were unable to attack each other since some of the units were limited to their own skills. As an example, it might happen where there were three knights remained for RO-AI and two flying machines remained for DE-NN-AI. Both RO-AI units and DE-NN-AI were unskilled to attack each other. Hence, a timer has been included in order to solve the limitation.

V. TEST FUNCTION

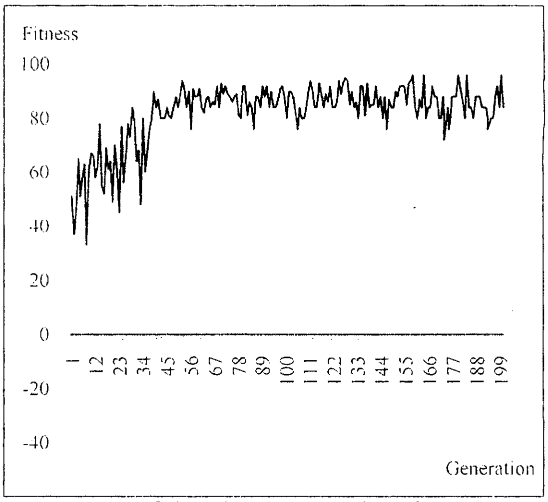
The objective of this experiment is to win the game in competing with a stronger group of RO-AI unit. Highest number of remaining unit is the key to be the winner for the games play. Hence, the test function used involved the food limit characteristic. The proposed fitness function is listed as below.

$$F_i = F_{Ui} - F_{Ei} \qquad (1)$$

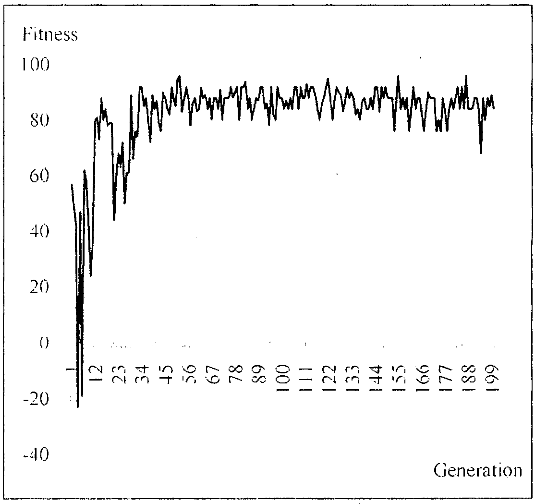
F_i represents fitness value of an individual. F_{Ui} represents remaining food of DE-NN-AI and F_{Ei} represents remaining food of RO-AI. A positive value of F_i means DE-NN-AI is winning the match. Otherwise, the DE-NN-AI is losing the battle if F_i of the round is negative.

VI. RESULT AND DISCUSSION

There were 10 set of experiments conducted in this study. Each experiment involved an optimal of 200 generation runs during the optimization processes. The experimentation results showed all 10 generated controllers performed very well and the generated DE-NN-AI could defeat the RO-AI in all runs. Fig. 3 below shows two out of all 10 experimentation results obtained with highest average fitness scored during the optimization stage. Fig. 3.(a) shows from 1st generation to 45th generation, the scores keep increasing from 40 to 80 and the scores reached an average of 85 score until its last generation. During early stage of the first few generations, a good solution has been successfully obtained and maintained during the optimization stage as the elitism with archive has been applied in order to avoid of losing good solutions. The generated good solution was formed a balanced number of ground and air units for our DE-NN-AI which consists of few riflemen, footmen and knights. Nevertheless, during generation 30th to 45th, that particular optimal solution had decided to change the strategy in spawning more air units which were gryphon raider, flying machine and dragonhawk raider rather than the a combination of ground and air units. It happened as the DE-NN-AI found that it could easily mimic against the RO-AI with more air units used.



(a) Collected results on experiment 8



(b) Collected results on experiment 6

Figure 3. Experimentation results.

Furthermore, it was harder for RO-AI to mimic the DE-NN-AI air units due to some RO-AI ground units were limited their skill in fighting against the DE-NN-AI air units. At last, the optimum solution found was formed by a group of gryphon raider without other type of units support. It happened as the DE-NN-AI found that (1) there were more units remained with gryphon raider used during the games play, (2) it was easier to mimic the RO-AI with only air unit used, (3) the flying machine was not require as it equipped less hit point and low attack power. This experiment has clearly shows the DE-NN-AI learned some behaviors (1) minimizing ground units used during games play as some ground units were limited their capability to attack air units even they were equipped with high hit point, high attack power and higher armor, (2) decision making in remaining and maximizing the number of best units rather than combine a set of different units, and (3) minimizing the time taken to mimic the RO-AI even this feature was not previously included in the test function.

Fig. 3.(b) shows almost similar situation happened as discussed in Fig. 3.(a). From Fig. 3.(b), it clearly depicts the solution found was capable to defeat the RO-AI even the combination unit in RO-AI was different for every generation. The generated DE-NN-AI lost two matches during early stage of the evolution (generated less than zero fitness scores) but the AI learned quickly to generate different combination of ground and air units to mimic the RO-AI. Nevertheless, the DE-NN-AI unit lost against in battle during 21^{st} generation. With the assistant of DE algorithm, the DE-NN-AI started to spawn more air units rather than balanced ground and air units and the DE-NN-AI again mimicked the RO-AI after generation 22^{nd}. The DE-NN-AI on the go to generate better solution and lastly came out with a great solution on generation 36^{th} onward until end of the optimization life. The optimum DE-NN-AI was formed with all dragonhawk unit compared to gryphon raider in the previous discussion. Even both controllers were generated all air units, interestingly the solution obtained in experiment 6 generated slightly higher fitness score compared to experiment 8, 81.915 and 82.075 respectively. This is something interesting as gryphon raider units used should generate higher average fitness score because gryphon raider always equipped with highest hit point as well as extremely high attack power compared to dragonhawk as shown in Table 1. Different level of equipped armor might be is the reason of scored less from gryphon raider.

In other case, it clearly shows that the elitism DE-NN-AI used has successfully maintained the good solution during the optimization stage. Fig. 4 shows the evidence with the conducted experiment scored 52.085 average fitness score and won during the games play. Typically, the average fitness score obtained in the early stage of the experiment was around 60 with a group of combined major rifleman units and some ground and air units.

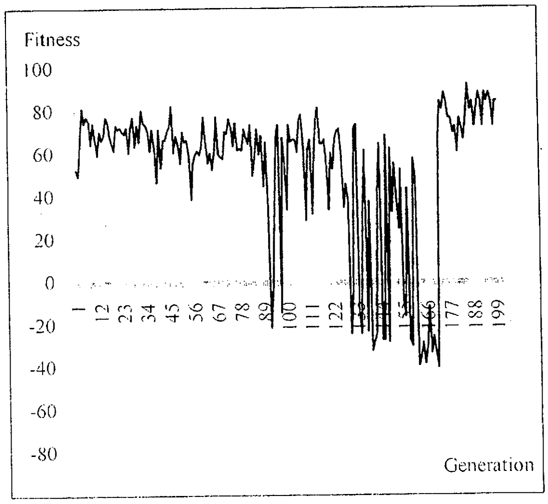


Figure 4 Collected results on experiment 3.

Nevertheless, the DE-NN-AI lost against the battle during generation 92^{nd}, 93^{rd} and 97th as the spawned RO-AI units were majored in very strong ground units such as knights, footman and mortal team. Then, the solution was changed to spawn more air units for the DE-NN-AI team. However, the DE-NN-AI units lost again for almost all battles during generation 130^{th} - 165^{th}. It happened as the RO-AI generated lots of rifleman and knights to counter with the DE-NN-AI unit. The RO-AI knights killed the DE-NN-AI ground units just within few seconds of the games play as the DE-NN-AI units were formed by a large group of riflemen and mortal team that having very less or armor ability whilst the RO-AI rifleman mimic the DE-NN-AI air units. The generated solutions were changing during the optimization stage until generation 170 onwards, at last, the optimal solution found. A group of gryphon raider took place.

In an overall, the average generation for reaching early convergence is 80.4. The average fitness scored is 73.719 and the average maximum fitness is 96.6. The overall experimentation results were tabulated as in Table 3.

TABLE 3 OVERALL EXPERIMENTATION RESULTS.

Experiment	Generation to Converge	Average fitness	Maximum Fitness
1	79	76.775	98
2	128	65.665	96
3	173	52.085	92
4	72	77.995	99
5	66	73.995	100
6	33	82.075	96
7	118	75.380	96
8	39	81.915	96
9	35	78.745	100
10	61	72.555	93
Average	80.4	73.719	96.6

Most of the experiments conducted were ended up with a group of air units of gryphon raider except in experiment 6 as discussed previously, dragonhawk raider took the place. Obviously, a group of gryphon raider is the global solution for this tournament fighting. The gryphon raider is considered as a strong unit among all other human race units. Gryphon raider is equipped with second highest hit point and second highest attack power even with zero defensive skill. Furthermore, gryphon raider is immunity to most of the ground units such as footman, knight, spell breaker, and mortar team.

VII. CONCLUSION

The experimentation results have clearly showed the optimal game controllers could be obtained through the hybridization of FFANN and DE. The generated controllers were able to choose which unit to be spawned to encounter with a stronger group of enemy. From the results obtained, it clearly shows the DE-NN-AI learned some behaviors (1) minimizing ground units used during games play as some ground units were limited their capability to attack air units even they were equipped with high hit point, high attack power and high armor, (2) good decision making in remaining and maximizing the number of best units rather than combine a set of different units, and (3) minimizing the time taken to mimic the RO-AI even this feature was not previously included in the test function.

VIII. FUTURE WORK

The inclusion of different races in unit can be tested with FFANN with DE controller. Micromanagement is always a great issue in gamming AI. It may affect the outcome of the controller. Furthermore, the inclusion of multi-objectives concept such as minimizing resources or minimizing food required during the optimization processes maybe able to generate better controllers during games play. Other studies such as incremental learning, co-evolutionary, interactive evolution strategy, etc could be applied to generate better controllers.

ACKNOWLADGEMENT

This research work is funded under the project FRG0221-TK-1/2010 granted by the Ministry of Higher Education, Malaysia.

REFERENCES

[1] R. Storn, and K. Price, "Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report TR-95-012, ICSI, http://http.icsi.berkeley.edu/-storn/litera.htm.

[2] J. Ilonen, J.-K Kamarainen, J. Lampinen, "Differential Evolution Training Algorithm for Feed-Forward Neural Networks", Neural Processing Letters 7, 1, pp.93-105, 2003.

[3] G.D. Magoulas, V.P. Plagianakos, M.N. Vrahatis, "Neural Network-Based Colonoscopic Diagnosis using On-Line Learning and Differential Evolution", Applied Soft Computing 4(4), pp. 369–379, 2004.

[4] A. Rae, S. Parameswaran, "Application-specific heterogeneous multiprocessor synthesis using differential-evolution", 11th International Symposium on System Synthesis, pp. 83–88, December 1998.

[5] B.V. Babu, and K. Gautam, "Evolutionary Computation for Scenario-Integrated Optimization of Dynamic Systems", Proceedings of International Symposium & 54th Annual Session of IIChE (CHEMCON-2001), CLRI, Chennai, December 19-22, 2001.

[6] B.V. Babu, and K.K.N. Sastry, "Estimation of Heat Transfer Parameters in a Trickle Bed Reactor using Differential Evolution and Orthogonal Collocation", Computers and Chemical Engineering, Vol. 23 (No. 3), pp. 327-339, 1999.

[7] B.V. Babu and S.A. Munawar, "Differential Evolution for the Optimal Design of Heat Exchangers", Proceedings of All India Seminar on "Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond", organized by IE (I), Orissa State Centre Bhuvaneshwar, 13 March, 2000.

[8] B.V. Babu, and Rishindra Pal Singh, "Optimization and Synthesis of Heat Integrated Distillation Systems Using Differential Evolution", Proceedings of All India Seminar on "Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond", organized by IE (I), Orissa State Centre Bhuvaneshwar, 13 March, 2000.

[9] B.V. Babu, and Rakesh Angira, "Optimization of Water Pumping System Using Differential Evolution Strategies", Proceedings of The Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS-2003), Singapore, pp. 25-30, 15-18 December 2003.

[10] S. Mendes, J. Gomez-Pulido, M. Vega-Rodriguez, M. Pereira, J. Sanchez-Perez, "Fast Wide Area Network Design Optimisation using Differential Evolution", AdvComp 2007 – International Conference of Advanced Engineering Computing and Applications in Sciences, Papeete, French Polynesia, Tahiti, November 2007.

[11] R. Joshi and A. Sanderson, "Minimal Representation Multisensor Fusion Using Differential Evolution.", IEEE 0-8186-8138-1, 1997.

[12] U. Boryczka, P. Juszczuk "Comparative Study of the Differential Evolution and Approximation Algorithms for Computing Optimal Mixed Strategies in Zero-sum Games", the 2nd International Conference on Computational Collective Intelligence -Technology and Applications, Springer-Verlag in series LNCS/LNAI, pp 363-372.

[13] A. Slowik, M. Bialko, "Training of Artificial Neural Networks Using Differential Evolution Algorithm",M..Human System Interactions, pp. 60 – 65, 2008.

[14] H. M. Abdul-Kader, "Neural Networks Training Based on Differential Evolution Algorithm Compared with Other Architectures for Weather Forecasting38", IJCSNS International Journal of Computer Science and Network Security, Vol. 9, No. 3, March 2009.

[15] Y. Li, D.F Wang, S.M. Jiao, P. Han,"Thermal process identification using differential evolution algorithm and radial basis function neural network", Proceedings of the Chinese Society of Electrical Engineering, Vol. 30, no. 8, pp. 110-116, 15 Mar 2010.

[16] S.M. Lucas, G. Kendall, "Evolutionary Computation and Games", Computational Intelligence Magazine, IEEE Volume 1 , Issue: 1Digital Object Identifier: 10.1109/MCI.2006.1597057Publication Year: 2006 , pp. 10 – 18.

[17] M. Buro, "RTS games as test-bed for real-time ai research", Proceedings of the 7th Joint Conference on Information Science (JCIS 2003), pp. 481-484, 2003.

[18] M. Buro, "*Call for ai research in RTS games*", Proceedings of the AAAI-04 workshop on AI in games, San Jose, pp.139-142, 2004.

[19] T. Szczepanski and A. Aamodt, "Case-based reasoning for improved micromanagement in real-time strategy games", Proceedings of the Workshop on Case-Based Reasoning for Computer Games, 8th International Conference on Case-Based Reasoning, ICCBR 2009, Seattle, pp. 139-148, July 2009.

[20] K.O. Chin, J. Teo, A. Saudi, "Multi-objective artificial evolution of RF-localization behavior and neural structures in mobile robots," Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp.350-356, 1-6 June 2008.