Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Games (CIG 2007)

Evolutionary Computations for Designing Game Rules of the COMMONS GAME

Hisashi Handa
Graduate School of Natural Science and Technology
Okayama University
Tsushima-Naka 3-1-1, Okayama 700-8530, JAPAN
handa@sdc.it.okayama-u.ac.jp

Norio Baba
Department of Information Science
Osaka Educational University
Kashihara 4-698-1, Osaka Prefecture, JAPAN
baba@cc.osaka-kyoiku.ac.jp

Abstract— **In this paper, we focus on game rule design by using two Evolutionary Computations. The first EC is a Multi-Objective Evolutionary Algorithm in order to generate various skilled players. By using acquired skilled players, i.e., Pareto individuals in MOEA, another EC (Evolutionary Programming) adjusts game rule parameters i.e, an appropriate point of each card in the COMMONS GAME.**

Keywords: Evolutionary Computation, Game Rule Design, Multi-Objective Evolutionary Algorithms

I. Introduction

In recent years, there has been an increasing interest in Gaming[1] to deal with complex problems in which human decisions have far reaching effects on others. Historically speaking, gaming has its origin in war games [2]. However, after the Second World War, it has been applied to various peaceful purposes. A large number of business games have been developed with the purpose of training students in business school [3][4]. Further, some environmental games have also been developed in order to help people consider seriously about the environmental state of the world [6][7]. Gaming has also successfully been utilized for operational purposes [5].

In order to design Gaming systems, "trial and error" have been required so far: the constitution of rules and their parameters is optimized through actual game playing in order to attain the objective of the Gaming systems. Therefore, the development of useful Gaming systems takes a lot of efforts since all the possible actions/strategies by test players and potential players is not precisely assumed by designers of such Gaming systems in advance. That is, during and after trial and error, players often take un-assumed actions or sometimes have strategies which are out of primary objective of the Gaming systems.

In this paper, we suggest that EAs could be used to improve the COMMONS GAME [6]. In particular, in order to attain this objective, we shall try to utilize Evolutionary Computations in the following two steps:

1) First, we shall utilize Multi-Objective Evolutionary Algorithms (MOEA) [9] to generate various skilled players.

2) Further, Evolutionary Programming (EP) [10][11] is used to derive appropriate combinations of the rules concerning the point of each card.

[1]Occasionally, game theory has been confused with gaming. Gaming means the use of a game for one of the various purposes such as teaching, training, operations, entertainment, etc [1][3]

II. The proposed method

A. Overview

In proper game rules, there should be no strategy, which outperforms any other strategies at any time. On the other hand, it also may be bore games such that there is no difference between any policies . Games with proper game rules keep players tense even if some strategies completely dominate other one at certain situations. Moreover, players in such games tend to think of opponent's strategy and to modify their strategies to defeat opponents.

As mentioned in previous section, trial and error is required in order to acquire such game rules. Even if we adopt some systematic approaches such as interactive evolutionary computations, a large number of plays is needed to make game excitement. Therefore, it is a burden to designers and testers of games. Thus, we have proposed multi-layer neural network approach which learns play records by humans. Moreover, by using this neural network agent, we have shown that Evolutionary Computations can design game rules. However, it still intends game designers to prepare a large number of teacher patterns for representing various strategies.

In this paper, two kinds of Evolutionary Computations are employed as depicted in Figure 1: Multi-Objective Evolutionary Algorithm and ordinal (single objective) Evolutionary Computation. The Multi-Objective Evolutionary Algorithm is for generating various skilled players. The ordinal Evolutionary Computation is for designing game rules by using acquired "various players" in the MOEA. The basic notions of them are described in the following subsections, and, the implementation of them in the case of the COMMONS GAME is mentioned in section III.

B. MOEA for generating intelligent players

In recent years, various approaches concerning the applications of Evolutionary Algorithms to the field of games have been proposed [19], [20]. Multi-Objective optimization is one of the most promising fields in Evolutionary Computation study. Due to the population search of Evolutionary Computations, Multi-Objective Evolutionary Algorithms (MOEA) can evolve candidates of Pareto optimal solutions. Hence, in comparison with conventional Evolutionary Computations,

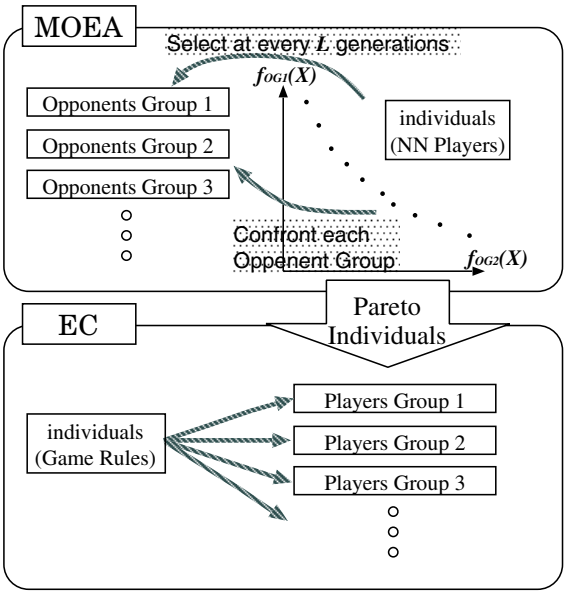

Fig. 1. Overview of the proposed framework

MOEA can simultaneously find out various solutions. In this paper, we employ NSGA-II, proposed by Deb *et al.*, to evolve game players with Neural Networks [9]. NSGA-II utilizing crowded tournament selection, and ranking method with non-dominated sort, is one of the most famous MOEA.

As depicted in Figure 1, the MOEA employed in this paper has several opponent groups, whose players are randomly copied from Pareto front set: At every L generation, some opponent groups are replaced by certain criteria, which indicates the difficulties of games. The fitness of this MOEA is defined by performance of game play, e.g., score, against each opponent group. That is, the number of objective functions in this MOEA is the same as the number of opponent groups. The genotype of individuals denotes parameters which enable individuals to perform intelligent behaviors for all the possible input/output. This paper employs Neural Networks to represent players so that the weights of the Neural Networks are used as the genotype. However, the representation of players is not restricted to Neural Networks. For instance, linear functions, table representations, sets of rules and so on, could be used to represent players and Evolutionary Computations can successfully cope with them.

C. Evolutionary Computation with evolved players for designing Game rules

After running the MOEA in the last subsection, an evolutionary computation is run to design rules of games. As delineated in Figure 1, a number of player groups, whose members are selected from Pareto set in the MOEA, is prepared in advance. Individuals in this evolutionary computation indicate rules of games, and are evaluated through game playing by the player groups.

Game rule design in this paper is just for adjusting parameters of points of cards so that the Fast Evolutionary Programming proposed by Xin *et al.* [11] is adopted. The reason of this is that 1) it is quite easy to implement, and 2) it performs well due to the Cauchy distribution mutation, which has longer tail distribution than Gauss distribution. The second reason is quite important for the game rule design by evolutionary computations because the fitness landscape for the game rule design tends to be like plateaus. Thus, long jump by the Cauchy mutation might be useful. In order to design game rule systems, evolutionary computations with much complicated individual representation, such as GP, could be used.

Individuals in Evolutionary Programming are composed of a pair of real valued vectors (X, η), where X and η indicate design variables in given problems and variance parameter used in self-adaptive mutation [10]. The Fast Evolutionary Programming is carried out in following steps [11]:

1) Generate the initialize population of μ individuals
2) Evaluate individuals
3) Each individual (parent) creates single offspring X', η' by using the following equation,

$$\begin{aligned} x'_j &= x_j + \eta_j \delta_j, \\ \eta_j &= \eta_j \exp(\tau' N(0,1) + \tau N_j(0,1)), \end{aligned}$$

   where η_j denotes j^{th} component of vector η. δ_j, $N(0,1)$, and $N_j(0,1)$ denote a Cauchy random variable, a normal distribution random number with mean zero and standard deviation one used in all the design variables, and a normal distribution random number used in a design variable j, respectively. Moreover, coefficient τ and τ' are commonly set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$, respectively.
4) Evaluate offspring
5) Conduct pairwise comparison over parents and offspring. For each individual, q opponents are chosen uniformly at random. For each comparison, the individual receives "win" if its fitness is not smaller than opponent's one.
6) Select the μ individuals, which have the most wins, from parents and offspring
7) Stop if halting criterion is satisfied. Otherwise go to Step 3.

III. Implementations

A. COMMONS GAME

The COMMONS GAME, a kind of environmental games, was developed by Powers *et al.* in 1980 [6]. Since we live in a world having only finite natural resources such as fishes and forests (commons), it is wise to consider their careful utilization. The COMMONS GAME may be quite helpful in stimulating discussion of this problem. Fig. 2 shows the layout of the original COMMONS GAME.

In the following, we give a brief introduction to this game. Six players are asked to sit around a table. Following a brief introduction of the game, the game director tells the players that their objective is to maximize their own gains. During

TABLE I
INPUT VARIABLES FOR NEURAL NETWORK PLAYERS

1.	Difference between the total points and the average points of corresponding player
2.	Rank of corresponding player
3., 4.	State of the environment
5.	Changes in the environment
6.	Round Number
7.–11.	Weighted sum of each card having been selected by all of the players
12.–16.	Weighted sum of each card having been selected by corresponding player
17.–21.	The number of each card being selected in the previous round
22.–26.	The card being selected by each player in the previous round

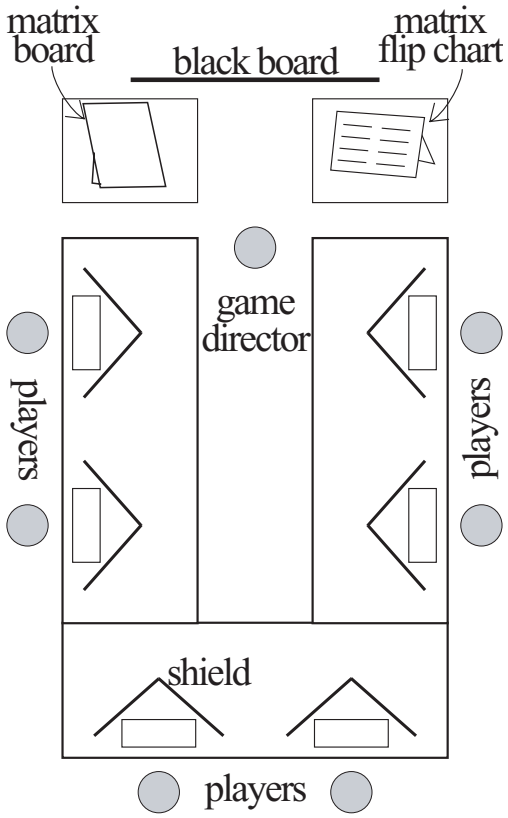

Fig. 2. Layout of the original COMMONS GAME

the first three rounds players can play only with a green or red card. After the three rounds, they are free to play one of the five cards: green, red, black, orange, or yellow. In each round, players hide their cards behind a cardboard shield to ensure individual privacy.

Each colored card has its own special meaning, and has the following effect upon the total gains of each player:

Green card
: A green card represents high exploitation of the commons: players who play a green card can get a maximum reward. However, they loses 20 points if one of the other players plays a black card in the same round.

Red card
: A red card indicates a careful utilization of the commons: Red cards are only worth about forty percent as many points as green cards.

Black card
: This card has a punishing effect on the players with green cards: Players who have played a black card have to lose 6 points divided by the number of black cards played at the same round, but are able to punish green card players by giving them -20 points.

Orange card
: Orange cards give an encouraging effect to red card players: Players who have played this card have to lose 6 points divided by the number of orange cards played at the same round but are able to add 10 points to red card players.

Yellow card
: A yellow card denotes a complete abstention from utilization of the commons: Players who play this card get 6 points.

Depending upon the players' strategies, the state of the commons change: If players are too cager to exploit the commons, then deterioration of the commons occurs. Players have a chart on the board representing the payoffs for the red and green cards under different conditions of the commons.

Although players are informed that there will be 60 rounds, each game ends after 50 rounds. After each 8^{th} round, players have a three-minute conference. They can discuss everything about the game and decide every possible way to play in future rounds. For those interested in further details, however, we recommend reading the paper written by Powers *et al.* [6].

B. MOEA for generating various skilled players

1) Coding Method: Our objective is to simultaneously construct plenty of Neural Network players with different strategies. The multi-layered neural network model adopted in this paper has 26 input units, one hidden layer with 30 units, and 5 output units. The weights between the input and the hidden layers, the weights between the hidden layer and the output layer, and the thresholds of the units in the hidden layer and the output layer are evolved by MOEA. That is, the number of genes in an individual is 965. The input values presented to this neural network model are summarized in Table I and explained the following paragraph. The units in the output layer correspond to each color. Hence, the Neural Network players select the card with the most output value at each rounds.

In order to represent the current state of the commons, two inputs, consisting of a number in the matrix board and fine-grained environmental status, are prepared. For inputs 6.–9., 5 different inputs corresponding to each color are prepared.

TABLE III

COMPARISON OF RESULTANT NN PLAYERS

	n/a	100	50	25	20
10	○	○	○	○	×
20	○	○	○	○	
25	○	◇	○		
50	×	×			
100	○				

TABLE II

EFFICIENCY OF CARDS

Player's card	$E_i(C)$	Situations
R	$+1$	No black player, but some green players
	-1	Otherwise
B	$+1$	No green player
	-1	Otherwise
G	$+1$	Some black players
	-1	Otherwise

2) Substitution of opponent groups: As mentioned in subsection II-B, some opponent groups are replaced at every L generations. Opponent groups, which are often beaten by individuals, or which tend to be monotonic games, are replaced by new opponent groups whose members are copied from the current Pareto individuals. The following three criteria, which is based on the cards played by individuals at fitness evaluation in the MOEA, are used to determine the replacement:

- the variance of the number of cards
- the total sum of the efficiency of cards (cf. Table II)
- the total points of individuals

By using the notion of "Pareto Optimality," the replacement is carried out: all the opponent groups dominated by other opponent groups are replaced by new one. Further replacements of opponent groups, chosen randomly, are also carried out if the number of such dominated opponent groups is less than half of the number of opponent groups.

3) Experimental Results: NSGA-II with 200 individuals is examined for various substitution interval $L(= 10, 20, 25, 50, 100)$. In addition, NSGA-II without the substitution of opponent groups, which is represented by "n/a" in Table III is also carried out. The number of opponent groups is set to 10.

As a consequence of experiments, the number of Pareto individuals of all the results for substitution interval L is 200, that is, individuals in the final population are non-dominated by each other. In order to make comparisons among various substitution intervals L, the following experiments are examined: First, new 1200 opponent groups are constituted from two final populations with different substitution intervals. For each individual in the two final populations, a game is carried out for each opponent group. We made comparison by using the total point of scores for all the opponent groups and for all individuals. Table III summarize the result of this coma prison. $\diamond$ in the table indicates no significant difference between corresponding substitution intervals. $\circ$ and $\times$ denote that the substitution interval at corresponding row is better and worse than the one at corresponding column, respectively. As described in this table, the proposed method with $L = 20$ outperforms others.

C. Rule Design by Evolutionary Programming

We have so far enjoyed a large number of playings of the COMMONS GAME. Although those experiences have given us a valuable chance to consider seriously about the commons, we did find that some players lost interest, in the middle of the game, because the COMMONS GAME is comparatively monotonous. In order to make the game much more exciting, we have tried to find the reason why some players lost interest in the middle of the COMMONS GAME. We have come to the conclusion that the way that each player receives points when he chooses one of the five cards sometimes makes the game playing rather monotonous.

In particular, we have concluded that the following rule make the game playing monotonous: In the original COMMONS GAME, green card players receive a penalty, -20 points, when some player chooses a black card. On the other hand, black card players receive a point $-6/$(numbers of players who have chosen a black card). Orange card players receive a point $-6/$(numbers of players who have chosen an orange card).

We consider that some change in the points -20 and -6 mentioned above would make the COMMONS GAME much more exciting. In order to find an appropriate point for each card, we shall try to utilize EP.

1) Coding Method: In this paper, we employ three variables, W_G, OB, and W_o to represent rules. The meaning of them is described as follows:

1) Penalty P_G for green players – we shall propose an appropriate way for penalizing green players which takes the environmental changes into account:

$$P_G = -W_G \times (\text{Gain G}),$$

where W_G means the numerical value that is determined by the Evolutionary Programming, and "Gain G" denotes the return that the green players can get if any other player does not choose "black Card."

2) Point AOB that black players loose – we shall propose an appropriate way (for asking black players pay cost in trying to penalize green players) which takes the environmental changes into account:

$$AOB = OB/NOB,$$

where OB means the numerical value that is determined by the Evolutionary Programming, and NOB means the number of players who have chosen the black cards.

3) Point OR that orange players add to the red players – we shall propose an appropriate way (for helping red players maintain the commons) which takes the environmental changes into account:

$$OR = W_o \times (\text{Gain R}),$$

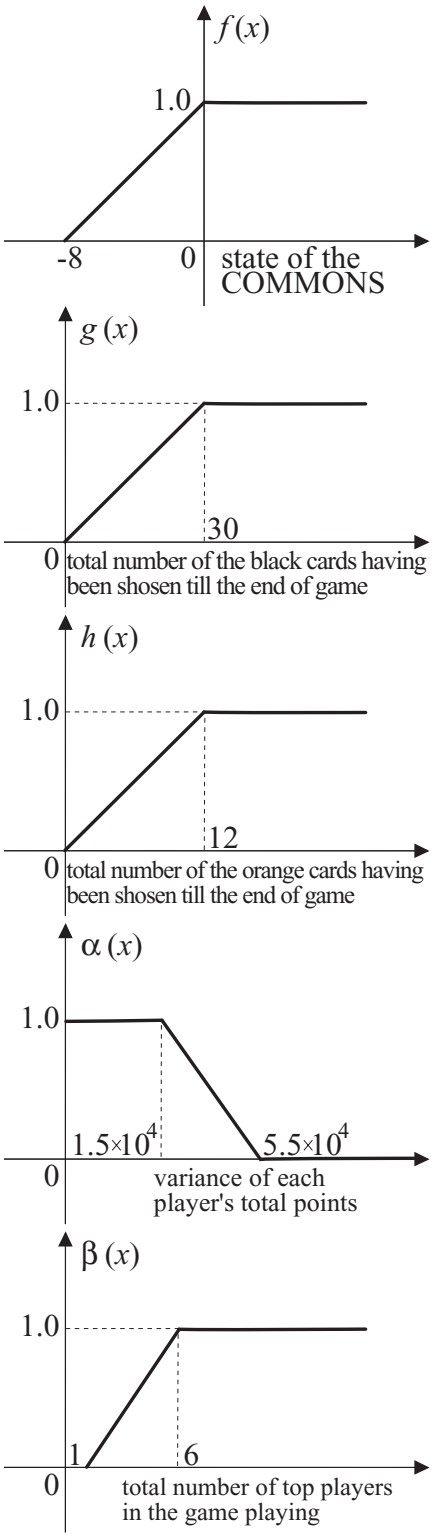

Fig. 3. Functions which constitute evaluation function $T(X)$

where W_o means the numerical value that is determined by the Evolutionary Programming, and "Gain R" denotes the return that the red players can get.

2) Fitness Evaluation: In order to evaluate the rule parameters mentioned in the previous subsection, 200 games per an individual are carried out. Before runs of Evolutionary Programming, six neural network players in the final population of MOEA are randomly chosen per game, as players group in Figure 1. Namely, 1200 neural network players are selected (including duplicated selection).

After each game, a rule parameter is evaluated by the following:

a) A game in which black cards & orange cards are seldom chosen (almost all of the players only choose a green card or a red card) is quite monotonous. Therefore, the more black (or orange) cards are chosen in a game, the higher value of the evaluation function should be given.

b) A game in which the ranking of each player often changes is exciting. Therefore, the more changes of the top player during a game run, the higher evaluation value should be given. A game in which there is a small difference between the total points of the top player and those of the last player is very exciting. Therefore, the small variance in the total points of each player at the end of a game, the higher the evaluation value should be given.

c) When the environmental deterioration had occurred heavily, each player can receive only a small amount of return. Under such a state of environment, the game should become monotonous. Therefore, a small evaluation value should be given if a game has brought heavy deterioration to the environment. On the other hand, a high evaluation value should be given when a game has brought a moderate final state of the environment.

By taking into account the above points, we have constructed the following evaluation function $T(x)$:

$$T(x) = f(x) + g(x) + h(x) + \alpha(x) + \beta(x),$$

where x denotes a chromosome. The function values of $f(x)$, $g(x)$, $h(x)$, $\alpha(x)$, and $\beta(x)$, correspond the following:

$f(x)$: The environmental state at the end of the game;
$g(x)$: The total numbers of the black cards having been chosen;
$h(x)$: The total numbers of the orange cards having been chosen;
$\alpha(x)$: The sum of the variance of the cards having been gained by each player;
$\beta(x)$: The total numbers of the changes of the top player.

3) Experimental Results: Evolutionary Programming with 10 population size has been carried out for 50 generations. The changes of averaged fitness value during the evolution are depicted in Fig. 4. This graph is plotted by averaging over 20 runs. The changes over the whole generations are not so much. However, the main contribution of these changes are caused by the changes of $\beta(x)$. This means that the utilization of EP has contributed a lot in changing top players.

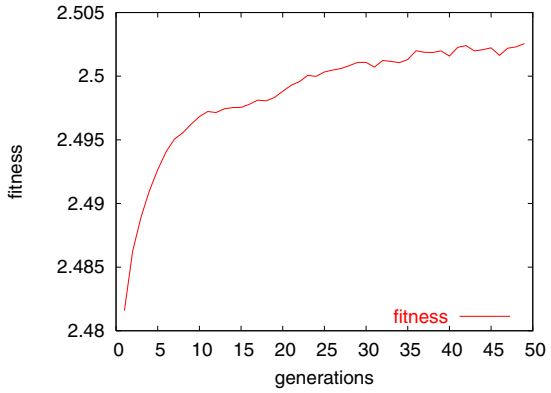

Fig. 4. Changes of averaged fitness value during evolution

The reason why other sub-functions, such as $f(x)$, $g(x)$ and so on, did not affect the evolution process is that the neural network players are already sophisticated enough: They play various kinds of cards, including black and orange cards so that the final environmental state has not been deteriorated so seriously.

We analyzed the best individuals in 20 runs. there are two groups in the best individuals: they are around ($W_G = 0.25$, $OB = 0.04$, $W_o = 0.28$), and ($W_G = 0.29$, $OB = 0.16$, $W_o = 0.2$), respectively. This analysis reveals that much penalization of green players causes the changes of top players.

IV. Conclusions

In this paper, we have tried to utilize two kinds of EAs, i.e., MOEA and EP for making the COMMONS GAME exciting. The MOEA has been used for generating various types of skilled players. The objective function of this MOEA is performances against different opponent groups. These opponent groups are substituted by current Pareto individuals at every predefined interval. Further, the EP has been introduced to find out appropriate combinations of the point of each card. As shown in the Fig. 4, we have succeeded in finding highly advanced rules compared with that of the original COMMONS GAME.

The future research effort should be directed to investigate whether the neural network players evolved in this research can cope with highly skilled human players. Further, we would like to check whether the acquired rules can help human players enjoy excited game plays. These are left for our future work.

Acknowledgement

The authors would like to express their thanks to the Foundation for Fusion of Science & Technology (FOST) and the Grant-In-Aid for Scientific Research (C) by Ministry of Education, Science, Sports, and Culture, Japan, who have given them partial financial support.

References

[1] Duke, R. D., "Gaming: The Future's Language," Sage Publications, 1974
[2] Hausrath, A., "Venture Simulation in War, Business and Politics," McGraw-Hill, 1971
[3] Shubik, M., "Games for Society, Business and War," Elsevier, 1975
[4] Shubik, M., "The Uses and Methods of Gaming," Elsevier, 1975
[5] Duke, R.D. and Duke, K., "Development of the Courail Game, Operational Gaming: An International Approach," Stahl, I. Editor, IIASA, pp. 245–252, 1983
[6] Powers, R., "The Commons Game," Instruction Booklet, 1980
[7] Ausubel, J. H., "The Greenhouse Effect: An Educational Board Game," Instruction Booklet, IIASA, 1981
[8] Greenblat, C. S., "Designing Games and Simulations," Sage Publications, 1988
[9] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A Fast and Elitist multi-objective Genetic Algorithm: NSGA-II," *IEEE Trans. Evolutionary Computation*, Vol. 6, No. 2, pp. 182–197, 2002
[10] Bäck, T., and Schwefel, H.-P., "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, Vol. 1, No. 1, pp. 1–23, 1993
[11] Yao, X., Liu, Y., and Lin, G., "Evolutionary programming made faster," *IEEE Trans. Evolutionary Computation*, Vol. 3, No. 2, pp. 82–102, 1999
[12] Baba, N., *et al.*, "A Gaming Approach to the Acid Rain Problem," *Simulation & Games*, Vol. 15, pp. 305–314, 1984
[13] Baba, N., "The commons game by microcomputer," *Simulation & Games*, Vol. 15, pp. 487–492, 1984
[14] Baba, N., "The Greenhouse Effect Game by Microcomputer," IIASA Professional Paper, PP-84-1, 1984
[15] Tesauro, G., "Temporal Difference Learning and TD-Gammon," *Communications of The ACM*, Vol. 38, 58-68, 1995
[16] Baba, N., "An Application of Artificial Neural Network to Gaming," in *Proc. SPIE Conference*, Invited Paper, Vol. 2492, pp. 465–476, 1995
[17] Baba, N., Kita, T., and Takagawara, Y., "Computer Simulation Gaming Systems Utilizing Neural Networks & Genetic Algorithms." in *Proc. SPIE Conference*, Invited Paper, Vol. 2760, pp. 495–505, 1996
[18] Baba, N., "Application of Neural Networks to Computer Gaming," *Soft Computing in Systems and Control Technology*, Tzafestas, S.G. Editor, World Scientific Publishing Company, Vol. 3, pp. 379–396, 1999
[19] Graham Kendall and Sushil Louis (editors), "Proc. IEEE 2005 symposium of Computational Intelligence and Games," IEEE press, 2005.
[20] Sushil Louis and Graham Kendall (editors), "Proc. IEEE 2006 symposium of Computational Intelligence and Games," IEEE press, 2006.