Evolutionary Algorithm for Game Difficulty Control

SANG-WON UM[*], JONG-SOO CHOI[*], JIN-TAE KIM[**], HO-KEUN SONG[****], HASUNG KOO[**]

[*] Department of Image Engineering, GSAIM, Chung-Ang University, Korea

[**] Department of Computer & Information Science, Hanseo University, Korea

Abstract: - In this paper we propose an evolutionary algorithm against player (EAP) for game that controls the difficulty of a game based on the player's propensity and proficiency fundamental using the Genetic Algorithm (GA). This paper describes how we use the GA to control the level of difficulty in a game based on a user's skill. Most game AI techniques so far have been focused on the realistic and smart behavior of game units or game appearance. It a player competes with exciting opponents in a game, game AI is involved in not game appearance or game environments but exciting opponents. AI techniques make game-play richly, but unfortunately they have rarely been used in games. We suggest a game algorithm that enables a game to change the difficulties by itself based on the player's suitability to the game using the GA.

Key-Words: - Game, Game AI, Controlling Difficulties, Genetic Algorithm

1. Preface

Presently one of fastest developing industry is game industry. Recent games commonly have two features, one is fine graphic feature and the other is networking feature. Since most developers believe crafting a game graphic has become most important, they focus on realistic graphics or game appearance. But graphic development seems to reach a limitation.

Another peculiar feature is networking. Networking of a game is inevitable result concerning the modern AI techniques. People are very skillful who grew up loving computer games. Therefore the lifecycle of the game comes to be short little by little. If they are easily skillful for a game, they want more difficult opponents but it offers limited opponents. It causes players to feel disgust for the game. So many developers complement it using networking.

But if we concern purpose and motivation in playing the game, it is necessary that we research the game AI. If there is a computer game as intelligent as the game player, they so hard to get tired of a game. But modern AI techniques are at first step; moreover it is not comprehended to everything of AI technique in the computer game. Sometimes a game appears intelligent. But it is not the traditional AI technique by researchers that used to just trick optimized to the game [1-2].

So far the most of AI in field of the game have used to establish more realistic situation. It is not take part in the game-play but used to feel more actually or to avoid simple repetition. For instance, in case of the StarCraft, when NPC moves after receiving a command from the player, AI technique (e.g. finding path algorithm, A*) is used. But it is not directly related to the game-play. Just it appears more realistic to the player. It is smart terrain introduced in the Sims that is revolutionary AI applied to the computer game. The Sims is enough to make the game play more interesting and abundant.

2. Genetic Algorithm

2.1 Genetic Algorithm

The genetic algorithm is one of the optimized algorithms from computer simulation for evolution process of the law of the survival of the fittest. General elements of GA are the Fitness function, Gene, Generation, Selection, Crossover, and Mutation.

Where the Fitness function is the problem to be solved and Gene is the candidate of the solutions and Generation is the set of solutions and the last three parts, Selection, Crossover, Mutation, are the fundamental operator [3].

2.2 Schema Theory

The GA is to be accomplished quite with the structure which is very simple. First it selects two genes which are adapted to family environment well. Crossover is an interbreeding two child genes half inherit the characters of two parents gene. And Mutation is process randomly transform bit of one or less number with it executes at very small ratio. Mutation prevents the fact that it falls in to the local optimum. And we have suitable its solution to repeat until the level which is environment's wants. GA prove mathematically is arrange exactly schema theory or schema theorem. Schema theory was proved previously for GA which based on mathematics by mathematicians.

This paper escape deeply explanation because is not the paper for GA or schema theory, and more detailed the fact where attitude is described in the reference literature [3-5].

3. Game Analysis

The speed up or amount of opponents means increasing difficulty in the shooting game. In this section, we analysis representative game genre and identify possibility that GA should have been possible to be how application to different genre game.

3.1 The Arcade Game

The many people have been played the arcade game before wide spread personal computer. The arcade games have very simple scale and interface. We chose two simple games to analyze the arcade game.

The vehicle shooting game is representative an arcade game genre. Player competes with opponents riding a vehicle. Concern the enemy speed and the number of missile which is fired by enemy. It old-fashionably does not change speed or number of missile own enemy on the game.

We chose the Galaga which is representative the vehicle shooting game. If the enemy speed or number of missile changes at any time, the Galaga has become a more exciting and various.

If it applies a GA for the vehicle shooting game,
Chromosome: C [include Speed, Missile]
Gene: Change of C per 10 unit times,
G(t) = {C0, C1, C2, ..., C9}
Fitness functions:
-The variation of score which is obtained by a player during a certain time.
-The variation of the enemy units during a certain time.

The action game appeared dislike the vehicle shooting game. We chose Bankpanic which is an arcade action game. The Bankpanic was published by the Sega in 1984, player have to protect the deposit from burglars. This game is very worse that the order of visitors is predetermined. Of course, the developer of Bankpanic would have constructed this game robustly. But if player is a little clever, he maybe knows the order of visitor next time. How long time people are willing to play such game? Maybe, the developer would spend a lot of time to construct this game.

In the case of this game,
Chromosome: The order of visitors. C=[Pl, Pc, Pr]
Gene: The chromosome changes per 10 unit times. G(t)={C0, C1, C2, …, C9}
Fitness functions:
-The variation of score which is obtained by a player during a certain time.
-The ratio of burglars per the number of visitors.

3.2 The PC Game

The PC game is a category of the game with an arcade game. Many people are daily playing the PC game.

The Diablo is a recent famous the role-playing game. It is a computer games, and compounds form of action, role-playing, and adventure genre. The number of opponents is constant in one stage of mission game, and the items obtained in the stage are also limited. On the assumption that number of opponents is limited, if they are closely each other, player faces trouble to remove all opponents. There are difficulty follow in Cohesion of opponents.

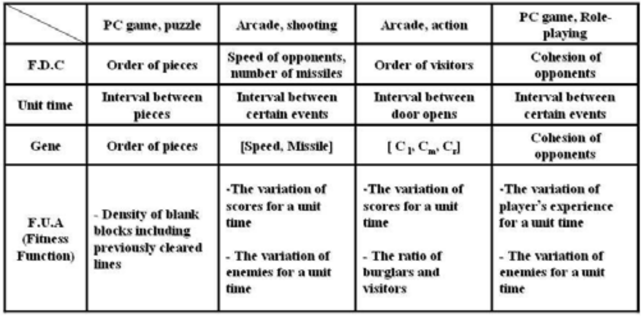
In the Diablo,
Chromosome: There is not Chromosome.
Gene: Cohesion which is average distance between each opponent.
Fitness functions:
-The variation of player's own experiences (score) per certain time.
-The variation of enemy number per certain time.

The puzzle game is oldest game in mankind. Tetris game is the representative a puzzle game that people would play at least once. What we would like to focus on is that the pattern of pieces. There are must be the pattern which each player feels easy or hard. We control difficulty of game using pattern which means order of pieces as genes.

In the Tetris which is the puzzle game,
Chromosome: There is not Chromosome.
Gene: The order of pieces.
Fitness function: Density of blank block considering previously clear lines.

Until now we analyzed four games of some genres. The Table 1 is the analyzed result.

Table1. The analysis of game genre and factor of GA. We will experiment simulation on the Tetris because we think the analysis of Tetris is obvious than the others.



	PC game, puzzle	Arcade, shooting	Arcade, action	PC game, Role-playing
F.D.C	Order of pieces	Speed of opponents, number of missiles	Order of visitors	Cohesion of opponents
Unit time	Interval between pieces	Interval between certain events	Interval between door opens	Interval between certain events
Gene	Order of pieces	[Speed, Missile]	[C_1, C_m, C_r]	Cohesion of opponents
F.U.A (Fitness Function)	- Density of blank blocks including previously cleared lines	-The variation of scores for a unit time - The variation of enemies for a unit time	-The variation of scores for a unit time - The ratio of burglars and visitors	-The variation of player's experience for a unit time - The variation of enemies for a unit time

4. Implement of experiment

In the case of Tetris, there will be easy pattern or hard one for the users to play according to the order of pieces. If four I-shape pieces and a square piece coming out, you lay down parallel four I-shape pieces and the rest of blank area was occupied square piece, then you can see removing 2 lines in your visual. This is an example of easy patterns. However, this example is under the assumption that the last line is flat. The degree of easy or hard pattern is very variable and flexible differ from the situation or the players.

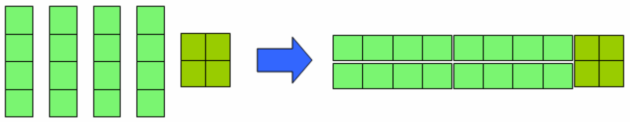


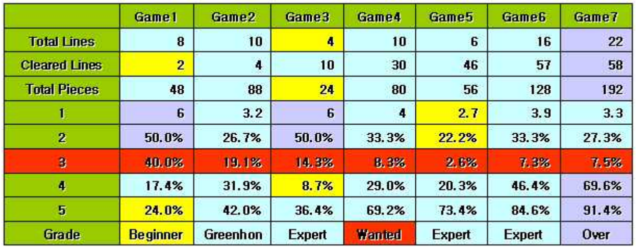
Figure 1. Easy pattern of pieces

We guess that this can be applied to, as well as Tetris, the other genre games which we mentioned before.

4.1 Difficulty and Fitness

If the factors to control difficulty and evaluate fitness are determined, then algorithm is processed according to general sequence of the Genetic algorithm. The factors of difficulty and fitness are decided to the intention of developers. Just the reason to analyze many games, we considers various kind genre so that the algorithm generalize to adapt all over the game. Table 2 is an analysis of fitness function in the Tetris, we selected the third case which is representative player's adaptness.

Table2. The analysis of fitness function



	Game1	Game2	Game3	Game4	Game5	Game6	Game7
Total Lines	8	10	4	10	6	16	22
Cleared Lines	2	4	10	30	46	57	58
Total Pieces	48	88	24	80	56	128	192
1	6	3.2	6	4	2.7	3.9	3.3
2	50.0%	26.7%	50.0%	33.3%	22.2%	33.3%	27.3%
3	40.0%	19.1%	14.3%	8.3%	2.6%	7.3%	7.5%
4	17.4%	31.9%	8.7%	29.0%	20.3%	46.4%	69.6%
5	24.0%	42.0%	36.4%	69.2%	73.4%	84.6%	91.4%
Grade	Beginner	Greenhon	Expert	Wanted	Expert	Expert	Over

1. Blank blocks of each lines: more than 25% - the beginner; less than 12% - the expert
2. The density of blank block except clear lines.
3. The density of blank block including clear lines.
4. The density of block per game board.
5. The density of block per game board including clear lines.

4.2 Implement of experiment

The experiment of the algorithm used CG Tetris which the source was opened public. The factor of GA follows in experiment.

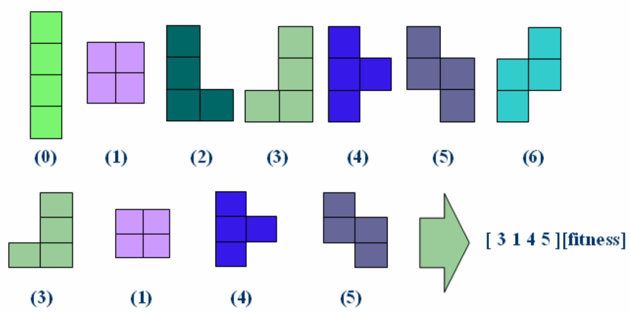


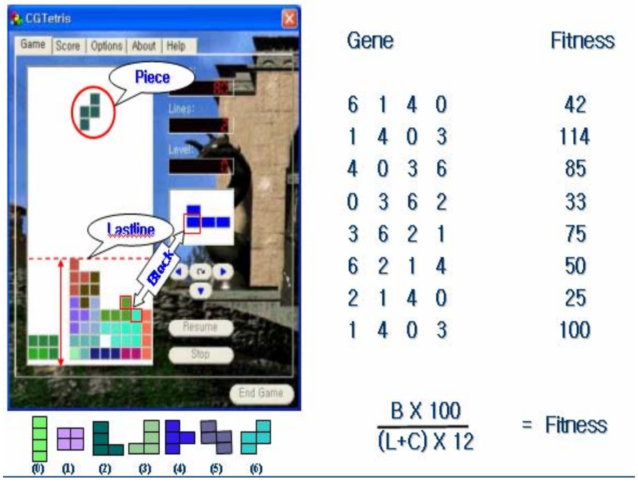
Figure 2. The pieces and encoding Gene

Gene: The order of four pieces.
Generation: The set of eight Genes are sorted.

Crossover: Select two genes which are selected by developer wants Fitness, and perform one point crossover.
Mutation: Perform about 10±5% ratio with crossover operator.
Fitness: The density of blank blocks including clear lines.

$$Fitness = \frac{B \times 100}{(C+L) \times 12} (\%)$$

$$B : Blank\,Block \quad C : Clearline$$
$$L : Lastline$$



Gene	Fitness
6 1 4 0	42
1 4 0 3	114
4 0 3 6	85
0 3 6 2	33
3 6 2 1	75
6 2 1 4	50
2 1 4 0	25
1 4 0 3	100

$$\frac{B \text{ X } 100}{(L+C) \text{ X } 12} = \text{Fitness}$$

Figure 3. Algorithm and proposed words

Figure 3 shows that CG Tetris, the adaptive target of Genetic algorithm, and its rough description about process. For the purpose of easy understand, we defined some notation like ‘Piece’ or ‘Block’. The Piece is composed of four blocks, and it is represented to the numbers 0 to 6. In the case of CG Tetris, the Game-board is composed of 12×23 blocks. The Fitness equation contains some elements like B, C, and L. B is the number of blank blocks in the game-board, C is the number of cleared lines, and L is the number of exist lines in the game-board. The denominator of Fitness function is a total number of lines, contained to cleared lines, thus the Fitness function means density of blank blocks.

Let T is the generated total number of pieces, and then next relation is possible.

$$(L+C) \times 12(b) - B(b) = T \times 4(b)$$
$$(L+C) \times 12 = 4T + B \qquad \text{--------------(1)}$$

In the equation (1), T×4 is a total number of generated blocks, summation of 4T and B equal to summation of cleared and existed lines. 12 means one line is composed twelve blocks. Using the relations of Equation (1) and Fitness equation, we can obtain next equations.

$$\frac{B \times 100}{4T + B} = F(\%)$$
$$100B = 4FT + FB$$
$$(100 - F)B = 4FT$$
$$\frac{B}{T} = \frac{4F}{100 - F}$$
$$\therefore \frac{B}{T} = C \qquad (\text{ F : fitness, constant })$$

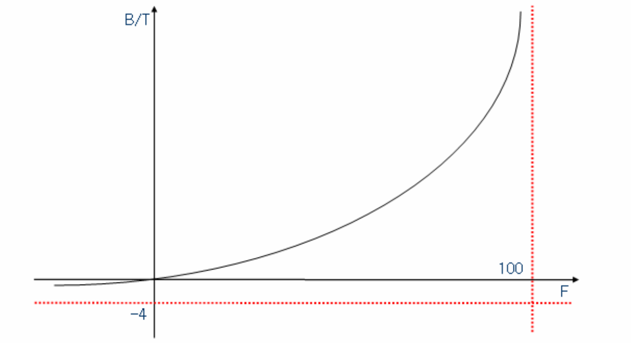


Figure 4. The graph of B/T follows Fitness

So, we can control the Fitness value in this way, to decide how many blank blocks should be occurred on each block is generated. Figure 4 is the graph of B/T follow Fitness which is wanted by developer, F is a constant then B/T is constant too. We used that equation for estimate player’s adaptness.

In this paper, the Fitness value is considered to an adaptive element about each gamer in the Tetris game. Maybe in a viewpoint of developer, to decide this adaptive element is easy. The experiment is realization of rand() function which is random function using in many game. There is no exception in CG Tetris. The Tetris was constructed integer between 0 and six which is occurred by using rand() function.

Input of function GA(I) is current Lastline, Clearedline, Blankblock. GA(I) output an integer calculated by proposed algorithm like rand() function. Figure 4 is the block diagram which will apply general game to proposed algorithm.

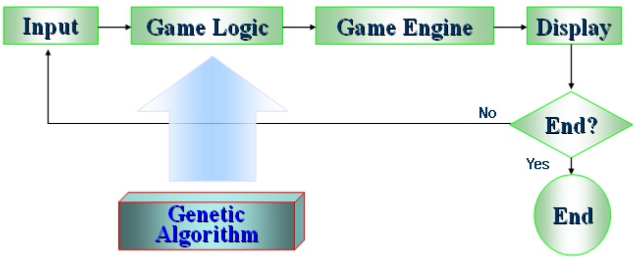


Figure 5. The block diagram of a Game

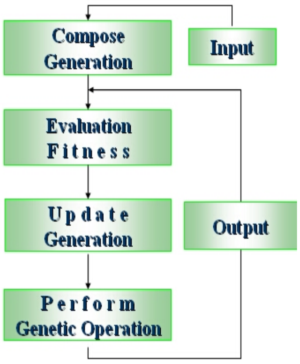


Figure 6. The block diagram of proposed algorithm

Figure 6 is the flow of proposed algorithm. First, it generates the Generation then estimate the Gene by L, C, and B. And it modifies and sorts the Generation by the Fitness. Lastly it output adequate integer.

As the listed algorithm, it is commonly possible to apply to analyzed four or the other games. As the same to table 1, if that apply by modification of Gene and Fitness function, it would be progressed and progressing game than existed game.

4.3 Result of experiment

The experiment is accomplished by the process to confirm the result by execution existed CG Tetris and program applicant proposed algorithm for eight people. The wanted fitness using in the experiment is executed to get density of blank block as the proportion 3.8% and 5.7%. Next part is the result of experiment.

There is Clear line about each game in Figure 7. As comparison of average, they got lower result than origin, and more difficult by 5.7% fitness than 3.8%. The variance is more and more small processing experiments in the rightmost of Figure 7. We want that the game was ended same time for anyone who expert or beginner. But the gap of those is very enormous to overcome.

We were satisfied that the variance of tester's clear line more reduce than origin game by experiment. And difficulty control is possible simple change fitness which is the density of blank block.

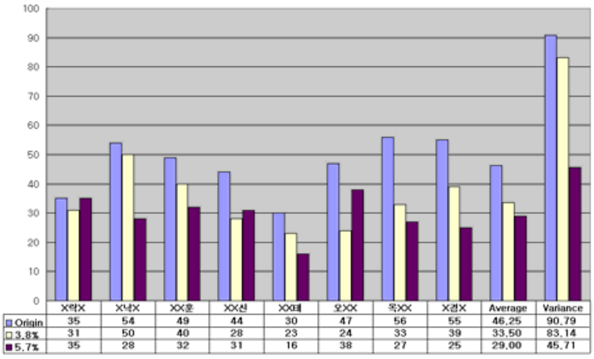


Figure 7. Clearedline of each person

5. Conclusion

So far, we research the possibility of difficulty control using Genetic algorithm by analysis of each field represented game-shooting, action, puzzle, role-playing and we confirmed. And, we experiment changing the difficulty using pattern of pieces applied Genetic algorithm. In this paper, we control the difficulty of a game based on the player's propensity and proficiency fundamentally using Genetic Algorithm. UDCA is designed to be possible to commonly use in many game genre.

If UDCA was concerned to the beginning of project, it was had very robust and powerful performance and it will make the various game changed by player's adaptness. Developer most knows that the factor of difficulty and fitness.

References:

[1] J. E. Laird. "Using a computer game to develop advanced AI," IEEE Tran. Computer, 2001.

[2] S. Woodcock. "Game AI: The State of the Industry," Gamasutra Magazine, 2000.

[3] D. Goldberg, Genetic algorithms in search optimization and machine learning, Addison-Wesley, 1989.

[4] D. E. Goldberg and K. Sastry, "A Practical Schema Theorem for Genetic Algorithms Design and Tuning," IlliGAL Report No. 2001017.

[5] R. Poli and W. B. Langdon. "A New Schema Theory for Genetic Programming with One-point Crossover and Point Mutation," School of Computer Science, The University of Birmingham.