



Effect of Parameters of Genetic Algorithm on Results in Action Games

Myun-Sub Lee*

Department of Computer Science and Engineering, Incheon University

ABSTRACT

In this paper, it was intended to identify effect of parameter values of genetic algorithm on results. As the study subject, an action game with 10 actions was used. In the action game, intelligent characters were set to find optimal solutions by evolving by themselves along with repetition of generations. Among several parameters of the genetic algorithm, population size, cross rate, and mutation rate were changed and applied. The range of population size was set to be from 50 to 950, and the cross rate and the mutation rate were applied, changing from 0.7% to 1.0% and from 0.005% to 0.24% respectively. As the results, it was achieved to find optimal solution after 90 generations. It was found that the combination of parameters to identify optimal solution within the most rapid time were 300 in population size, 1 in cross rate, and 0.01 in mutation rate. Because this combination of parameters cannot be found with some trials, there are some difficulties that a number of repeated experiments with various combinations of parameters are required

© 2015 KKITS All rights reserved

KEYWORDS : Genetic algorithms, Action games, Fitness, Cross over, Mutations

ARTICLE INFO: Received 6 November 2015, Revised 11 December 2015, Accepted 11 December 2015.

*Corresponding author is with the Department of Computer Science & Engineering Incheon University, (Sondo-dong) 19, Academy-ro, Yeonsu-gu, Incheon, KOREA.
E-mail address: nantian@inu.ac.kr

1. 서론

유전자 알고리즘은 지역 탐색과 전역 탐색 사이에서 조화를 이루며 해를 탐색하는 알고리즘으로서 그 성능은 여러 파라미터와 밀접한 관계를 가

지며 개체의 크기, 교배 확률, 돌연변이 확률이 중요한 파라미터이다[1][2]. 개체 크기는 그 안에 얼마나 많은 개체가 있는지를 말해 주는 것으로 성능에 직접적인 영향을 준다. 만일 그 크기가 너무 작으면 유전자 부동으로 인해 원하지 않은 방향으로 빨리 적응해 가고, 너무 크면 성능 향상에 비해 너무 많은 처리 시간이 걸린다. 교배는 연색체간에 유전자 교환을 통하여 전역 탐색을 돕지만 재산을 거쳐 형성된 염색체 내에서 유전자를 교환하기 때문에 유전자의 변화는 제한적이다. 교배 확률이 너무 낮으면 다음 세대에서 새로운 개체 발생이 적게 되어 탐색이 제한되고, 높게 설정되면 빨리 수렴하는 특징이 있다. 돌연변이 확률을 너무 낮게 설정하면 지역해로 수렴하며 돌연변이 확률을 높게 설정하면 급격한 자연 환경의 변화로 잠재적으로 우수한 개체들이 다음 세대로 전달되지 못하고 사라질 수 있다[3][4].

본 논문에서는 유전자 알고리즘의 결과에 영향을 미치는 여러 가지 파라미터 중에서 개체의 크기, 교차 확률과 돌연변이 확률 세 가지 값을 변화시켜 가면서 빠른 시간 안에 최적의 값을 찾는지를 알아보았다. 실험 방법으로는 10 종류의 행동을 하는 대전 액션 게임에 적용하였다. 반복된 실험을 통하여 개체수 300, 교배확률 1, 돌연변이 확률 0.01일 때 세대수 90에서 최적의 해를 찾을 수 있었다.

2. 유전자 알고리즘

유전자 알고리즘은 제일 먼저 초기 집단이 생성되고 생성된 염색체 집단의 적합도가 평가된다. 적합도에 따라 좋은 형질의 유전자가 확률적으로 더 많이 선택되어 다음 세대의 집단을 형성한다. 선택된 개체에 대하여 교배와 돌연변이 조작을 하게 된다. 이와 같이 한 세대동안 재생산, 교배, 돌연변이

이를 거치면서 새롭게 형성된 집단은 다시 평가되고 앞의 과정을 반복하면서 최적해나 종료 조건이 만족 할 때까지 반복된다[5].

초기 집단으로 생성된 개체들은 많은 세대를 거치는 동안 좋은 해의 유전자를 보유하게 된다. 초기 집단의 생성에는 무작위 초기화 하는 방법과 유도된 초기화 방법이 있다[6]. 전자는 단순히 난수 발생기에서 생성하여 개체크기*염색체 길이로 초기화된다. 실세계의 문제에 응용할 때 해에 대한 사전 지식이 있으면 전역해로 수렴 속도가 빨라지므로 후자를 선택하는 것이 좋다. 개체의 크기 즉 개체의 수를 결정하는 것은 문제의 난이도나 성질에 의존하지만 일반적으로 수십 개 이상 발생시킨다. 개체수가 너무 적으면 병렬적 처리를 특징으로 하는 유전자 알고리즘의 장점이 발휘되지 못하고 반대로 너무 많으면 한 세대 당 연산 양이 많아지는 문제점이 있다. 개체의 크기를 너무 크게 하면 성능을 크게 개선하지 못한다고 알려져 있다[7]. 이러한 이유는 크기를 증가시키더라도 탐색속도가 좋아지지 않기 때문이며, 개체의 크기로는 30-200 사이에서 설정한다.

적합도(fitness) 평가는 환경에 대한 적응 능력을 반영한 것으로 한 세대가 진행될 때 마다 평가하여 적합도가 높은 개체는 다음 세대로 유전될 확률이 높게 된다. 즉 선택과정에서 적합도가 높은 개체는 선택 확률이 더 높다.

선택 연산자는 환경에 잘 적응한 해들은 살아남도록 하고, 그렇지 못한 해들은 도태되도록 염색체를 선택된다[8,9]. 각 개체들은 자신의 적합도에 비례하여 선택되며 다음세대를 생산할 수 있는 기회를 차등적으로 부여 받는다.

교배 연산자는 한 쌍의 유전자들이 염색체를 교환하는 과정이다[10]. 교배확률(P_c) $P_c=1$ 이면 선택된 염색체에서는 교배가 매번 발생하고 $P_c=0$ 이면 교배가 발생하지 않는다. 교배 연산자에 의해서 개

체들 사이에 유전 정보가 서로 교환되지만 모든 해 공간을 탐색하기 위한 유전 정보가 현재 군집 내의 개체들 들어 있지 않다면 교배 연산자를 아무리 적용하고, 세대가 진행되더라도 더 이상의 탐색이 이루어질 수 없다. 따라서 새로운 유전 형질을 부여할 수 있는 돌연변이 연산자는 필수적이다. 돌연변이 확률(Pm)이 100%이면 모든 염색체의 유전자들이 돌연변이가 발생되고, 0% 이면 아무것도 발생하지 않는다. 자연계에서 돌연변이가 일어날 확률은 매우 희박하므로 유전자 알고리즘에서 매우 낮게 설정한다. 이 확률을 높게 하면 무작위 탐색과 같으므로 좋은 해의 성질을 바꿀 가능성이 있다. 반대로 돌연변이 확률이 너무 작으면 지역해에 빠질 수가 있다. 일반적으로 Pm은 0.5% - 1% 사이로 설정한다.

3. 대전 액션 게임

본 절에서는 대전 액션 게임에 대하여 설명한다. 유전자 알고리즘에 적용되는 여러 가지 파라미터 값의 변화가 결과에 어떤 영향을 미치는지를 알아보기 위해 간단한 대전 액션 게임에 유전자 알고리즘을 적용하였다. 액션 게임에 사용된 염색체는 캐릭터간의 거리와 상대 캐릭터의 행동에 따라 지능 캐릭터가 어떤 행동을 할 것인지를 규정한다. 이에 따라 염색체는 2차원 구조를 가진다. 각 캐릭터가 할 수 있는 행동의 종류가 m 개이고, 상대방과의 거리를 n 단계로 구분한다면, 염색체는 <그림 1>과 같이 m*n(11*7) 배열의 구조가 된다. 그리고 배열내의 각각의 값은 특정 상황에서의 지능 캐릭터의 행동을 정의하는 것으로 0에서 m-1 까지의 값으로 표현된다. 그림에서 네모안의 숫자는 지능 캐릭터의 행동으로서 두 캐릭터 사이의 거리와 상대 캐릭터가 어떤 행동을 할 때 지능 캐릭터의 행동을 보여 주고 있으며 랜덤하게 생성된 구조이다.

예를 들면 거리가 2이고 상대 캐릭터가 위 주먹 공격(7)을 할 때 지능 캐릭터의 행동은 9(위발 공격)를 의미한다.

선택 연산은 룰렛휠 선택[10] 방법을 사용하였으며, 적합도는 어떤 캐릭터가 각각의 거리에서 100 시간 단위 동안 게임에서 획득한 점수를 적합도로 정의하였다. 본 연구에서는 한 세대 내에서 가장 좋은 적합도를 가지는 개체는 반드시 다음 세대에 자신의 우성인자를 전달하는 엘리트 보존 전략을 적용하지 않았다. 교배 연산자는 유전자의 형태가 이차원이므로 다차원 교배인 블록 균등교배를 적용하였다. 돌연변이 연산은 염색체 표현이 정수이므로 돌연변이 확률에 해당되던 난수를 발생시켜 선택된 값으로 대체하였다.

상대행동 \ 거리		거리						
		0	1	2	3	4	5	6
0	정지상태 (ID)	4	9	1	10	0	9	2
1	전진 (GO)	7	2	6	1	4	3	7
2	후진 (BK)	9	7	1	0	8	10	1
3	막기 (GD)	3	5	9	3	2	7	10
4	앞기 (DN)	2	7	6	1	10	0	6
5	점프 (JP)	2	6	9	3	8	3	3
6	아래주먹 (DN)	3	9	7	0	5	9	9
7	위 주먹 (UN)	10	7	9	2	7	3	2
8	아래발공격 (DK)	3	1	9	7	9	2	8
9	위 발공격 (UK)	7	4	1	6	10	8	2
10	필살기 (SP)	0	9	3	2	5	4	6

그림1. 염색체의 구조
Figure 1. Configuration of Chromosome

<그림1>은 게임 규칙으로 게임에서 캐릭터가 취

할 수 있는 행동의 종류로는, 정지 상태(ID), 전진(GO), 후진(BK), 막기(GD), 앉기(DN), 점프(JP), 아래 주먹공격(DP), 위 주먹공격(UP), 아래 발 공격(DK), 위 발공격(UK), 그리고 필살기 공격(SP)등 11가지이다. 두 캐릭터는 모든 거리에서 어떠한 공격도 가능하지만 점수로 인정받기 위해서는 표1에서와 같이 공격에 따라 제한된 유효거리 내에서 공격이 이루어질 때에만 점수가 인정된다[11].

상대방의 공격에 대처하는 방법으로는 GO, BK 등으로 상대방과의 거리를 조절하거나, GD, JP, DN 등의 방어 자세를 취하거나, 같이 공격하는 방법이 있다.

표 1. 공격 형태 및 점수
Table 1. Kind of attack and point

게임 규칙							
거리	0	1	2	3	4	5	6
행동	0	1	2	3	4	5	6
D P (1점)	D P						
U P (2점)	U P						
D K (3점)			D K				
U K (4점)			U K				
S P (5점)				S P			

거리 조절 방법은 상대방이 현재 취하고 있는 공격의 유효거리 밖으로 이동하는 것이다. 예를 들어 상대방이 거리 2에서 DP 공격을 해 올 경우에 BK 하면 상대방의 공격이 무효가 된다. 그러나 만약 거리가 1인 경우에는 GO나 BK을 해도 타격을 받는다. 이런 경우에는 방어 자세를 취하는 방법은 따로 정하였다.

<표 2>는 앞서 제시한 게임 규칙들을 이용하여 각 상황에서의 최적의 행동을 구한 것이다. 최적의 행동이란 주어진 상황에서 자신의 점수의 증가치와 상대방 점수의 증가치의 차이를 최대로 만드는

자신의 행동을 의미한다. 예를 들어 서로간의 거리가 0이고 상대방이 ID 행동을 취하고 있을 때를 가정해 보자. ID는 공격 행동이 아니므로, 상대방의 점수의 증가치는 0이다. 한편 <표 1>에 의하면 거리 0에서 취할 수 있는 유효한 공격으로는 DP와 UP가 있는데 DP가 UP에 비해서 더 높은 공격 점수를 가진다. 그러므로 이 상황에서는 UP 공격이 최적의 행동이 된다.

<표 2>에서 * 로 표시된 것은 어떠한 행동을 취해도 무방하다는 의미이다.

표 2. 최적일 경우 염색체 구조와 행동의 종류
Table 2. Configuration of Chromosome , Kind of attack

게임규칙								
거리	행동	0	1	2	3	4	5	6
0	ID	7	7	9	10	10	10	*
1	GO	7	7	9	10	10	10	*
2	BK	7	7	9	10	10	10	*
3	GD	*	*	*	10	10	10	*
4	DN	6	6	8	10	10	10	*
5	JP	7	7	9	10	10	10	*
6	DP	7	7	9	10	10	10	*
7	UP	7	7	9	10	10	10	*
8	DK	7	7	9	10	10	10	*
9	UK	7	7	9	10	10	10	*
10	SP	7	7	9	10	10	10	1

예를 들어 상대방이 거리 0에서 GD 자세를 취하고 있다면, 지능 캐릭터가 어떠한 행동을 취하더라도 두 캐릭터의 점수에는 전혀 변화가 없다. 또한 1로 표시된 것은 1번 행동(GO) 외의 어떠한 행동도 무방하다는 의미이다.

4. 실험 결과 및 검토

본 논문의 실험에 사용된 최초의 파라미터 값으로는 최대 세대수 1000, 교배율 $P_c=0.7$, 돌연변이 확률 $P_m=0.0005$ 개체 크기 P_s 는 50으로 시작하여 각 파라미터 값을 변화시키면서 나온 결과가 <그림2>이다. 개체 크기는 50부터 950까지 50단위로 증가하였으며, 교배율은 0.7부터 1.2까지 0.02 단위로 증가하였고, 돌연변이율은 0.0005부터 0.026까지 변화시켜 가면서 실험을 반복하였다. 그림의 결과는 세 가지 파라미터 값을 변화시켰을 때 최적해를 찾은 세대수를 의미한다. <그림2>의 개체수를 변화에서 개체수 $P_s=300$ 일 때 가장 빠른 90세대만에 최적해에 도달하였다. 개체수가 50일 때 최적해에 도달하는 세대수가 갑자기 증가 하였는데 이 이유는 개체수가 너무 적게 되면 탐색범위가 좁아 새로운 타입의 유전자가 나타나지 않아 다음 세대에 좋은 성질의 유전자를 전달하지 못한 결과라고 생각한다. 개체수를 변화를 그래프로 나타낸 것이 <그림 3>이다. <그림 3>에서 최적해에 도달하는 세대수가 랜덤하게 나타나는 이유는 엘리트 보존 전략을 적용하지 않았기 때문이다.

<그림 2>의 교배율 P_c 를 변화시킨 결과를 보면 P_c 가 1일 때 90 세대에서 최적해에 도달한 것을 볼 수 있다. 이는 선택된 염색체에서 교배가 매 회마다 발생하는 것을 의미한다. <그림2>의 P_c 에 대한 그래프가 <그림4>이다. 여기에서도 결과 그래프에서 세대수에 리플이 나타나는 이유는 엘리트 보존 전략을 적용하지 않았기 때문이다. $P_c=1$ 일 때 90세대에서 최적해를 찾은 이유는 부모 염색체에서 유전 정보를 매 회마다 다음 세대로 전달하였기 때문이라고 생각된다. <그림2>의 돌연변이 확률을 변화시킨 결과에서 P_m 이 0.01일 때 90세대에서 최적해에 도달하였으며 <그림5>에 표시하였다. <그림5>에서 돌연변이 확률 0.01일 때 가장 좋으며 이

보다 확률이 적거나 크게 되면 점점 더 세대수가 오래 걸리는 것을 알 수 있다. 확률이 적을 경우에는 새로운 타입의 염색체를 생성하지 못해 해가 수렴하는데 시간이 오래 걸리기 때문이다. <그림 2>에서 P_m 이 0.026 일 때는 세대수를 500,000회로 늘려 실험을 했을 때 마지막 회에서 최적해에 97.4% 즉 2곳에서 최적해를 찾지 못하였다. 돌연변이 확률이 작게 되면 부모 염색체가 우수한 형질의 유전 정보를 확보하지 못할 경우 계속하여 다음 세대로 나쁜 유전 정보가 전달되기 때문에 새로운 해를 찾는데 실패하기 때문이다.

개체수 변화		교배율 변화		돌연변이율 변화	
P_s	세대수	P_c	세대수	P_m	세대수
50	1407	0.7	328	0.0005	1332
100	300	0.72	354	0.001	861
150	327	0.74	326	0.002	547
200	155	0.76	315	0.004	432
250	173	0.78	298	0.006	327
300	90	0.8	206	0.008	197
350	130	0.82	231	0.01	90
400	178	0.84	203	0.012	201
450	184	0.86	248	0.014	181
500	170	0.88	194	0.016	262
550	196	0.9	209	0.018	237
600	215	0.92	248	0.02	405
650	212	0.94	242	0.022	683
700	203	0.96	218	0.024	769
750	501	0.98	111	0.026	500,000
800	534	1	90		
850	501	1.2	90		
900	633				
950	903				

그림 2. 적용한 파라미터 값
Figure 2. Parameter value

이는 돌연변이 확률이 크게 되면 좋은 염색체가 다음 세대로 전달되기 전에 돌연변이를 일으켜 좋은 형질의 유전자 다음 세대로 전달되지 못하기

때문이다. 따라서 본 논문에서의 적용한 대전 액션 게임에서 가장 좋은 조건의 파라미터 조합은 개체 수는 300, 교배확률은 1, 돌연변이 확률은 0.01 이다.

이러한 조합을 찾아내는 것은 몇 번의 실험으로 찾아지는 것이 아니라 반복적으로 다양하게 파라미터 값을 변화시키면서 수렴점을 찾아야 하는 어려움이 있다.

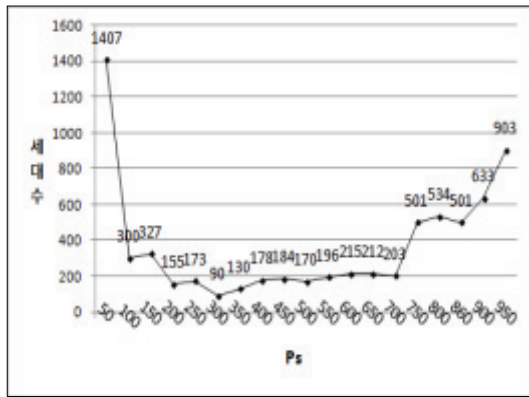


그림 3. 개체수와 세대수의 관계

Figure 3. The relationship between Generation and Population size

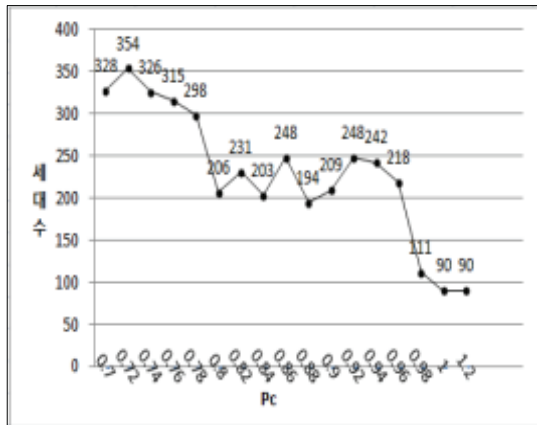


그림 4. 교배율과 세대수의 관계

Figure 4. The relationship between Generation and Crossover

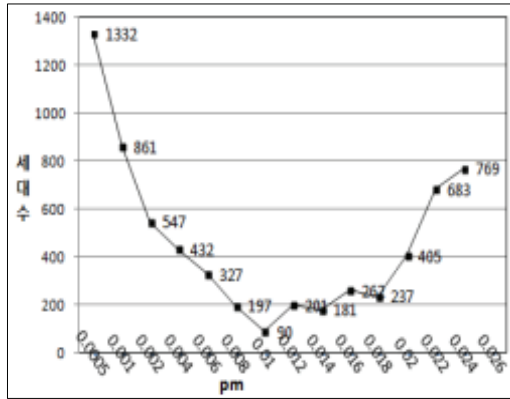


그림 5. 세대수와 돌연변이의 관계

Figure 5. The relationship between Generation and Mutation

이 조합은 본 논문의 경우에만 유효한 것이다. 다른 문제의 해를 찾다가 할 때 유전자 알고리즘을 적용한다면 이와 같은 방법으로 반복 실험을 한다면 좋은 결과가 있을 것이라고 예상된다.

5. 결 론

본 논문에서는 유전자 알고리즘의 파라미터 값이 결과에 어떠한 영향을 미치는지 알아보았다. 유전자 알고리즘 파라미터 중에서 개체의 크기, 교배 확률, 돌연변이 확률을 변화시키면서 대전 액션 게임에 적용하였다.

개체의 크기를 50부터 950까지 50 단위로 19 단계로 적용했을 때 개체수 300에서 90 세대만에 최적해를 구하였다. 교배 확률은 0.7에서 1까지 변화시키면서 적용한 결과 확률 1에서 90 세대만에 최적해를 구하였다. 돌연변이 확률은 0.0005부터 0.0024 까지 변화시키면서 적용했을 때 확률 0.01일 때 90세대에서 최적해에 도달하였다.

본 논문에서 유전자 알고리즘을 적용한 대전 액션 게임에서 가장 좋은 조건의 파라미터 조합은 개체수는 300, 교배확률은 1, 돌연변이 확률은 0.01이다. 이러한 조합을 찾아내는 것은 몇 번의 실험으로 찾아지는 것이 아니라 반복적으로 다양하게 파라미터 값을 변화시키면서 수렴점을 찾아야 하는 어려움이 있다. 유전자 알고리즘을 적용 할 때 이 방법을 적용한다면 파라미터 값에 대한 영향을 미리 알 수가 있어 실행시간을 단축할 수가 있다. 본 연구는 유전자 알고리즘을 응용하는 분야에서 기초 자료로 이용될 수 있을 것으로 예상된다.

References

[1] Adalberto Ramirez Muz, *Handbook of genetic algorithm: New research*, Nova, 2012.

[2] Rick Riolo, and Terence Soule, *Genetic programming theory and practice VI*, Genetic and Evolutionary Computation, Vol. 6, Dec. 2, 2008.

[3] Vose, Michael D. *The simple genetic algorithm*, Mit Press, 2007.

[4] Penelope Sweetser, and Peta Wyeth, *A model for evaluating player enjoyment in games*, Game Developers Conference, 2004.

[5] Tho, Dang Xuan, *Genetic algorithms and application in examination scheduling*, Grin Verlag, 2013.

[6] Zhang G, Gao L, and Shi Y. *An effective genetic algorithm for the flexible job-shop scheduling problem*, Expert Syst Appl 2011.

[7] Goldberg, *Genetic algorithm in search, optimization and machine learning*, Addison-Wesley Publishing Co.Inc., N.Y., 1989.

[8] Goldberg, David. *The design of innovation: lessons from and for competent genetic*

algorithms, Norwell, MA: Kluwer Academic Publishers. 2002.

[9] Juul. J Working with the Player's Repertoire. *International Journal on Intelligent Games and Simulation*, pp. 54-61. 2004.

[10] W. Sterren, Tactical Path-Finding with GA, in *Game Programming Gem 3*, Boston: Charles River Media, pp. 294-306. 2002.

[11] Myunsub Lee, *Intelligent characters for fighting action games using genetic algorithm*. Kookmin University. 2005.

액션 게임에서 유전자 알고리즘의 파라미터가 결과에 미치는 영향

이면섭

인천대학교 컴퓨터 공학부

요 약

본 논문에서는 유전자 알고리즘(Genetic Algorithm)의 파라미터 값이 결과에 어떠한 영향을 미치는지 알아보았다. 실험 대상으로는 10개의 행동을 하는 액션 게임에 적용하였다. 액션 게임에서 지능 캐릭터는 세대(Generation)가 거듭될수록 스스로 진화하여 최적의 해(optimal solution)를 찾도록 하였다. 실험에서는 유전자 알고리즘의 여러 개의 파라미터 중에서 개체의 크기(Population size), 교배 확률(Cross rate), 돌연변이 확률(Mutation rate)을 변화시키면서 적용하였다. 개체의 크기(Population size)의 범위는 50부터 950까지 하였으며, 교배 확률(Cross rate)은 0.7%에서 1%까지, 돌연변이 확률(Mutation rate)은 0.005%부터 0.24% 까지 변화시켰다. 실험 결과 90 세대만에 최적해를 찾았다. 가장 빠른 시간에 최적해를 찾는 파라미터 조합은 개체의 수(Population size)는 300, 교배확률(Cross rate)은 1, 돌연변이 확률(Mutation rate)은 0.01 이다. 이러한 조합을 찾아내는 것은 몇 번의 실험으로 찾아지는 것이 아니라 반복적으로 다양하게 파라미터 값을 변화시키면서 실험을 해야 하는 어려움이 있다.



Myun Sub Lee received the B.S. degree in Electrical Engineering from Kookmin Univ. in 1985. He received the M.S. degree in the Electrical Engineering from Inha University in 1987 and the Ph.D. degree in the Department of Computer Engineering from Kookmin University in 2005. He was a professor in Dept. of Computer Science and Engineering at Incheon University from 1990 to 2014. His current research interests include artificial intelligence, intelligent Games.

E-mail address: nantian@incheon.ac.kr