

유전 알고리즘을 이용한 삼목 게임 전략 분석

이병두

세한대학교 체육학부 바둑학과

blee026@korea.com

Analysis of Tic-Tac-Toe Game Strategies using Genetic Algorithm

Byung-Doo Lee

Dept. of Baduk Studies, Division of Sports Science, Sehan University

요 약

바둑은 단순한 규칙에도 불구하고 매우 복잡한 전략보드 게임이다. 몬테카를로 트리탐색을 이용하여 컴퓨터 바둑 프로그램들이 접바둑으로 프로기사를 제압해 왔다. 몬테카를로 트리탐색은 전략의 개념보다는 몬테카를로 시뮬레이션에 의해 계산된 승률에 근간을 한다. 반면에 적절한 적합도 함수로 된 유전 알고리즘은 게임 내 최적 해를 찾아낼 수 있다. 삼목 게임(또는 ○× 게임)은 가장 대중적인 게임 중의 하나이다. 저자는 삼목 게임에서의 최선의 전략을 찾고자 했다. 실험 결과로 유전 알고리즘은 효율적인 전략들을 찾을 수가 있으며, 바둑과 서양장기와 같은 여타 보드게임들에 적용할 수 있음을 보였다.

ABSTRACT

Go is an extremely complex strategy board game despite its simple rules. By using MCTS, the computer Go programs with handicap game have been defeated human Go professionals. MCTS is based on the winning rate estimated by MC simulation rather than strategy concept. Meanwhile Genetic algorithm equipped with an adequate fitness function can find out the best solutions in the game. The game of Tic-Tac-Toe, also known as Naughts and Crosses, is one of the most popular games. We tried to find out the best strategy in the game of Tic-Tac-Toe. The experimental result showed that Genetic algorithm enables to find efficient strategies and can be applied to other board games such as Go and chess.

Keywords : Go(바둑), Tic-Tac-Toe(삼목), Genetic algorithm(유전 알고리즘), MCTS (몬테카를로 트리탐색), Strategy(전략), Fitness function(적합도 함수)

Received: Nov. 12, 2014 Accepted: Dec. 11, 2014
Corresponding Author: Byung-Doo Lee (Sehan University)
E-mail: blee026@korea.com

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1598-4540 / eISSN: 2287-8211

1. 서론

인간이 만든 게임 중 컴퓨터가 아직까지 제압 못하고 있는 마지막 게임인 바둑은 대략 2,500년 전에 중국에서 기원된 2인-제로섬-완전정보게임이다[1,2]. 바둑은 일반적으로 포석, 중반전, 끝내기 등의 세 단계로 구성되어 있다. 포석에서는 전략(strategy) 개념이 중요하며, 중반전에서는 전술(tactics) 개념이 중요하다. 마지막 단계인 끝내기에서는 선수(先手), 후수(後手), 집 크기 계산 등의 개념이 중시된다[3].

1990년대부터 전 세계적으로 개발된 컴퓨터바둑에 있어, 중반전인 끝내는 컴퓨터바둑의 기력(棋力) 수준이 프로기사와 대등하게 되었으며, 중반전은 상당한 아마 수준까지 이르고 있으나, 초반전인 포석에 대한 연구는 미진한 편이다[4-7]. 그간 컴퓨터바둑의 기력 향상을 위해 많은 인공지능 연구자들은 여러 최신 인공지능 기법을 적용하였으나, 그 기력이 프로기사에 못 미친다는 사실을 알게 되어 새로운 방법을 고대하고 있었다[3].

그 대안책으로 몬테카를로 트리탐색(MCTS: Monte-Carlo Tree Search)을 바둑에 적용한 결과, 마침내 2006년에 Crazy Stone과 MoGo라는 컴퓨터바둑이 9줄 바둑에서 최초로 프로기사를 제압하였으며, 최근에는 [Table 1]에서 보듯이 19줄 바둑에서도 4점 집바둑으로 프로기사를 제압하고 있는 등 매우 선전하고 있다[1,3,8].

[Table 1] List of the first games won by computer Go programs against human Go professionals[1,8]

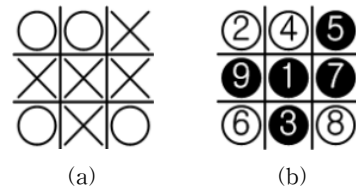
Year	Handicap	Human level	Computer Program
2008	9	8 dan	MoGo
2008	8	4 dan	Crazy Stone
2008	7	4 dan	Crazy Stone
2009	7	9 dan	MoGo
2009	6	1 dan	MoGo
2010	6	4 dan	Zen
2012	5	9 dan	Zen
2013	4	9 dan	Crazy Stone

MCTS는 위치평가함수(position evaluation function)를 필요로 하지 않는 최대우선탐색(best-first search) 방법으로 탐색공간에 대한 무작위적 탐험(randomized exploration)에 근간을 두고 있으나[1,3,9], 바둑 게임에서 중요한 전략과 전술을 적절히 적용할 수 없는 약점을 안고 있다. 이에 저자는 향후 컴퓨터바둑 프로그램 제작에 앞서 전략과 전술을 활용할 수 있는 유전 알고리즘(Genetic algorithm)을 삼목(Tic-Tac-Toe) 게임에 적용시켜 최선의 전략을 갖는 수순(手順)을 찾고자 했으며, 그 결과를 MCTS의 결과와 비교하고자 했다.

2. 본론

2.1 삼목 게임

삼목 게임은 전 세계적으로 잘 알려진 게임 중의 하나이며, 두 대국자(X, O)가 번갈아가며 3×3 칸에 X와 O를 써서 가로, 세로 또는 대각선상에 동일한 모양이 연속하여 3개가 형성되면 이기는 게임이다. 참고로 [Fig. 1](a)는 첫 번째 대국자인 X가 가로 방향으로 승리를 한 장면이다.

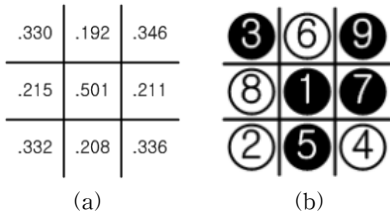


[Fig. 1] (a) A finished Tic-Tac-Toe game won by the first player, X; (b) An exemplary move sequence of the game in the form of (a)

[Fig. 1](b)는 전략을 고려하여 [Fig. 1](a)의 형태가 될 수 있는 X와 O의 수순을 흑돌과 백돌위에 표시를 한 예가 된다. 저자는 삼목에서 전략을 고려한 수순을 표시하기 위해 [Fig. 1](a)와 같은 X와 O의 형태가 아닌 [Fig. 1](b)와 같이 흑돌과 백돌 위에 수순을 표시하고자 한다.

2.2 패턴 분석

삼목 게임은 9곳을 차지할 수 있다. 한 곳에 대해 흑돌, 백돌, 공배로 처리되기 때문에 전체 게임 복잡도는 총 $19,683 (= 3^9)$ 이 된다. 삼목에서 최선의 게임 결과를 얻기 위해서는 [Fig. 2](a)에서 보듯이 100,000번의 MCTS 시뮬레이션 결과 첫 수가 중앙인 경우 50.1%, 귀인 경우 33.6%, 변인 경우 20.6%의 평균 승률을 갖기 때문에 첫 수로 중앙에 두어야 한다[1,3].



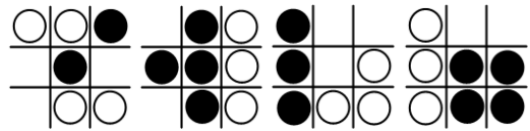
[Fig. 2] (a) Winning rates of each position and (b) the best move sequence after 100,000 simulations with MCTS[3]

참고로 [Fig. 2](b)는 [Table 2]에서 보듯이 MCTS를 이용하여 흑인 최대 대국자(Max player)와 백인 최소 대국자(Min player)에 대해 최소-최대 탐색을 통해 구해낸 최선의 수순을 표시한 것이다[3].

결국 첫 대국자인 흑이 중앙에 둔 경우의 게임 복잡도는 총 $6,561 (= 3^8)$ 이 된다. 이 중 게임에서

발생될 수 있는 패턴을 구하기 위해서는 다음과 같은 두 가지 경우는 성립되지 않기 때문에 제거되어야 한다.

(경우 1) [Fig. 3](a)에서 보듯이 흑이 첫 번째 게임자이기에 흑의 착점 수보다 백의 착점 수가 많은 경우는 성립되지 않는다.



(a) case 1 (b) case 2 (c) case 3 (d) case 4

[Fig. 3] Abnormal patterns

결국 (경우 1)로 제거된 수는 [Table 3]에서 보듯이 총 4,438개가 되어 남아있는 정상적인 패턴은 총 2,123개가 된다.

[Table 3] Number of normal patterns

		no. of eliminated patterns	no. of remained patterns
Initial			6,561
Elimination	case 1	4,438	2,123
	case 2	286	1,837
Total		4,724	1,837

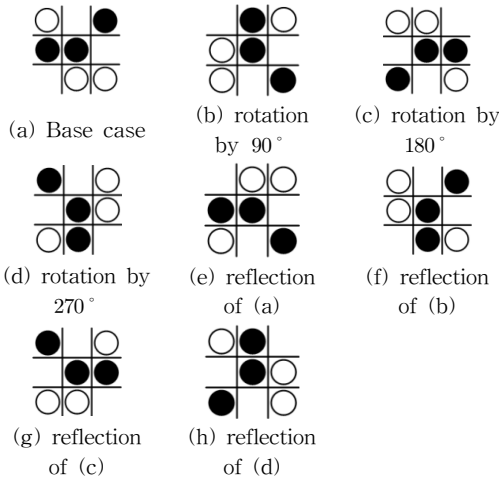
(경우 2) [Fig. 3](b)와 같이 흑백이 동시에 이긴 경우와 [Fig. 3](c)와 같이 흑이 이겼는데 흑과

[Table 2] The rewards of each position generated after 100,000 simulation[3]

	pos 0	pos 1	pos 2	pos 3	pos 4	pos 5	pos 6	pos 7	pos 8
initial	.330	.192	.346	.215	.501	.211	.332	.208	.336
move 1	.430	.578	.435	.578	B_1	.567	.423	.559	.434
move 2	.519	.367	.169	.516	B_1	.377	W_2	.512	.515
move 3	B_3	.664	.740	.765	B_1	.625	W_2	.325	.004
move 4	B_3	.005	-.171	-.338	B_1	.010	W_2	.504	W_4
move 5	B_3	.335	.668	.665	B_1	.337	W_2	B_5	W_4
move 6	B_3	W_6	.000	.500	B_1	.501	W_2	B_5	W_4
move 7	B_3	W_6	1.000	.000	B_1	B_7	W_2	B_5	W_4
move 8	B_3	W_6	.000	W_8	B_1	B_7	W_2	B_5	W_4
move 9	B_3	W_6	B_9	W_8	B_1	B_7	W_2	B_5	W_4

백의 착점 수가 같은 경우, 또는 [Fig. 3](d)와 같이 백이 이겼는데 흑의 착점 수가 백보다 많은 경우는 성립되지 않는다. 그러므로 (경우 2)로 제거된 수는 [Table 3]에서 보듯이 총 286개가 되어 최종 정상적인 패턴 수는 총 1,837개가 된다.

정상적인 1,837개의 패턴이 삼목에서 발생할 수 있는 전체 경우의 수가 되나, 임의 패턴에 대해 [Fig. 4]에서 보듯이 회전과 상하 대칭으로 생성되는 것은 같은 형태가 되니, 이는 1개의 원형(base case) 패턴으로 처리하면 된다[10,11].



[Fig. 4] Rotations and reflections of a base case

즉 [Fig. 4](a)와 같은 원형패턴에 대해 시계방향으로 90°, 180°, 270° 회전으로 생긴 3개의 경우는 제거되어야 하며, 아울러 원형패턴과 회전에 의해 생긴 3개의 패턴들에 대한 상하 대칭인 4개 역시 제거되어야 한다. 결국 회전과 상하 대칭으로 제거된 수는 [Table 4]에서 보듯이 총 1,582개가 되어, 잔여 원형패턴의 총 수는 255개가 된다.

[Table 4] Number of base cases

		no. of eliminated patterns	no. of remained patterns
Initial			6,561
Elimination	abnormal patterns	4,724	1,837
	rotations/reflections	1,582	255
Total		6,306	255

또한 정상패턴 1,837개와 원형패턴 255개에 대해 게임깊이(game depth)에 따른 게임 종료(흑승, 백승, 무승부)와 게임 미종료로 분류해 보면 [Table 5]와 같다.

2.3 가능 전략 분석

[3]은 삼목 게임에서 최대의 승률을 보장하는 중앙에 착점을 한 후, 최상의 전략에 대한 수순을 찾기 위해 [Table 2]와 같이 MCTS를 이용한

[Table 5] Game results of normal patterns and base cases by game depth

		depth 1	depth 2	depth 3	depth 4	depth 5	depth 6	depth 7	depth 8	depth 9	total
Normal patterns	Black win	0	0	0	0	60	0	264	0	36	360
	White win	0	0	0	0	0	36	0	48	0	84
	Draw	0	0	0	0	0	0	0	0	8	8
	Not finished	1	8	56	168	360	444	256	92	0	1,385
	Total	1	8	56	168	420	480	520	140	44	1,837
Base cases	Black win	0	0	0	0	12	0	35	0	6	53
	White win	0	0	0	0	0	5	0	6	0	11
	Draw	0	0	0	0	0	0	0	0	1	1
	Not finished	1	2	8	24	48	59	35	13	0	190
	Total	1	2	8	24	60	64	70	19	7	255

100,000번의 시뮬레이션으로 생성된 평균 승률값을 중심으로 최소-최대 탐색을 수행하여 최선의 수순이 [Fig. 2](b)와 같으며 그 결과는 무승부가 됨을 밝혔다.

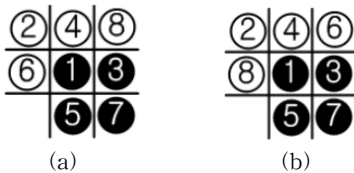
[Table 5]를 활용하여 게임이 종료된 정상패턴들에 대해 가능한 전략의 수를 살펴보자. 게임 깊이가 홀수인 경우의 마지막 착수는 흑이 되며, 짝수인 경우는 백이 된다. 한 예로 게임 깊이가 5인 경우의 착수는 흑(중앙 착점)→백→흑→백→흑이 되며, 전략의 수를 계산해 보면 다음과 같이 4개가 된다.

$$1(\text{흑}) \times 4(\text{착수(백)}) \times 4(\text{착수(흑)}) \times 4(\text{착수(백)}) \times 4(\text{착수(흑)}) \\ = 1 \times 4 \times 4 \times 4 \times 4$$

동일한 방법으로 게임 깊이에 따른 승부가 결정된 정상패턴들의 전략의 수를 살펴보면 다음과 같다.

- (깊이 5: 흑승) $4 \times 60 = 240$
- (깊이 6: 백승) $12 \times 36 = 432$
- (깊이 7: 흑승) $36 \times 264 = 9,504$
- (깊이 8: 백승) $144 \times 48 = 6,912$
- (깊이 9: 흑승/무승부) $576 \times (36 + 8) = 25,344$

결국 흑이 첫 수로 중앙에 두는 경우, 흑승 30,480개(= 240 + 9,504 + 20,736), 백승 7,344개(= 432 + 6,912), 무승부 4,608개로 총 42,432개의 전략이 존재함을 알 수 있다. 그러나 한 예로 깊이가 8인 백승에 대해 6,912개의 모든 전략이 유효한 것은 아니다. [Fig. 5](a)는 가능한 전략이 될 수 있으나, [Fig. 5](b)는 이미 백 6으로 게임이 종료되었기 때문에 가능한 전략이 될 수 없다.



[Fig. 5] (a) Possible strategy and (b) impossible strategy in game depth 8

결국 이와 같은 경우를 고려하면 총 42,432개의 전략 중 가능한 전략의 수는 최종 25,872개가 된다.

2.4 전략 적합도 함수

일반적으로 유전 알고리즘의 적합도 함수를 위해 단순히 게임의 결과(승, 패, 무승부)로 정의를 하나, 저자는 게임의 결과를 포함한 전략의 중요성을 감안한 휴리스틱 전략 적합도 함수를 (eq. 1)과 같이 정의하였다.

$$\text{Strategy Fitness} = \sum_{i=1}^l R_i \quad (\text{eq. 1})$$

여기서 R_i 는 게임 깊이가 i 에서의 착수에 대한 보상값(reward)이 되며, l 은 종료된 게임의 깊이가 된다.

즉 공격, 수비, 게임 결과를 고려하여 [Table 6]과 같이 전략을 18가지의 형태로 분류를 하였으며 그에 따른 보상값을 책정하였다.

- (경우 1) 흑의 2중 승리
- (경우 2) 마지막 착수인 흑 9로 이기는 경우
- (경우 3) 2중 수비를 하면서 이기는 경우
- (경우 4) 단순한 승리
- (경우 5) 2중 수비
- (경우 6) 효과적인 수비
- (경우 7) 전략이 없는 수비
- (경우 8) 효과적인 공격
- (경우 9) 단순한 공격
- (경우 10) 전략 없이 귀를 착수
- (경우 11) 전략 없이 변을 착수
- (경우 12) 마지막 착수인 흑 9로 비기는 경우
- (경우 13) 수비 없이 귀를 착수
- (경우 14) 수비 없이 변을 착수
- (경우 15) 2중 수비를 안 하는 경우
- (경우 16) 이기지 않고 다른 곳을 공격
- (경우 17) 이기지 않고 다른 곳을 수비
- (경우 18) 수비 안하고 다른 곳을 공격

[Table 6] Rewards of 18 strategies for strategy fitness function

No	Strategy	Case	Reward	No	Strategy	Case	Reward	No	Strategy	Case	Reward
1	black double win		+11	7	blocking w/o strategy		+6	13	playing corner w/o blocking		-10
2	win with last move		+10	8	effective offense		+5	14	playing side w/o blocking		-10
3	win with double blocking		+9	9	offense		+4	15	playing w/o double blocking		-10
4	win		+8	10	playing corner w/o strategy		+1	16	offense w/o win		-10
5	double blocking		+8	11	playing side w/o strategy		0	17	blocking w/o win		-10
6	effective blocking		+7	12	last move w/o win		0	18	offense w/o blocking		-10

한 예로 [Fig. 6](a)의 최종 전략 적합도(SF: strategy fitness)는 다음과 같이 각 게임 깊이에 대한 보상값을 합친 +39가 된다.

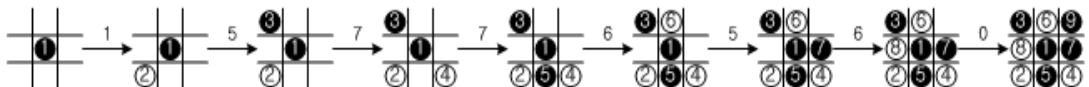
$$SF = 1 + 4 + 7 + 7 + 7 + 7 + 6 + 0 = 39$$

같은 방법으로 [Fig. 6](b)의 최종 전략 적합도는 +36이 되며, [Fig. 6](c)는 -36이 된다.

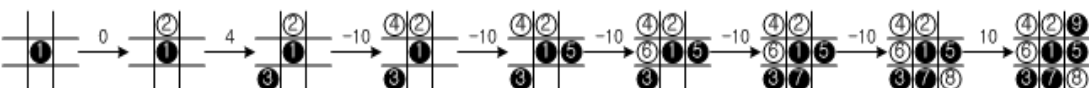
앞서 구한 삼목 게임에서 가능한 25,872개의 전략에 대한 전략 적합도의 분포를 살펴보면 최댓값 +39, 최솟값 -36, 중위값 -3, 평균값 -3.83으로 [Fig. 7]과 같이 분포되어 있음을 알 수 있다.



(a) The best move sequence generated by Genetic algorithm with strategy fitness function

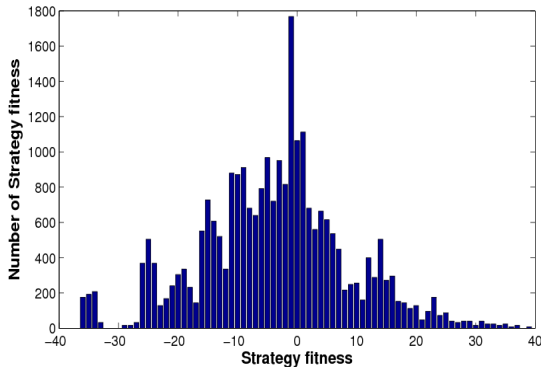


(b) The best move sequence generated by MCTS with 100,000 simulations



(c) The worst move sequence generated by Genetic algorithm with strategy fitness function

[Fig. 6] Sequences of moves generated by Genetic algorithm and MCTS



[Fig. 7] Frequency distributions of strategy fitness

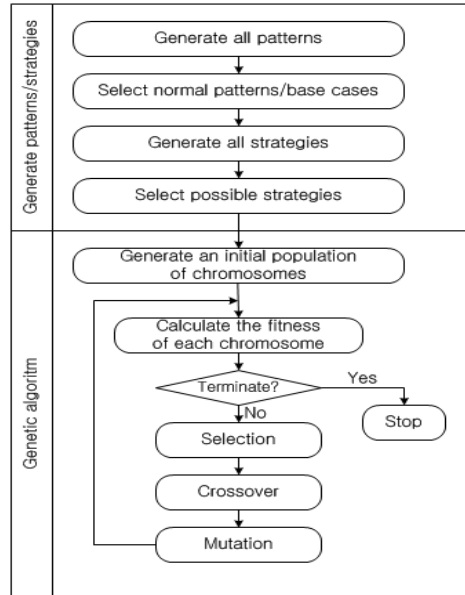
3. 실험

3.1 유전 알고리즘

진화 연산(Evolutionary computation)은 자연선택(Nature selection)과 유전학(Genetics) 계산 모델을 근거로 한 진화적 방법론에 기초를 두고 있으며, 유전 알고리즘(Genetic algorithm), 진화 전략(Evolution strategy), 유전 프로그래밍(Genetic programming) 등을 포함한다. 유전 알고리즘은 다윈의 진화론을 기초로 한 계산 모델로 존 홀랜드(John Holland)에 의해서 제안되었으며 최적화 문제를 해결하는 방법 중의 하나이다[12]. 즉 환경에 잘 적응하는 유전자만을 선택, 교배하며 때에 따라 돌연변이를 하여 차세대에 우수한 형질의 유전자를 생성하는 것이다. 결국 진화가 거듭될수록 주어진 환경에 적합한 유전자들만이 남아 있게 만드는 것이다[13].

3.2 유전 알고리즘 설계

전략을 고려한 유전 알고리즘을 적용하기 위해 본 실험에서는 [Fig. 8]과 같이 두 부분으로 구성하였다. 즉 삼목 게임에서 (1) 가능한 전략을 추출하기 위한 패턴 및 전략 생성 과정과 (2) 유전 알고리즘을 적용하는 과정으로 구분하였다.



[Fig. 8] Flow chart for implementing Genetic algorithm

앞서 설명했듯이 중앙에 흑돌이 선착을 한 경우 총 6,561개의 패턴이 생성되며, 이 중 성립되지 않는 경우를 배제하면 총 1,837개의 정상패턴과 255개의 원형패턴이 추출된다. 정상패턴 중 전략을 고려하면 42,432개의 전략이 되나 이 중 불가능한 전략을 제외하면 최종 25,872개의 전략이 존재함을 설명했다.

[10,11]은 삼목 게임에서의 최선의 전략을 구하기 위해 삼목에서 발생하는 패턴을 분석하여 원형패턴을 구한 후, 유전 알고리즘을 이용하여 최적의 전략을 찾고자 했다. 또한 최선의 전략 우선순위로 (1) 상대방의 삼연형(三連形)을 우선적으로 봉쇄하고, (2) 중앙이 비워져 있으면 그 곳을 두며, (3) 중앙에 비워져 있지 않으면 귀, 변 순으로 둘 것을 제안했다. 저자 역시 유사한 방법을 택했으나 주요 차이점으로는 (1) 전략에 대한 실제 보상값을 전략 적합도 함수에 적용하였으며, (2) 유전 알고리즘을 통해 구한 최적해와 MCTS를 통해 구해낸 최적해를 비교하여 결론을 유도하였다.

3.3 염색체 해집단 생성

난수를 이용하여 25,872개의 전략 중 $N = 250$ 개 염색체를 포함하는 해집단(population)을 (eq. 2)와 같이 생성시켰다.

$$P = \{p_1, p_2, \dots, p_{popsize}\} \quad (\text{eq. 2})$$

여기서 $popsize$ 는 250이 되며, 각 염색체 p_i 는 착수에 대한 위치, 최종 게임값이, 게임결과, 적합도, 상대적합도, 누적상대적합도 정보를 포함하고 있다.

3.4 평가

해집단 내에 있는 각 염색체는 정의된 전략 적합도 함수에 의거 적합도 평가가 이루어진다. 게임결과 및 전략의 중요성을 감안한 (eq. 1)과 같은 전략 적합도 함수를 적용하였다.

3.5 선택

우수한 염색체를 생성하기 위한 유전자 선택 방법으로 룰렛 휠(roulette wheel) 선택 방법을 사용하였다. 일반적인 유전자 선택 방법은 해집단 내에 있는 두 염색체를 선택하여 유전자 조작인 교차(crossover) 과정을 수행한다. 선택을 위한 염색체 p_i 에 대한 선택확률 q_i 는 (eq. 3)과 같다[14].

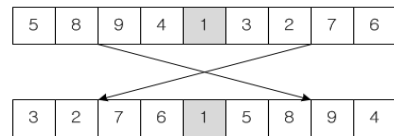
$$q_i = \frac{f(p_i)}{\sum_{j=1}^{popsize} f(p_j)} \quad (\text{eq. 3})$$

즉 적합도 $f(p_i)$ 의 값이 클수록 선택될 확률이 높게 된다.

3.6 교차

교차는 선택된 두 부모 염색체의 정보를 교환하

는 것이다. 본 실험에서는 단순하게 한 개의 부모 염색체를 선택한 후 [Fig. 9]와 같이 착수 1을 중심으로 좌우 수순의 값을 교환하여 자식 염색체를 생성하였다. 이때 생성된 자식 염색체가 불가능한 전략인 경우에는 변이과정을 거치도록 하였다. 교차에 사용된 교차율은 $p_c = 0.5$ 가 된다.



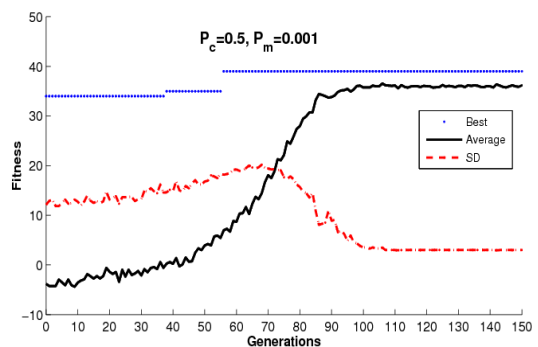
[Fig. 9] Crossover of a chromosome

3.7 변이

변이는 염색체의 유전자를 바꾸는 작업이 된다. 본 실험에서는 변이를 위해 새로운 염색체를 25,872개의 가능 전략으로부터 임의로 1개를 선택하였다. 변이에 사용된 변이율은 $p_m = 0.001$ 이 된다.

3.8 실험 결과

결국 25,872개의 전략 중 최선의 전략을 찾기 위해 해집단의 크기 $N = 250$, 교차율 $p_c = 0.5$, 변이율 $p_m = 0.001$, 150세대(generation)에 걸쳐 유전 알고리즘을 적용한 결과의 성능 그래프는 [Fig. 10]과 같았다.



[Fig. 10] Performance graph for 150 generations of 250 chromosomes with crossover rate 0,5 and mutation rate 0,001

[Fig. 10]에서 보듯이 최대 적합도(best fitness)는 +34, +35, +39로 단계적으로 증가했으며, 평균 적합도(average fitness)는 -3.8에서 +35.9로 수렴했으며, 표준편차(SD: standard deviation) 역시 12.1에서 3.0으로 수렴함을 알 수 있었다. 결국 유전 알고리즘을 적용한 결과 삼목 게임에서 최선의 전략을 갖는 수순은 전략 적합도가 +39인 [Fig. 6](a)가 되며, 반면에 최악의 전략을 갖는 수순은 전략 적합도가 -36인 [Fig. 6](c)가 됨을 알 수 있다. 한편 [3]이 밝힌 MCTS를 이용한 최선의 수순인 [Fig. 6](b)의 전략 적합도는 +36이 되는 것을 알 수 있다. 이는 전략 적합도에 직접적인 영향을 받는 유전 알고리즘과 승리 확률에 직접적인 영향을 받는 몬테카를로 트리탐색에 있어 각각 최선의 전략 차이가 있을 것이라는 당연한 사실을 확인했으며, 아울러 유전 알고리즘을 통해 새로운 최선의 전략을 찾아낼 수 있음도 보였다.

4. 결론 및 제언

4.1 결론

[3]은 MCTS를 통해 삼목 게임에서 최선의 승리를 보장받기 위해 첫수로 중앙에 두어야 한다는 것과, 아울러 최선의 수순은 [Fig. 6](b)와 같다고 밝혔다. 저자는 단순히 승리 확률을 갖고 수순을 찾아내는 MCTS의 한계를 극복하기 위하여 전략의 보상값을 부여한 유전 알고리즘을 삼목 게임에 적용하였다. 전략의 보상값에 전적으로 영향을 받는 유전 알고리즘의 단점에도 불구하고 [Fig. 6](a)와 같은 최선의 새로운 수순이 삼목 게임에 잠재되어 있다는 사실을 밝혀냈다.

이는 현재 개발되고 있는 컴퓨터바둑에서 단순한 MCTS 방법이 아닌 전략의 의미를 적용하여 새로운 전략기반 유전 알고리즘을 컴퓨터바둑에 적용할 수 있다는 새로운 사실을 알아냈다.

4.2 제언

유전 알고리즘을 19×19 바둑판에 적용하는 것은 매우 방대하고 난해한 작업이 된다. 대신에 문제영역이 다소 적은 사활문제나 바둑판의 규모가 작은 9×9 바둑판 등에 적용하는 것은 단순 실험을 위해 현실성이 있어 보인다. 이 역시 어떤 전략을 적용할 것인가와 그에 대한 보상값을 어느 정도 부여할지 여부가 어려운 과제로 남아있게 된다.

ACKNOWLEDGMENTS

This paper was supported by the Sehan University Research Fund in 2014.

REFERENCES

- [1] B.D Lee, "Monte-Carlo Tree Search Applied to the game of Tic-Tac-Toe", Journal of Korea Game Society, Vol. 14, No. 3, pp. 47-54, 2014.
- [2] B.D. Lee and J.W. Park, "Applying Principal Component Analysis to Go Openings", Journal of Korea Game Society, Vol. 13, No. 2, pp. 59-70, 2013.
- [3] B.D. Lee, "The best move sequence in playing Tic-Tac-Toe game", Journal of The Korean Society for Computer Game, Vol. 27, No. 3, pp. 11-16, 2014.
- [4] B.D. Lee, "Analysis of Korean, Chinese and Japanese Pro Go Player's Openings", Journal of Korean Society for Computer Game, Vol.26, No.4, pp. 17-26, 2013.
- [5] B.D. Lee, "Comparison of LDA and PCA for Korean Pro Go Player's Opening Recognition", Journal of Korea Game Society, Vol.13, No.4, pp. 15-24, 2013.
- [6] B.D. Lee, "Korean Pro Go Player's Opening Recognition Using PCA", Journal of Korean Society for Computer Game, Vol.26, No.2, pp. 228-233, 2013.

- [7] B.D. Lee, “Applying Neuro-fuzzy Reasoning to Go Opening Games”, Journal of Korea Game Society, Vol.9, No.6, pp. 117-126, 2009.
- [8] Wikipedia, “Computer Go”, from http://en.wikipedia.org/wiki/Computer_Go, 2014.
- [9] M.H.M. Winands and Y. Brórnsson, “Evaluation Function Based Monte-Carlo LOA”, from <http://www.ru.is/~yngvi/pdf/WinandsB09.pdf>, 2014.
- [10] G. Hochmuth, “On the Genetic Evolution of a Perfect Tic-Tac-Toe Strategy”, from <http://www.genetic-programming.org/sp2003/Hochmuth.pdf>, 2014.
- [11] A. Bhatt, P. Varshney and K. Deb, “In Search of No-loss Strategies for the Game of Tic-Tac-Toe using a Customized Genetic Algorithm”, GECCO’08(Genetic and Evolutionary Computation Conference 2008), pp. 889-896, 2008.
- [12] Wikipedia, “Genetic Algorithm”, from http://ko.wikipedia.org/wiki/%EC%9C%A0%EC%A0%84_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98, 2014.
- [13] I.J. Ahn and I.K. Park, “Design of Omok AI using Genetic Algorithm and Game Trees and Their Parallel Processing on the CPU”, Journal of the Korea Information Science Society, Vol. 37, No. 2, pp. 66-75, 2010.
- [14] S.H. Ling and H.K. Lam, “Playing Tic-Tac-Toe Using Genetic Neural Network with Double Transfer Functions”, Journal of Intelligent Learning Systems and Applications, Vol. 3, pp. 37-44, 2011.



이 병 두 (Lee, Byung Doo)

1982 한양대 원자력공학 학사
1991 서강대 정보처리학 석사
2005 Auckland University 컴퓨터공학 박사
2012~현재 세한대 체육학부 바둑학과 조교수
관심분야 : 컴퓨터공학, 인공지능, 컴퓨터바둑