

중분화 알고리즘을 이용한 다양한

진화 체커 플레이어의 생성

김경중 조성배

연세대학교 컴퓨터과학과

uribyul@candy.yonsei.ac.kr sbcho@cs.yonsei.ac.kr

Generation of Diverse Evolutionary Checkers Players Using Speciation Algorithm

Kyung-Joong Kim and Sung-Bae Cho

Dept. of Computer Science, Yonsei University

요 약

최근 진화 알고리즘을 이용하여 건축, 예술, 항공, 기계 분야의 제품을 설계하려는 시도가 다양하게 이루어지고 있으며, Digenetics는 진화 체커 프로그램을 컴퓨터 전시회에서 공개하고 판매에 들어갔다. 게임은 상업적으로도 가치가 높은 소프트웨어 산업분야의 중요한 영역이고, 인공지능 분야의 중요한 연구대상이다. 반면, 다양한 전문지식과 고속의 하드웨어만을 이용한 인공지능 구현의 한계로 인해 상업화되는데 어려움이 있었다. 진화 알고리즘은 개체의 우수성을 평가할 수 있는 방법만으로도 훌륭한 게임 플레이어를 생성해 낼 수 있어 많은 관심을 끌고 있다. 본 논문에서는 중분화의 일종인 크라우딩 알고리즘을 이용하여 다양한 체커 전략을 생성한 후 결합을 통해 일반 유전자 알고리즘을 이용한 전략에 비해 높은 성능을 냈다.

1. 서 론

그림 1은 체커보드 판을 보여준다. 체커는 흑과 백이 12개의 말을 가지고 하는 서양게임이다. 대표적인 체커 프로그램으로 Chinook이 있으며, 세계 최고의 체커 챔피언을 물리친 프로그램으로 유명하다[1]. <http://www.cs.ualberta.ca/~chinook/>을 통해 Chinook과 체커 게임을 둘 수 있다. Chinook의 특징은 게임 트리를 사용했으며, 엔드-게임 데이터베이스를 사용하였다는 점이다.

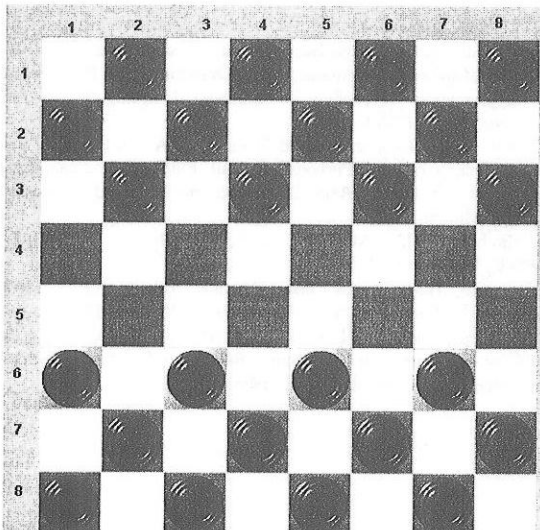


그림 1. 체커 보드

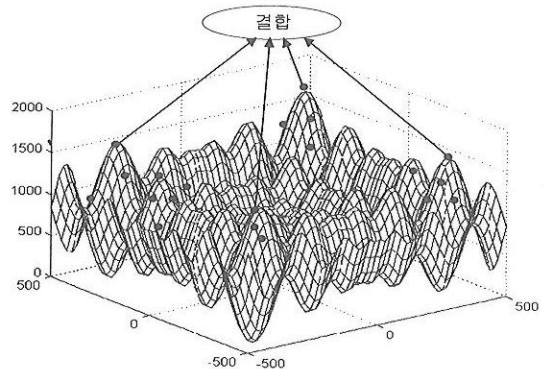


그림 2. 중분화를 이용한 다양한 해의 탐색과 결합

체커를 두는 방법을 좀 더 자세히 설명하면 다음과 같다. 가로 8줄, 세로 8줄의 보드위에 흑과 백으로 나뉘어진 알들이 12 개씩 주어진다. 체커 게임에서 이기기 위해서는 상대방의 알들을 모두 죽이든지 상대방이 움직일 수 없게 하면 된다. 체커 게임의 무승부는 일정한 수이상을 넘었음에도 승부가 가리워지지 않은 경우 내려진다.

흑과 백의 알은 1번에 대각선으로 1칸씩 움직일 수 있으며, 상대방의 마지막 줄에 도달하면 "왕"이 된다. "일반" 알은 한번에 한칸만을 움직일 수 있고 앞으로 전진만 가능하다. 반면 "왕"은 한번에 한칸을 움직이지만 앞 뒤로 움직일 수 있다. 상대방알이 대각선 위치에 있을 경우 점프를 통해 상대방 알을 뛰어넘을 수 있으며 이 경우 상대방 알은 보드에서 제거된다.

진화 알고리즘은 NIPD, 백가본, 체커 등 다양한 게임을 진화적으로 설계하기 위해 연구되고 있다. 본 논문은 진화알고리즘을 이용하여 체커게임 플레이어를 설계하고 중분화 알고리즘을 이용하

여 다양한 전략을 생성한 후 결합을 통해 보다 일반화된 플레이어를 만들어 낸다. 즉, 진화를 통해 탄생한 최고개체만을 이용하는 것이 아니라 종분화를 통해 다양한 개체의 생성을 유도한 후 이들의 협상을 통해 효과적인 전략생성을 유도한다. 그림 2는 진화알고리즘을 이용하여 탐색한 다양한 해들을 결합하는 아이디어를 보여준다. 본 논문은 Chellapilla와 Fogel이 제안한 보드 표현 방법과 신경망 구조를 바탕으로 종분화 알고리즘을 적용하여 보다 나은 성능을 유도하였다.

2. 진화 전략의 생성

전략은 사용자가 게임을 진행하는 동안에 상대편을 제압하기 위한 방법을 나타낸다. 기계와 사람의 차이라면 기계는 전략을 세우기 보다는 모든 가능한 경우를 탐색하여 상대방이 곤란한 상황에 처하거나 기계가 승리하기 쉬운 상황을 골라 움직임을 택한다. 반면 사람은 모든 경우를 탐색하기 보다는 3개에서 4개의 가능한 경우를 두고 고심 끝에 한 수를 두게 된다. 진화를 이용한 체커 게임은 기존의 인공지능 게임프로그램에 비해 전문지식이 필요없다는 점에서 진보하였지만, 게임트리를 통한 탐색에서는 변화가 없다. 아직 인간의 전략생성방법을 모두 따라하기에는 진화방법도 한계가 있다.

게임트리는 최적의 보드를 결정하기 위한 인공지능 방법이다. 게임트리는 현재의 보드에서 가능한 모든 수를 미리 두어 트리를 형성한 후 최적의 움직임을 선택한다. 예를들어 현재의 보드가 있고 이번엔 둘 차례가 기계라고 하면, 기계는 움직일 수 있는 모든 경우를 따진다. 즉, 현재의 보드가 루트가 되고 기계가 움직일 수 있는 모든 가지 수만큼 자식이 생기는 것이다. 각각의 자식노드로부터 사람이 움직일 수 있는 모든 가지 수가 탐색되고 그 수만큼의 자식노드가 생긴다. 탐색하는 수가 깊어질 수록 기계가 다루어야 하는 보드의 수는 폭발적으로 증가한다. 이러한 이유로 탐색 깊이를 제한하는 방법을 사용한다.

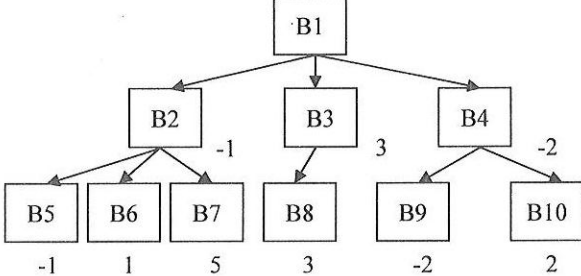
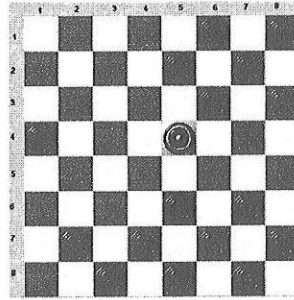


그림 3. Min-Max 알고리즘을 이용한 게임트리의 예

Min-Max 게임트리 탐색방법은 게임트리를 통해 최적의 보드를 결정하기 위한 선택방법으로 이용된다. 각각의 보드는 기계가 이기는 경우 1로, 사람이 이기는 경우 0으로 그리고 불리한 상황의 정도에 따라 0과 1사이에서 평가된다. 탐색 깊이를 3으로 제한하는 경우 모든 잎새노드의 중요도를 0과 1사이로 평가한다. Min-Max 알고리즘은 자식노드의 평가값으로부터 부모노드의 평가값을 결정하는 방법이다. 이때 사람의 움직임일 경우 부모노드의 평가값은 자식노드들의 Min값으로, 기계의 움직임일 경우 자식노드들의 Max값으로 평가하게 된다. 이러한 방법을 통해 루트노드의 보드평가값이 결정되고 루트노드는 Max값을 보인 자식노드로 이동을 선택한다. 그림 3은 B1부터 B10까지의 노드로 이루어진 게임트리의 예를 보여주며, Max값인 B3로 이동하게 된다.

진화 체커게임은 게임전략자체를 진화시키는 것이 아니라 계

입트리의 잎새 노드를 평가하기 위한 신경망을 진화하는 것이다. 하지만, 신경망의 성향에 따라 게임트리의 보드 선택이 달라지기 때문에 일종의 전략이 생성된다고 할 수 있다. 오프-역전과 알고리즘을 이용하여 신경망을 설계하는 것은 높은 성능을 기대할 수 있지만, 게임문제처럼 평가함수가 미분가능하지 않은 경우는 설계자체가 불가능하다. 이러한 이유로 진화알고리즘을 이용한 신경망의 설계가 이용된다. 진화 알고리즘은 미리 정의된 신경망의 구조아래에서 연결강도와 바이어스를 최적화하기 위해 사용된다.



0	0	1	1
0	0	1	1
0	0	1	1
1	0	-K	0
0	0	0	0
0	0	-1	-1
-1	-1	-1	-1
0	-1	-1	-1

그림 4. 보드의 표현

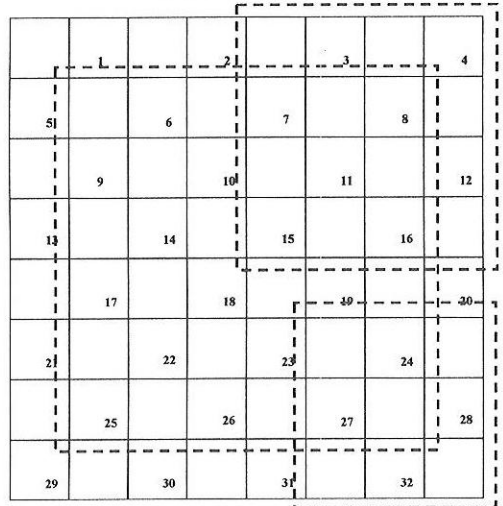


그림 5. 전체 보드의 부분보드 예

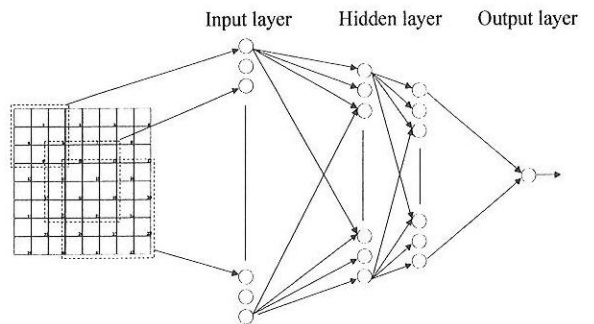


그림 6. 기본 신경망의 구조

신경망의 구조는 91개의 노드를 가지는 입력층, 40개의 노드를 가지는 제 1 은닉층, 10개의 노드를 가지는 제 2 은닉층, 1개의 노드를 가지는 출력층으로 구성된다. 신경망의 입력은 91개이며, 전체 보드의 일부분이 입력으로 사용된다. 하나의 보드는 벡터로 표현되며, 알이 놓여 있는 경우 +로 표시하고, 사람의 알이 놓여 있는 경우 -로 표시한다. "일반" 알의 경우 1로 표시하고 왕이 놓여 있는 경우 K로 표시하는데 K의 값은 사진에 설정한다. 노드로 들어가는 입력값은 보드를 표현한 벡터의 원소들의 합이 된다. 출력은 0과 1사이의 값으로 표현되며 1에 가까울수록 좋은 보드이다. 그림 4는 보드를 벡터로 표현하는 예를 보여준다.

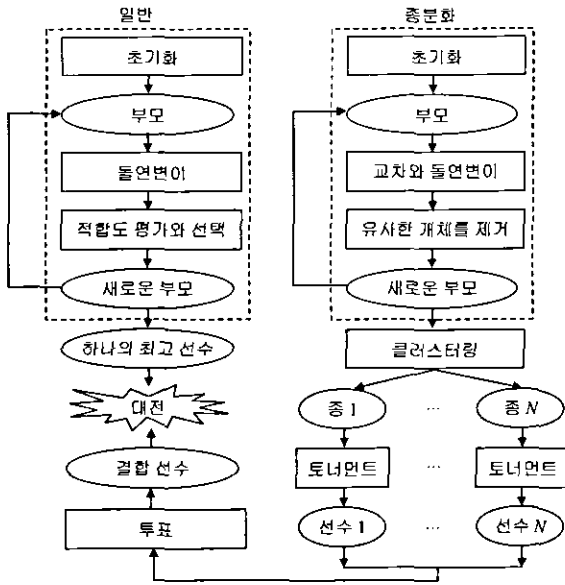


그림 7. 전체 흐름도

신경망의 91개 입력에는 8개의 가로줄과 8개의 세로줄로 이루어진 보드의 일부분이 들어가게 된다. 즉, 3개의 가로줄과 3개의 세로줄로 이루어진 36개의 부분보드, 4개의 가로줄과 4개의 세로줄로 이루어진 25개의 부분보드, ..., 8개의 세로줄과 8개의 가로줄로 이루어진 1개의 전체보드를 합하면 36+25+16+9+4+1=91개가 된다. 그림 5는 보드의 부분보드를 보여주며, 그림 6은 신경망의 구조를 보여준다.

일반 진화 알고리즘 진화전략을 이용한다. 즉, 염색체는 실수로 표현되며, 선택과 돌연변이만으로 진화한다. 반면 종분화 진화 알고리즘은 일반 유전자 알고리즘을 이용하여 진화하며 종분화를 위해 크라우딩 알고리즘을 사용한다. 크라우딩은 부모와 자식중 유사한 개체를 제거하는 방법을 사용하여 종의 다양성을 유지한다[2]. 진화전략을 통해 생성된 최고의 전략과 종분화를 통해 생성된 다양한 전략의 결합이 대전을 통해 상호간의 성능을 비교한다. 종분화를 통해 생성된 마지막 세대로부터 대표개체를 선택하여 결합하게 되는데, 대표개체를 선택하기 위해 밀도에 기반한 클러스터링 방법이 이용된다[3]. 그림 7은 위와 같은 과정을 흐름도로 보여준다. 아래 수식은 진화전략에서 연결가중치를 변화하는 방법을 보여준다. 이 수식에서 s 는 자기 적용 파라미터를 나타내고 w 는 가중치를 나타낸다. $gauss()$ 는 가우스 함수를 나타낸다.

$$s[i] = s[i] \times e^{r \times gauss()}$$

$$w[i] = w[i] + s[i] * gauss()$$

3. 실험 및 결론

표 1은 두가지 서로 다른 체커 전략을 진화하기 위해 사용한 파라미터를 보여주고, 표 2는 두 전략간의 대전결과를 보여준다. 실험 결과 종분화된 전략의 연합이 일반 진화를 통해 생성된 전략에 비해 우수한 성능을 보이는 것을 볼 수 있다. 실험 결과로부터 유추할 수 있는 사항은 일반전략과 종분화 전략의 대결은 일반전략에 비해 1:1의 상황인 경우 엇비슷하지만, 종분화 전략을 결합한 경우 상당히 우수한 성적을 거둔다는 것을 알 수 있다.

최근 다양하게 연구되고 있는 진화를 이용한 게임전략 생성의 일환으로 체커 게임의 전략을 진화클를 통해 생성했다[4,5]. 보다 나은 성능을 위해 종분화 알고리즘을 이용하여 다양한 개체를 탐색하고 결합을 수행했다. 실험결과 종분화 전략의 결합은 일반진화클를 통해 탄생한 개체에 비해 우수한 성능을 보였다.

표 1. 전략을 진화하기 위한 파라미터 설정

	진화 전략	종분화
집단 크기	100	100
돌연변이율	0.01	0.01
교차율	사용안함	1.0
세대수	50	50

표 2. 두가지 전략간의 대전 결과

	1:1	1:20
패배	24	16
승리	22	37
무승부	22	15

감사의 글

이 연구는 과학기술부가 지원한 뇌과학 연구 프로그램에 의해 지원되었음.

참고문헌

- [1] R. Shaeffer, P. Lake, and M. Bryant, "Chinook: The world man-machine checkers champion," *AI Magazine*, vol. 17, no. 1, pp. 21-29, 1996.
- [2] S. Mahfoud, "Nicheing methods for genetic algorithm," *Doctoral Dissertation, University of Illinois Urbana*, 1995.
- [3] M. Easter, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Knowledge Discovery and Data Mining*, pp. 226-231, 1996.
- [4] K. Chellapilla, and D. B. Fogel, "Evolving an expert checkers playing program without using human expertise," *IEEE Transaction on Evolutionary Computation*, vol. 5, no. 4, pp. 422-428, 2001.
- [5] K. Chellapilla, and D. B. Fogel, "Evolution, neural networks, games, and intelligence," *Proc. IEEE*, vol. 87, pp. 1471-1496, Sept 1999.