

# 게임지식을 이용한 종분화 진화 체커 플레이어

김경중, 조성배  
연세대학교 컴퓨터과학과  
{kjkim, sbcho}@cs.yonsei.ac.kr

## Speciated Evolutionary Checkers Player Using Game Knowledge

Kyung-Joong Kim, Sung-Bae Cho  
Dept. of Computer Science, Yonsei University

### 요 약

진화 알고리즘이 최근 게임의 전략을 설계하기 위한 방법으로 많이 활용되고 있다. 이 방법은 후보 전략들이 서로 경쟁하면서 보다 나은 전략을 스스로 찾아나가는 방식으로 체커의 경우 마스터 수준의 컴퓨터 프로그램이 생성되기도 했다. 진화방식은 전문가의 도움을 최소화 하면서 높은 수준의 게임 전략을 생성해 낼 수 있다는 장점이 있는 반면에 손쉽게 정의될 수 있는 널리 알려진 전문지식을 갖는데 매우 많은 세대를 필요로 하기도 한다. 본 논문에서는 잘 정리되어져 있는 체커의 오프닝 및 마무리 지식을 진화 과정 중에 이용하여 진화의 속도와 성능을 향상시키는 방법을 제안한다. 본 연구에서는 일반진화와 종분화 진화 각각에 대해 전문지식을 삽입한 후 결과를 비교해 본다. 실험결과 종분화 진화에 전문지식을 삽입한 경우가 가장 좋은 성능을 보였으며 전문지식의 사용이 진화의 속도 및 성능 향상에 기여하는 것을 확인하였다.

### 1. 서 론

진화 알고리즘은 사람의 간섭을 최소화 하면서 개체들과 환경의 상호작용을 통해 우수한 해를 탐색하는 방법이다. 최근 진화 알고리즘을 이용하여 게임전략을 설계하려는 연구가 진행되고 있다. 개체들 사이의 경쟁만을 이용하여 마스터 수준의 체커 프로그램이 진화되기도 하였다[1]. 일반적으로 게임전략을 설계할 때 활용되는 많은 전문지식을 활용하지 않고 진화에만 의존하여 높은 성능의 플레이어를 탄생시켰다.

이론적으로 진화를 사용하여 기존에 알려진 게임지식을 모두 포함할 수 있는 전략을 만들어 낼 수 있다. 하지만 문제는 그러한 해를 찾아내기 위해 매우 많은 시간이 소요되는 문제점이 있다. 본 논문에서는 기존에 잘 정리되어져 있는 체커 게임의 오프닝 및 마무리 지식을 활용하여 진화에 소요되는 시간을 줄이고 진화 플레이어의 성능을 향상시키는 방법을 제안한다. 기존에 종분화를 사용하여 진화 체커 플레이어의 성능을 향상시키려는 연구가 있었다[2]. 본 연구에서는 일반진화를 사용한 경우와 종분화를 사용한 경우 각각에 대해 전문지식을 진화 과정 중에 삽입하고 최종적으로 생성된 개체의 성능을 비교 평가한다.

### 2. 배경

체커는 전통적으로 8×8보드에서 각자 12개의 돌을 가지고 하는 게임이다. 대각선방향으로 한 번에 한 칸씩 돌을 움직일 수 있으며 상대방 돌이 움직이려는 대각선 칸에 있을 경우 점프하여 잡을 수 있다. 점프를 해야만 하는 상황에서는 반드시 점프를 해야 한다. 일반 돌이 상대편 보드의 맨 끝에 도달하면 왕이 되고 앞뒤로 모두 움직일 수 있다. 그렇지 않으면 일반 돌은 반드시 앞으로만 움직여야 한다. 돌이 모두 잡히거나 더 이상 움직일 수 없다면 게임은 끝난다.

일반적으로 게임은 세 단계로 이루어진다. 게임의 초기에는 복잡도가 상당히 높기 때문에 기존에 알려진 정석을 활용하는 오프닝 단계이다. 오프닝 단계에서는 주로 오랜 기간 동안 최고수준의 선수들이 많은 게임을 통해 확인한 안전한 수순을 밟게 된다. 중간단계에서는 오프닝 단계보다 상대적으로 게임의 복잡도가 줄긴 했지만 여전히 완전하게 게임을 풀기에는 복잡도가 높다. 이 단계에서는 게임트리를 활용하여 앞으로 일어날 수 있는 모든 경우를 제약된 탐색 범위 안에서만 검색하여 각 경우를 평가함수로 점수를 매기고 이를 토대로 현재 둘 수 있는 수를 결정한다. 이 단계에서는 평가 함수의 설계가 가장 중요한 역할을 하며 많은 전문지식이 활용된다. 마무리 단계는 돌의 개수가 몇 개 없기 때문에 대규모 데이터베이스를 활용하여 모든 상황에 대해 게임의 결과를 기록하는 방법을 사용하여 최종 결과를 실시간으로 얻어 낼 수 있다.

게임의 전략을 진화를 사용하여 설계하는 연구는 체커, 오델로, 바둑, 체스, 백가몬, 포커, 블랙잭 등에 이용되었으며 체커와 체스의 경우 상대적으로 높은 수준의 플레이어를 만들기도 했다. 이러한 연구는 일반적으로 많은 진화 시간을 요구하고 몇몇 게임을 제외하고는 성공적인 사례를 보여주지 못하고 있다.

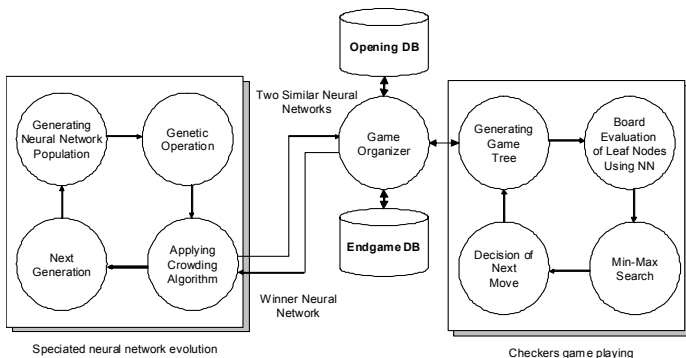


그림 1. 제안한 방법의 개념도

3. 게임지식을 활용한 중분화 진화 체커

그림 2는 제안하는 게임지식을 활용한 진화 체커의 전체 구조를 보여준다. 오프닝 지식은 American Checkers Federation에서 정리한 대표적인 오프닝 80개를 사용하였다. 게임의 중반부분에서는 진화 신경망을 평가 함수로 사용한 게임트리를 이용하여 다음 수를 결정하였다. 진화 신경망은 보드의 좋고 나쁨을 잘 평가하도록 진화하였다. 보드에 있는 돌의 수가 10개 이하인 경우 엔드 게임 데이터베이스를 사용하여 게임의 승/패/무승부 여부를 조회한 후 다음 수를 결정하였다.

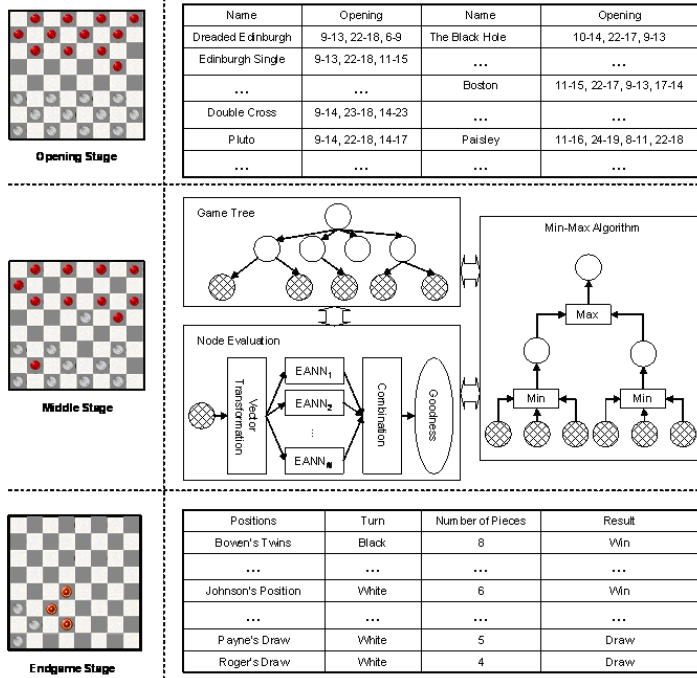


그림 2. 게임지식을 활용한 진화 체커 (게임의 초반에는 오프닝 지식을 활용하고 게임의 중반에는 진화 신경망을 평가함수로 사용한 게임트리를 이용하며 게임의 마지막 부분에서는 엔드게임 데이터베이스를 이용한다.)

3.1 게임트리

게임트리는 현재 보드 상태에서 일어날 수 있는 모든 상황을 트리형태로 나타낸 것이다. 게임트리의 깊이가 2 라면 2수 앞에 일어날 수 있는 모든 상황이 게임트리에 표현된다. 일반적으로 게임트리의 깊이가 깊어지면 트리의 크기가 지나치게 커지기 때문에 탐색 깊이를 제약한다. 제한된 탐색 깊이 아래에서 가장 말단 노드의 좋고 나쁨을 평가함수를 이용해서 측정한다. 이 측정치는 min/max 연산을 통해 상위 노드로 전달되며 다음 수를 결정하는 데 활용된다.

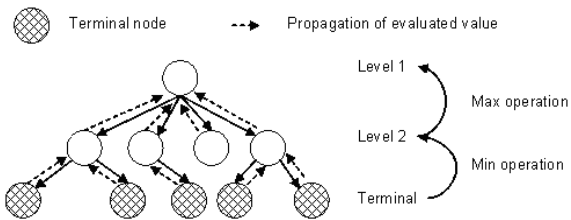


그림 3. 게임트리의 작동원리

3.2 진화 신경망을 사용한 보드의 평가

그림 4는 진화 신경망의 구조를 보여준다. 진화 신경망의 구조는 고정되어 있으며 가중치만 진화를 통해 결정한다. 진화 신경망의 입력은 8x8 보드의 부분보드들이다. 부분보드는 전체 보드에서 4x4, 5x5, 6x6, 7x7, 8x8 크기의 부분적인 보드만을 뽑아낸 것이다. 각 부분보드는 벡터로 변형되며 자신의 돌은 +로 상대방의 돌은 -로 표현되며 일반 돌은 1을 빈칸은 0을 왕은 K를 할당한다. 진화 신경망의 출력은 -1에서 1사이의 실수이며 1에 가까울수록 보드 상태가 좋다는 것을 의미한다.

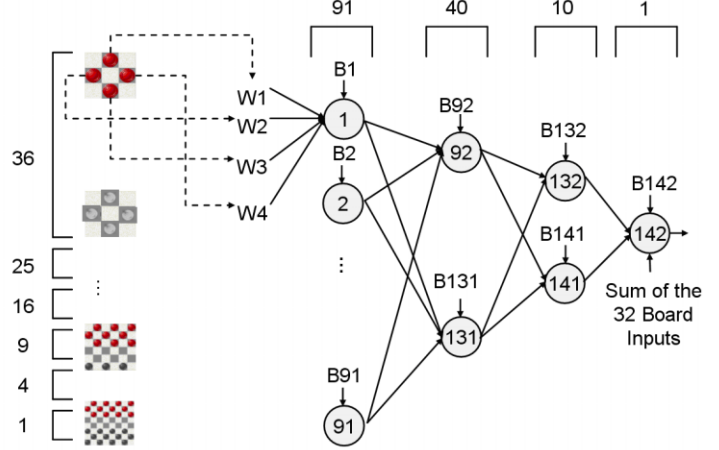


그림 4. 게임 보드를 평가하기 위한 진화 신경망 구조

3.3 진화 탐색

각 신경망은  $K$ 값과 가중치와 바이어스에 대한 자기적응 파라미터를 지니고 있다. 부모  $P_i, i=1, \dots, p$ 로부터 자식  $P_i, i=1, \dots, p$ 이 생성된다.

$$\sigma_i(j) = \sigma_i(j) \exp(\tau N_j(0, 1)), \quad j=1, \dots, N_w$$

$$w_i(j) = w_i(j) + \sigma_i(j) N_j(0, 1), \quad j=1, \dots, N_w$$

$N_w$ 는 신경망에서의 가중치와 바이어스의 개수를 나타낸다.  $\tau = 1/\sqrt{2N_w}$ 로 정의된다.  $N_j(0, 1)$ 은 표준 가우시안 랜덤 함수이다. 자식의  $K$ 값은 다음과 같은 수식으로 얻어진다.

$$K'_i = K_i + \delta$$

$\delta \in \{-0.1, 0, 0.1\}$  사이에서 임의로 골라진다. 그림 5는 진화 알고리즘의 개요를 보여준다.

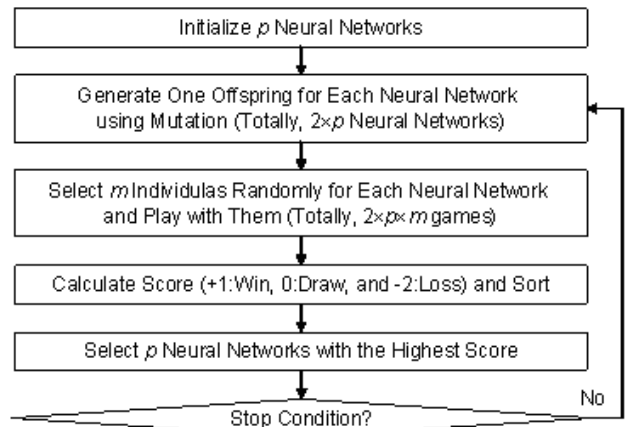


그림 5. 진화 알고리즘의 개요

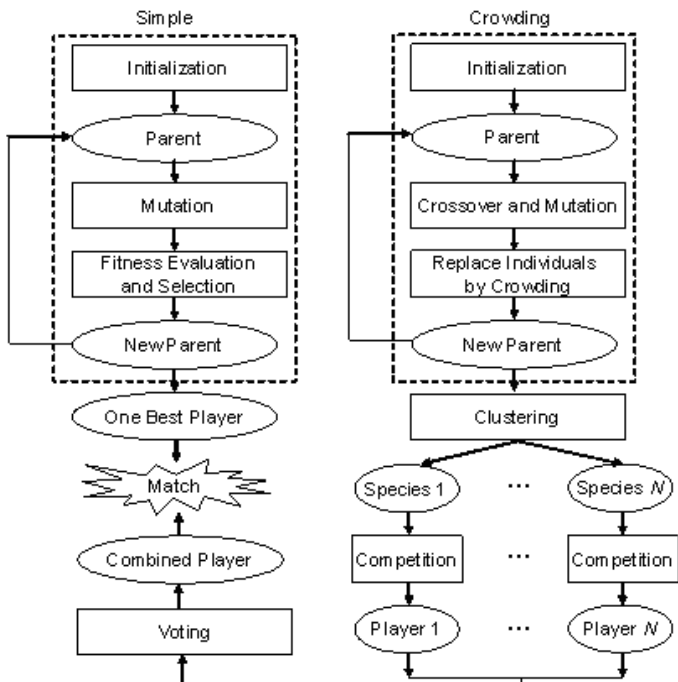


그림 6. 종분화 진화 신경망의 생성

3.3 종분화 진화 신경망의 생성

종분화 진화 신경망은 일반 진화신경망의 성능을 개선하기 위해 사용된다. 단일 신경망 보다는 상호보완이 가능한 여러 개의 신경망을 결합할 때 보다 좋은 성능을 기대할 수 있다. 본 논문에서는 종분화를 사용하여 다양한 개체를 생성한다. 진화가 종료되면 클러스터링을 사용하여 여러 종별로 개체를 묶고 각 종별 대표개체를 동일한 클러스터에 속한 개체들 사이의 경기를 통해 결정한다. 각 종별 대표를 투표방식으로 결합하여 최종 움직임을 결정한다. 그림 6은 종분화 진화 신경망의 전체 개념도를 보여준다.

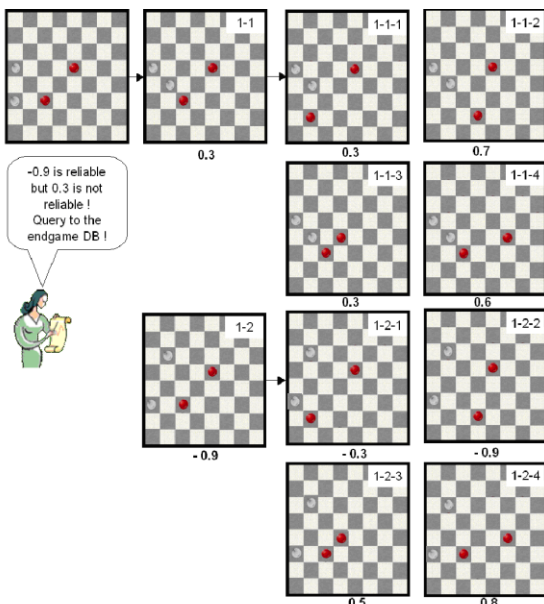


그림 7. 엔드게임 DB의 선택적 활용

3.4 엔드게임 DB의 활용

그림 7은 엔드게임 DB를 활용하는 방법을 설명해 준다. 아래의 수식을 활용하여 엔드게임 DB를 사용할지를 결정한다.  $f$ 는 진화신경망이 평가한 보드에 대한 평가수치이다. 즉 진화 신경망의 평가치가 불확실한 경우에만 엔드게임 DB를 활용한다. 엔드게임 DB는 세계 챔피언 체커 프로그램인 CHINOOK이 사용한 것을 이용하였다.

$$(f > 0.75 \text{ and } f > 0.25) \text{ or } (f < -0.25 \text{ and } f < -0.75)$$

4. 실험 및 결과

집단의 크기는 100으로 하였다. 일반진화의 경우 돌연변이만을 사용하였고 종분화 진화는 교차와 돌연변이를 모두 사용하였다. 종분화 진화의 교차율은 1.0이고 돌연변이율은 0.01이다. 50세대동안 진화시켰다. 그림 8은 진화 시간 비교를 보여준다. 일반진화에 대해 도메인지식을 쓴 경우 진화 속도가 향상되었다. 표 1은 실험결과를 정리해주고 있다. 실험결과 지식을 이용하여 진화한 경우가 일반진화나 종분화 진화보다 향상된 성능을 보였다.

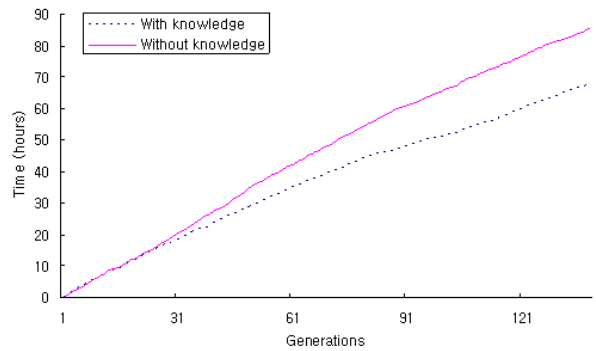


그림 8. 진화속도의 비교

표 1. 대전 결과 (승/패/무승부)

지식	일반 진화	Generations				
		1~15	16~30	31~45	46~60	Total
Red	White	5/0/10	3/3/9	3/0/12	5/3/7	16/6/38
White	Red	4/3/8	4/2/9	5/4/6	4/2/9	17/11/32

지식	종분화 진화	Generations				
		1~15	16~30	31~45	46~60	Total
Red	White	5/1/9	4/3/8	6/0/9	8/2/5	23/6/31
White	Red	7/3/5	5/2/8	8/4/3	6/2/7	26/11/23

참고문헌

[1] D. B. Fogel, *Blondie24: Playing at the Edge of AI*, Morgan Kaufmann, 2001.  
 [2] K.-J. Kim, and S.-B. Cho, "Evolving speciated checkers players with crowding algorithm," *Proc. of Congress on Evolutionary Computation*, vol. 1, pp. 407-412, 2002.