
확률론에 근거한 정통 마작의 패 평가 알고리즘

A Tile Evaluating Algorithm for Orthodoxy Mahjong Based on a Probability Theory

김상진

경운대학교 컴퓨터공학과

Sang-Jin Kim(sjkim@ikw.ac.kr)

요약

본 논문을 통해 정통마작 게임 중의 자신이 가진 각 패의 값어치를 평가하기 위한 확률적인 방법을 소개한다. 마작은 현재 지구상에서 가장 인기 있는 보드게임의 하나이다. 마작게임에 있어서 가장 중요한 기술은 자신이 가진 패들 중 가장 값어치가 적은 패를 골라 버리는 방법이다. 소유한 각 패의 중요도를 구하기 위하여 먼저 각 패를 중복 사용하여 구성 가능한 모든 조를 생성한다. 이들 중 소유한 패로 구성 가능한 조 조합의 후보들을 구한다. 마지막으로 이들 후보 중 가장 유리한 후보를 택하고 이 후보를 구성하는데 사용된 조와 각 패의 조구성에 사용된 유용성을 감안하여 패를 평가한다.

■ 중심어 : | 마작 | 보드게임 | 알고리즘 | 확률 |

Abstract

We introduce a tile grading algorithm for orthodoxy mahjong game. Mahjong is one of most popular board game in the world. In Mahjong, the most important part of skill is the worthless tile selecting method from one's own hands and discard it. To get the value of the tiles in one's hands, at first, we generate every possible cho by using each overlapped tiles. Given these cho, we make the configurable candidate from one's own tiles. At last, we select the best candidate from the candidates and evaluate the tiles depend on usefulness of configuring the cho.

■ keyword : | Mahjong | Board-game | Algorithm | Probability |

I. 서론

중국에서 시작된 마작은 단일 게임으로는 세계 최대의 동호인을 확보하고 있는 게임이다. 우리나라를 비롯하여 일본 및 아시아 지역의 대부분이 마작 게임을 즐기고 있으며 미 대륙과 유럽으로도 급속히 번져나가고 있는 추세이다. 이렇게 마작 게임이 널리 퍼진 것은 컴

퓨터를 통해 게임을 즐길 수 있도록 여러 가지 프로그램을 개발하고 이를 인터넷을 통해 사용자층을 확보해 온 영향이 크다. 현재 우리나라만 해도 인터넷을 통해 마작을 해볼 수 있는 곳이 여러 곳 있으며 많은 회원과 동호회를 가지고 있다[1].

현재 컴퓨터를 이용한 마작 게임은 PDA나 휴대폰 등을 매체로 하여 일반인들이 손쉽게 접근할 수 있도록

다가서고 있으며, 인터넷을 통해 다른 사용자와 대국을 해볼 수 있도록 지원하는 사이트[8]도 많이 찾아볼 수 있다. 일본에서는 사람과 사람간의 컴퓨터 게임에서는 컴퓨터가 단지 운영자의 역할만을 해주도록 프로그램 되어 있으면 충분하나, 컴퓨터가 플레이어의 역할을 담당해야 할 때는 마작 플레이어가 판단해야할 중요한 문제들을 컴퓨터가 대신할 수 있도록 프로그램 되어야만 한다. 정통 마작 게임과 같이 패를 얻은 후 자신에게 필요 없는 패를 버리는 식의 게임에서는 어떤 패를 선택해서 버리는 것이 가장 자신에게 유리한가를 판단하는 것이 승리의 관건이 된다.

본 논문에서는 정통 마작에서 자신이 현재 가진 패들 가운데 어떤 패를 버리는 것이 유리한가를 분석하기 위해 가진 패의 중요도 계산하는 것을 주제로 삼는다. 실전 마작에서는 계산 되어진 중요도를 참고하여 가장 중요도가 적은 패를 찾아내기 위하여 제안된 패 선택 알고리즘이 사용될 수 있을 것이다.

II. 정통 마작 게임 룰

본 장에서는 마작 게임 진행 방법을 간략히 소개 한다. 마작게임의 룰은 나라마다 많은 차이를 보이고 있으며 같은 나라에서도 수많은 룰이 통용되고 있으므로 여기에서는 중국 정통 마작[2]을 기본으로 하여 일반적으로 사용되며 공통적인 룰을 소개하도록 하며, 논문의 주제와 관련이 없는 룰은 설명을 생략할 것이므로 룰에 관한 상세한 설명은 다른 서적을 참고하도록 한다. 논문에서는 플레이어가 가진 패들 중 버릴 패를 선택하는 알고리즘을 다루고 있으며 이 알고리즘은 거의 대부분의 마작 게임에서 공통적으로 사용될 수 있는 부분에 해당한다[1].

1. 마작 게임에 사용되는 패의 종류

마작패는 6종류로 분류되고, 42가지의 그림을 가지며 총 144개의 패로 구성된다.

숫자값을 가지는 종류로 숫자패는 총 108장이며 만자패(万字牌), 병자패(餅字牌), 조자패(條字牌)로 나뉜다.

각 종류마다 1부터 9까지의 숫자를 나타내고 있으며 각 종류별 모든 숫자는 4개의 같은 모양의 패로 구성되어 있다.

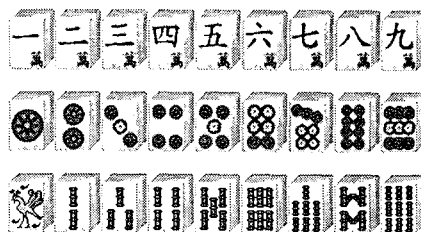


그림 1. 3종류의 숫자패

문자패(文字牌)는 전패(箭牌)와 풍패(風牌)로 나뉘며 총 28장이다. 전패는 중(中), 발(發), 백(白) 각각 4장씩 합 12장이며, 풍패는 동(東), 남(南), 서(西), 북(北) 각 4장씩 합 16장으로 구성된다.



그림 2. 일곱가지 글자패

화패(花牌)는 춘(春), 하(夏), 추(秋), 동(冬), 매(梅), 난(欄), 국(菊), 죽(竹) 각 1장씩 합 8장으로 구성된다. 화패는 게임에서 보너스 점수를 얻게 하는 역할로만 사용되며 본 논문에서는 언급하지 않는다.



그림 3. 8가지 화패

2. 마작 게임 진행

마작 게임의 대국 방식은 4인의 플레이어가 참여 하는 것을 기본으로 한다. 각 플레이어의 동, 남, 서, 북 네 위치에 자리하고 최초로 동이 선이 되어 게임을 시작한다. 게임이 시작되면 패가 섞여 한쪽에 쌓는다. 이후 선은 쌓인 패에서 모든 플레이어에게 순서에 따라 13개씩의 패를 배분해 준다. 초기 배분이 끝나면 선부터 패를

하나 받게 되며, 받은 패와 본인이 가지고 있는 패 중 필요 없는 패를 선택하여 버림으로 순서가 넘겨진다.

단, '핑(碰)'을 선언한 경우에는 선언한 사람이 순서를 빼앗아 올 수 있다. 이는 다른 사람이 낸 패를 자신이 규칙에 맞게 조합할 수 있을 경우, 그 사람에 의해 버려진 패를 가져오며 순서를 획득하는 방법이다. 이 경우도 새로 얻게 된 한 패를 포함한 14개의 패 중 불필요하다고 여겨지는 한 패를 버려야 한다. '핑'은 다른 사람이 버린 패와 같은 패를 두개 이상 가지고 있는 경우에만 선언할 수 있다.

또한, 순서상 직전에 해당 하는 사람이 버린 패로 연속된 세 개의 수패를 형성할 수 있을 경우 '츠(吃)'를 선언하여 새 패를 받는 대신 앞 사람이 버린 패를 취할 수 있다.

3. 마작 게임 종료

새로 패를 받거나, '핑' 또는 '츠'를 선언하게 되면 일시적으로 14개의 패를 손에 갖게 된다. 또한 상황에 관계없이 누군가가 버린 패를 내가 '쏘아(榮和)'를 선언하며 가져온다고 가정 했을 때도 14개의 패를 갖게 된다. 이때, 14개의 패가 특정 조건을 만족한다면 게임의 종료를 선언할 수 있고 게임의 종료를 선언한 사람이 게임에서 승리하게 된다. '쏘아'는 그 패로 인해 게임이 종료할 수 있을 때, 선언되어야 한다.

14개의 패가 종료 조건을 만족하기 위해서는 4개의 조와 1개의 머리(頭)를 가져야 한다. 하나의 조는 3개의 패로 구성되며, 3개의 패가 같은 모양일 경우, 이를 '쑤즈(順子)'라 하고, 3개의 수패가 연속된 값으로 구성될 경우, 이를 '커즈(刻子)'라고 한다. [그림 4]와 같이 쑤즈와 커즈는 하나의 조로 사용될 수 있다.



그림 4. 쑤즈와 커즈로 조를 구성한 예

또, [그림 5]와 같이 그림이 같은 모양의 패 두개를 머리로 사용할 수 있다.



그림 5. 9병으로 머리를 구성한 예

이와 같이 4개의 조와 하나의 머리를 갖게 되면 게임에서 승리할 수 있다. [그림 6]은 종료 조건을 만족하는 하나의 완성된 패 조합을 보이고 있다. '9만' 커즈, '백' 커즈, '1병' 쑤즈, '5조' 쑤즈 이렇게 4개의 조 그리고, '동' 머리로 구성된 완성패이다.

승자가 결정된 후에는 승자가 얻은 패의 조합을 고려하여 점수를 내게 되고 이 점수에 따라 패자는 적절한 대가를 지불하게 된다. 그러나 본 논문에서는 승리를 위한 패 선택에만 관심을 가지고 있으며 승리한 패의 조합이 어떤 점수를 가질 수 있는지는 고려대상이 아니므로 이에 대해서는 논의하지 않기로 한다.

III. 패 선택 알고리즘

보드게임에서의 인공지능은 확률적인 방법, 유전 알고리즘에 의한 방법, 신경망 알고리즘, 정련(Simulated Annealing)에 의한 개선 방법 등 다양한 방법으로서 시도가 가능하며 본 논문에서는 가장 일반적으로 사용되어지는 확률에 근거한 방법을 논한다[3].

2장에서 설명된 바와 같이 자신이 이미 가진 13장의 패에 남아 버린 패 혹은 새로 받은 패가 승리의 조건에 부합될 때, 승리를 선언할 수 있게 된다. 그러나 승리의 조건에 맞지 않는다면 자신이 가진 패 중 가장 중요성이 떨어지는 패를 버려야 한다. 이때 자신이 가진 패의 값어치를 분석해야 하는데, 이미 조를 이루고 있는 패는 분석할 필요가 없게 된다. 따라서, 가진 패를 어떤 조로 구성할 것인가를 먼저 선택하고 여기에서 조에 속하지 않은 패를 대상으로 중요성을 분석하여야 한다.

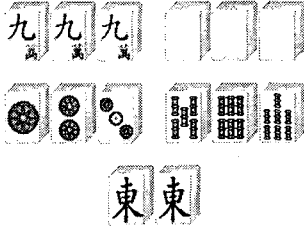


그림 6. 완성된 패 조합의 예

1. 구성 가능한 모든 조의 분석

14개의 패를 대상으로 이 패들이 구성 할 수 있는 조에는 어떠한 것이 있으며 어떤 조를 선택하는 것이 가장 유리한가를 계산하고 가장 유리한 조를 구성하는 방법을 설명한다.

주어진 패는 여러 가지 종류의 조합으로 구성될 수 있으며, 이들 조합 중 가장 유리한 조합을 선택하는 것이 무엇보다 중요하다. 예를 들어, [그림 7]과 같은 경우, 가능한 조의 조합은 '1만' 쏜즈, '2만' 쏜즈, '3만' 쏜즈의 조합을 구성할 수 있으나, 두개의 '2만' 쏜즈로 구성할 시에는 조합을 이루지 못한 3개의 패를 남기게 된다. 또, 최악의 경우, '3만' 커즈로 결정할 경우, 단 하나만의 조밖에 구성할 수 없게 되기도 한다. 이와 같이, 주어진 패를 어떻게 조로 구성하느냐는 승리를 결정지을 수 있을 만큼 중요한 역할을 담당하게 된다.



그림 7. 다양한 조로 구성될 수 있는 패

가장 큰 이득을 갖는 조를 구성하기 위하여 먼저, 가진 패를 중복으로 사용할 수 있다는 가정 하에서 모든 조를 구한다. 이를 구하는 알고리즘은 다음과 같다.

```

for each image number 1 ≤ i ≤ 34
    if |pi| ≥ 3 then Cho = Cho ∪ {Ci}
endfor
for each kind of numeric-tile k, number i
    m = min(|pk,i|, |pk,i+1|, |pk,i+2|)
    Cho = Cho ∪ {Sk,i}, m times
endfor
    
```

알고리즘에서 p_i 는 가지고 있는 패를 나타내는 백 (bag)이다. p_i 와 같이 백이 인덱스를 가진 경우, 이미지 번호 i 를 가지는 패로 구성된 부백(sub-bag)을 의미하며, $p_{k,i}$ 와 같은 인덱스를 가진 경우, 만·병·조를 표현한 k ($1 \leq k \leq 3$)타입의 수패에 숫자값 i 를 가지는 패의 부백을 의미한다. image는 마작에 사용되는 이미지 종류에 대한 일련 번호로 '1만'을 1으로 삼고 마지막 '북자'를 34로 한다. C_i 는 커즈 번호로써 총 이미지 종류 만큼의 커즈가 존재할 수 있다. $S_{k,i}$ 는 쏜즈 번호로 각 수패 종류 k 마다 7 (j)종류의 조가 구성될 수 있으며, 가진 패의 수에 따라 중복도 가능하다. Cho는 조합 가능한 조 번호를 저장하는 백으로 커즈 번호 C_i 와 쏜즈 번호 $S_{k,i}$ 를 저장할 수 있다.

중복 사용가능 조 구성 알고리즘은 먼저, 이미지 종류에 따라 패의 수를 구하여 배열 p 에 저장 하고, 같은 이미지를 3장 이상 가진 경우, 커즈가 가능하므로 조 구성 배열 Cho에 커즈 번호 C_i 를 추가한다. 또 수패의 경우, 3연속된 수패가 존재하는 경우, 쏜즈 번호 $S_{k,i}$ 를 역시 추가 한다. 이때, 같은 쏜즈 번호는 둘이상도 추가될 수 있다. 예를 들어, 1, 2, 3만 패가 각각 3개씩 있는 경우, '1만' 쏜즈가 3개 추가될 수 있다는 것이다.

2. 조합 가능한 조 구성의 후보

3.1절에서 설명한 알고리즘은 하나의 패를 중복 사용하여 구하여 질 수도 있으므로 가능한 조의 배열로부터 패를 중복 사용하지 않고도 구성할 수 있는 조 구성의 후보를 구하여야 한다.

다음은 패를 중복 사용하지 않는 방법으로 조합 가능한 조 구성의 후보를 구하는 알고리즘이다.

```

for each s ∈ Cho
  if Tile(s) ⊆ p then
    Candi = Candi ∪ {s}
  endif
endfor
for each s, t ∈ Candi, s ≠ t
  if s ⊂ t then Candi = Candi - {s}
endifor
for each s ∈ Candi
  if |s| > |BestCandi| or
    |s| = |BestCandi| and
    |Set(p - Tile(s))| < |Set(p - Tile(BestCandi))|
  then BestCandi = s
endifor
    
```

s는 Cho의 부분이며 모든 s에 대해 s에 포함된 모든 조를 구성하기 위해 필요한 패 Tile(s)들을 가지고 있는 상태라면, 이 조들의 백은 하나의 후보가 된다. 이렇게 구하여진 조들의 백은 이 백에 든 조를 다 포함하는 다른 후보가 존재하는 경우, 쓸모없는 것으로 간주되고 후보 집합에서 제거함으로써 연산의 효율을 높이도록 한다. 구하여진 후보들 중 가장 많은 조를 포함한 후보를 최상의 후보로 선택한다. 포함된 조의 수가 같은 경우에는 조를 구성하지 못한 패 중 머리를 구성할 수 있는 패를 많이 가진 후보를 최상의 후보로 선택한다.

3. 남은 패의 중요도 분석

최상의 후보를 선택한 후, 이 후보가 구성한 조에 포함되지 못한 패가 남은 조를 구성하고 하나의 머리를 구성하면 종료조건이 되므로 조구성에 사용되지 못한 패만이 남은 작업의 고려대상이 된다. 조구성에 사용되지 못한 패들의 백을 q라 하였을 때, 남은 패의 중요도를 구하는 알고리즘은 다음과 같다.

for each image number 1 ≤ i ≤ 34, |q_i| ≥ 1

$$vc = \prod_{n=0}^{2-|q_i|} \frac{4 - \text{Known}(i) - n}{4} \cdot R$$

$$vs = \sum_{j=1-2}^i \prod_{n=j}^{i+2} \text{Prob}(n)$$

$$v_i = \max(vc, vs)$$

$$\text{Selection} = \min(\text{Selection}, v_i)$$

endfor

게임 중 타인이 버린 패는 전면이 보이도록 버리게 되므로 어떠한 패가 나와 있으며 어떤 패가 앞으로 나올 수 있는지를 가늠할 수 있다. 예를 들어, 자신이나 타인이 버린 패에 '3만'이 4개 있다면, '3만'은 나올 수 없는 패인 것이다. 이미지 번호가 i인 패에 대해, 내가 가지고 있거나 누군가에 의해 버려진 패의 수를 Known(i)로 정의한다. 또, 배분 가능한 남은 패의 수를 r 다른 게이머의 수중에 있어 내가 받을 수 없는 패의 수는 14×3=42이라 하자. 내가 얻기를 기대하는 패가 남은 패 안에 있을 수도 있고 다른 게이머의 수중에 있을 수도 있다. 또 남은 패 중 1/4정도가 한사람에게 배당될 것이다. 따라서 자신이 기대하는 패를 받을 가능성은 R = r/(r+42)/4가 된다.

먼저, 남은 패들이 어떤 이미지 i에 대해 커즈를 구성할 가능성을 고려해 볼 때, 그 이미지를 전혀 가지고 있지 않다면, 이미지 i를 3개 이상 얻을 확률만큼의 중요도가 있다. 1개를 가지고 있다면, 2개를 얻게 될 확률, 2개를 가지고 있다면 1개를 얻게 될 확률을 그 이미지의 중요도 u라고 볼 수 있다.

다음으로, 남은 패들이 새로 얻게 될 어떤 이미지의 도움으로 쏜즈를 구성할 가능성을 고려해 본다. 우선 어떤 이미지 i를 얻을 수 있는 가능성 Prob(i)을 다음과 같이 정의한다.

$$\text{Prob}(i) = \begin{cases} \frac{4 - \text{Known}(i)}{4} \cdot R & \text{for } |q_i| = 0, \\ 1 & \text{for } |q_i| > 0 \end{cases} \quad (1)$$

이 수식에서 자신이 가진 이미지에 대한 가능성은 1로 설정해 또다시 자신이 그 이미지를 갖게 될 가능성이 무의미하도록 하였다. 쑤즈를 구성하기 위해서는 3 연속된 수패를 가지고 있어야만 한다. 따라서 수중에 있는 어떤 이미지 i 가 쑤즈로 조를 구성하는 것은 자신이 $i-2$ 와 $i-1$ 을 또는, $i-1$ 와 $i+1$ 을 또는, $i+1$ 와 $i+2$ 를 가지는 3가지 경우로 볼 수 있다. 이 3가지의 합이 이미지 i 가 쑤즈를 구성할 가능성 us 가 된다. 첫 번째의 경우, 이 가능성은

$$us = \text{Prob}(i-2) \cdot \text{Prob}(i-1) \quad (2)$$

로 계산될 수 있다.

끝으로, 한 이미지가 커즈나 쑤즈를 동시에 만족할 수 없으므로 커즈를 이룰 가능성과 쑤즈를 이룰 가능성 중 하나만을 택하여야 한다. 여기에서는 둘 중 가능성이 높은 쪽을 택하도록 하였다. 이와 같이 모든 가진 이미지에 대한 중요도를 구한 후, 중요도가 가장 낮은 이미지를 선택하여 버릴 이미지로 결정한다.

IV. 실험 및 고찰

알고리즘을 검증하기 위하여 임의의 상황을 가정한 상태에서 각 패가 어떻게 평가되는지 분석하고, 이를 실제 게임에 적용하여 성능을 실험한다.

1. 패 평가 분석

제안된 알고리즘을 실험하기 위하여 [그림 8]과 같은 패를 가진 상황을 가정한다.

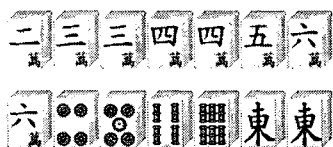


그림 8. 실험을 위한 상황 제시

이 경우, 제안된 알고리즘은 우선 구성 가능한 모드를 분석하여 [그림 9]에서 보인 '2만' 쑤즈, '3만' 쑤즈, '4만' 쑤즈를 찾아낸다.



그림 9. 가능한 3개 조

다음으로, 후보 구성 알고리즘은 조합 가능한 후보들을 만든다. 여기에서는 [그림 10]과 같이 두 가지 후보가 나올 수 있다.

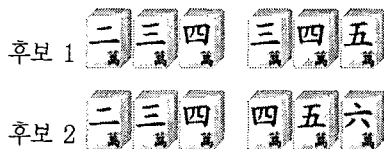


그림 10. 구성된 후보

이중 후보 1을 선택하면 머리로 이용하거나 커즈를 기대해 볼 수 있는 '6만'이 남으므로 후보 1을 선택한다.

후보 1을 택할 경우, 조 구성에 사용되지 못한 패는 [그림 11]과 같다.



그림 11. 조 구성에 사용되지 못한 패

남은 패의 중요도를 계산하여 버릴 패를 선택하는 마지막 단계에서는 자신이 가진 패와 더불어 '6만'과 '5병'이 누군가에 의해 버려져 있는 상황이라 가정해 본다. '6만'의 경우, 커즈를 구성할 가능성을 계산할 때, 이미 세 개의 '6만'이 알려진 상태이므로 1개의 '6만'을 기대하는 상황이 된다. 따라서 0.163의 값을 갖는다. 또한 '6만'이 쑤즈를 구성할 가능성은 '4만'과 '5만'에 대한 기대값 0.1612, '5만'과 '7만'에 대한 기대값 0.322, '7만'과 '8만'에 대한 기대값 0.429로써 이 값을 모두 합하면 2.125가 된다. 이와 같은 방법으로 남은 모든 패의 중요도를 계산하면 [표 1]과 같다.

표 1. 남은 패의 중요도 분석

이미지						
커즈값	0.163	0.163	0.053	0.161	0.161	0.327
쑈즈값	0.913	1.741	1.741	1.515	1.515	0
중요도	1.077	1.902	1.795	1.676	1.676	0.327

계산된 결과를 볼 때, 조를 구성하지 못한 패 중 중요도가 가장 낮아 버려지는 것이 바람직한 패는 동패인 것으로 분석 되었다. 일반적으로 문자패는 쑈즈를 구성하지 못하므로 특별한 전략을 갖지 않는 한 게임의 초반에 버려지게 된다. 분석 결과에서 조를 구성한 패 다음으로 가장 중요한 패로는 '4병'을 들 수 있다. '4병'은 버려진 패도 없어 커즈를 구성하기도 쉽고, 숫자 서열상 주위의 패들도 많이 남아 있는 편이라 쑈즈값도 높은 결과를 얻었다.

2. 알고리즘의 적용

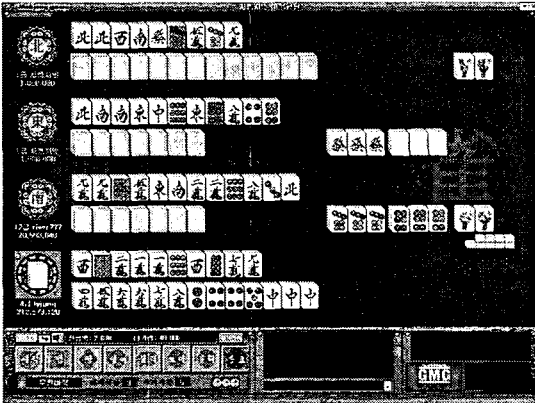


그림 12. 상용 사이트에서의 성능시험

알고리즘의 성능을 검증하기 위해 상용 게임 사이트에서 테스트하였다. 실험 대상으로 삼은 게임 사이트는 http://www.gmg.net/game/mahjong_new.asp이다.

[그림 12]는 게임 사이트에서 제공된 마작 프로그램에서 테스트 진행 중인 게임 화면을 보인 것이다. '북', '동', '남', '서' 순으로 네 명의 게이머가 참여해 있고 맨 아래 '서'자리에서 'tyjung'이라는 아이디로 알고리즘이

실험을 위해 게임에 참가하고 있다. '북'과 '동'에 자리한 두 게이머 '지엠지넷'은 게임 운영 사이트에서 제공된 인공지능이며 '남'에 자리한 'river777'은 게임에 참여한 또 다른 실제 인물의 아이디어이다. 각 자리는 버린 패 한 줄과 손에 쥔 패 한 줄로 구성되어 있다.

알고리즘을 상용 게임 프로그램에 적용하기 위하여 [그림 13]에 보인 중재 프로그램 'MjPlayer'를 작성하였다. 이 프로그램은 실제 게임 프로그램의 화면[그림 12]을 이미지 캡처하여 각 게이머의 패를 인식한 뒤, 이를 분석하기 좋은 모양으로 배열하여 보여준다. 또, 사람을 대신하여 게임을 진행할 수 있도록 제안된 알고리즘을 내장하고 있으며, 마우스와 키보드를 스스로 제어하여 상용 게임 프로그램에 적용하므로 'MjPlayer' 사용자는 게임에 접속한 후 손을 쓰지 않은 상태에서 게임의 진행과정을 관전할 수 있게 되어 있다.

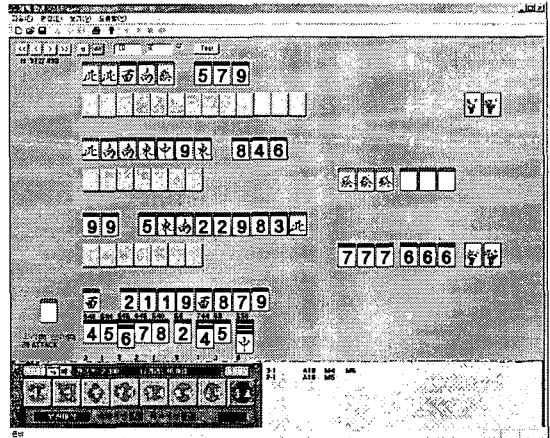


그림 13. 패 분석 및 실제 게임에 적용하기 위한 'MjPlayer'

'tyjung'이 가진 패에 대해서는 평가된 값을 각 패 위에 정수 형태로 표시하였고 패 아래에는 그 때까지 보여지지 않은 같은 모양의 패 수(4-*Known(img)*)를 표시하였다. 'MyPlayer'의 오른쪽 하단부는 현재 구성 가능한 조들의 후보를 분석한 것이다. 첫 번째 후보는 '중' 커즈, '4만' 쑈즈, '6만' 쑈즈, '4병' 머리로 3개의 조와 하나의 머리로 구성되어 있다. 두 번째 후보는 '중' 커즈, '5만' 쑈즈, '4병' 머리며 두개의 조와 하나의 머리로 구성되어 있다.

'MjPlayer'를 이용하여 대상 사이트에서 제공된 인공지능 셋과 무승부를 제외한 500회의 게임을 수행한 결과 144회의 승리로 28.8%의 승률을 거두었고 4명이 참여하는 게임에서의 기본 승률을 25%로 볼 때, 비교적 안정된 성능을 보였다고 판단된다.

일반인이 참여한 게임에 있어서 'tyjung'이 인공지능이라는 것을 알아채는 게이머는 없었으며 여러 달에 걸쳐 17,837회의 게임에 4,704회 승리하여 26.3%의 승률을 보였으나 화면 캡처 상태 불량, 프로그램의 멈춤 등과 같은 비정상적 상태에서의 게임이 일부 포함되어 있어 정확한 통계라 보기는 어렵다. 이러한 상황을 고려할 때, 정상적인 상태에서는 더 높은 승률을 기대해 볼 수 있다.

V. 결론

본 논문을 통하여 현재 세계적으로 수많은 사용자 층을 확보하고 있는 정통 마작 게임에서의 패에 대한 값어치 평가 알고리즘을 소개하였다. 이 알고리즘은 사용자와 컴퓨터가 마작게임을 할 수 있는 게임 프로그램을 설계할 때, 가장 핵심이 되는 부분이라 할 수 있다[7].

소개된 알고리즘에서는 평과 츠를 선언할 경우를 고려하고 있지 않다. 평은 순서에 관계없이 다른 사람이 버린 패를 가져와 커즈를 만들 수 있게 하고, 츠는 바로 앞 사람이 버린 패를 가져와 쏘즈를 할 수 있는 방법이다. 따라서 이 두 가지를 고려하여 알고리즘을 수정한다면 커즈에 대한 중요도 값이 더 올라가게 될 것이고 좀 더 정확한 패의 선택이 이루어질 수 있으리라 본다.

또, 승자의 점수에 따라 보상이 크게 달라지므로 후보 분석에 있어서 점수를 고려하는 방안 또한 깊이 고려되어야 할 것이다. 마작에서 점수를 계산하는 방법이 십에서 백여 가지에 이르기 까지 매우 다양한 조합이 사용되어지므로 이를 일일이 코드로 구현한다는 것은 무리가 따르며 적절한 스크립트로 구성된 후 이를 통해 점수를 계산하도록 해야 할 것이며[5] 이에 대한 연구도 진행 중에 있다.

참고 문헌

- [1] 임상혁, 김상진, *각국(중국, 한국, 일본) 마작 게임 설명서*, Mobizen Park, 2004.
- [2] 國家体育总局審定, *中國麻將京妓規則*, 人民體育出版社, 1998(7).
- [3] M. Buckland, *AI Techniques for Game Programming*, Thomson, 2002.
- [4] S. Baase, *Computer Algorithms*, Addison Wesley, January, 1991.
- [5] B. Mark and M. Darrach, *How Not To Implement a Basic Scripting Language*, AI Game Programming Wisdom, Charles River Media, Inc., 2002.
- [6] J. E. Laird and Michael van Lent, "Human-Level AI's Killer Application: Interactive Computer Games" *AI Magazine* Vol.22, No.2, pp.15-26, 2001.
- [7] 위자드소프트, *AI 에이전트 NPC 기반의 게임 스타일 이식 엔진의 개발*, 정보통신연구진흥원, 2005.
- [8] <http://www.gmg.net/game/etc/anpro.asp>

저자 소개

김 상 진(Sang-Jin Kim)

정회원



- 1994년 2월 : 계명대학교 컴퓨터 공학과(공학사)
- 1996년 2월 : 경북대학교 컴퓨터 공학과(공학석사)
- 2000년 8월 : 경북대학교 컴퓨터 공학과(공학박사)

• 1999년 9월~현재 : 경운대학교 컴퓨터공학과 교수
<관심분야> : 게임알고리즘, 3D그래픽스