韓國數學教育學會
<全國數學敎育硏究大會 프로시딩>
제 36회. 2006. 5. 49-61

Korea Soc. Math. Ed.
Proceedings of National Meeting of Math. Ed.
The 36th. May. 2006. 49-61

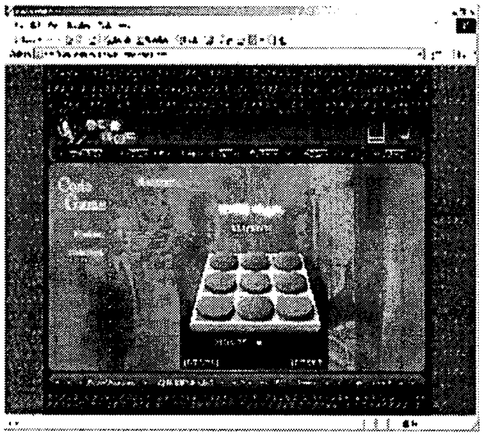
바둑판을 이용한 시그마게임과 선형대수학[1)]

이 상 구 (성균관대학교)

영화 '**뷰티플 마인드(A Beautiful Mind)**'의 홈페이지에서 소개된 3 by 3 바둑판을 이용한 시그마게임 '흑백(Blackout 또는 black and white 또는 MERLIN)'의 수학적 모델링을 하고, 그 모델을 선형대수학의 지식만을 이용하여 완전히 해결한 후, 그 과정에서 개발된 알고리즘을 이용하여 프로그램을 짜서 실제 19 by 19 크기의 바둑판에서의 시그마게임을 즐긴다. **이 게임의 최적해를 찾는 선형대수학의 방법**을 대학에서 배우는 내용과 연관하여 소개하며 학부 과정의 수학 지식이 주위의 문제해결에 얼마만큼 중요한 열쇠가 되는지를 소개한다. 이 과정은 선형대수학 입문 과정의 지식으로 설명이 가능하여 학부강의실에서 벡터공간과 기저에 관한 이론의 예로 이용할 수 있다는 수학교육적 의미가 있다.
이 과정을 통하여 실제 생활에서 찾을 수 있는 많은 수학적 모델에 대한 수학적 모델링과 그 모델의 수학적 처리 과정을 통하여 수학의 응용 및 활용에 대한 의미를 찾을 수 있음을 확인한다.
http://matrix.skku.ac.kr/BlackWhite2/BlackWhite.html

I. 서 론

바둑과 수학에 관하여는 적지 않은 연구논문과 프로그램이 개발되어 왔다. 본 원고에서는 '흑백게임'을 수학적으로 분석하여, 이 게임을 이기는 최적의 전략을 소개하고, 이 알고리즘을 이용하여 제작한 프로그램을 제공한다. 이 방법은 새로운 해법이면서도 선형대수학 입문 과정의 지식만으로 흥미를 끄는 실제 문제에 대한 구체적인 설명을 준다.

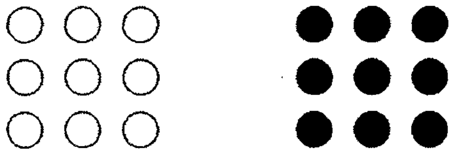


<그림 1> 흑백(Blackout)게임

실제로 이 문제에 대한 선형대수학적 해법이 가능하지 않냐고 질문을 던지고, 아이디어를 찾은 사람은 박종빈이라는 선형대수학을 듣던 학부 학생이었다. 그 연구 과정을 지도하며 완성한 내용을 소개한다. 이 결과는 학생의 연구 참여 경험을 주었다는 의미에서 미국과학재단(NSF)의 후원으로 약 10년 전부터 수학 분야만도 매년 여름 40-60개 대학에서 진행되는 학부생을 위한 연구경험프로그램[2)] 과 같은 교육적 효과를 보여주었다.

1) This work is supported by Com ^{2}MaC－KOSEF.
2) Research Experience for Undergraduates Programs http://www.ams.org/employment/reu.html

3×3 바둑판에 바둑알 9개가 있다. 게임의 목적은 바둑알을 선별적으로 클릭하는 과정을 거쳐 바둑알 9개 모두를 아래의 2가지 경우와 같이 같은 색으로 만드는 것이다. 이렇게 하기 위하여 클릭할 바둑알을 선택할 수 있는데, 이 과정에서 조건은 한 바둑알을 클릭하면 주변 이웃하는 바둑알들이 함께 뒤집어지게(toggle) 된다는 점이다.



<그림 2> 완전히 뒤집어진 상태

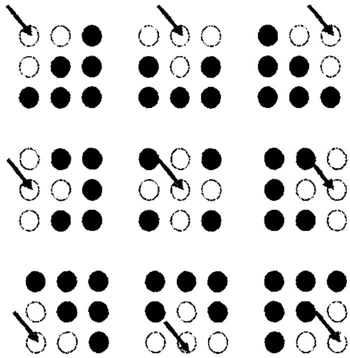
이 문제에서 일부의 바둑알이 뒤집힌 상태로 주어진 초기의 패턴에 따라서 답이 있는 문제와 답이 없는 문제가 존재하는가? 그런 존재성을 수학적으로 증명할 수 있을까? 즉 홈페이지 관리자가 “상대가 이길 수 없는 초기조건 (영원히 이길 수 없는 문제)”을 제시할 수 있을까? 그렇지 않다면 이 문제는 어떠한 초기조건이 주어져도 반드시 답이 존재하는 그런 부류의 문제일까? 만약 그렇다면 “그런 사실에 대한 증명은 가능한가?” 하는 아래 문제들을 생각할 수 있다.

1. 흑백게임은 언제나 해를 갖는가?
2. 해가 존재한다면 몇 개나 존재하나?
3. 존재하는 해는 어떻게 찾는가?
3. 이 과정을 최적의 알고리즘으로 표현할 수 있는가?
4. 알고리즘에 기인한 프로그램을 제작가능한가?

II. 본 론

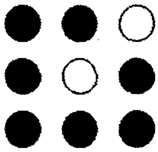
먼저 흑백 게임에서 이기기 위한 나름대로의 전략을 직관적으로 생각해보자. 일단 3×3 바둑판이 가질 수 있는 모든 바둑판 패턴의 개수는 몇 가지일까? 즉, 상계(upper bound)를 찾아보자. 이것은 쉽게 구할 수 있는데 9개의 바둑알이 스스로 2가지 경우를 결정하는 상황이므로 $2\times 2\times \cdots \times 2 = 2^9 = 512$가지의 다른 패턴이 생길 수 있다.

직관적으로 512가지 모든 경우의 수에 대해 클릭 1번으로 게임이 끝날 수 있는 $9\times 2 = 18$가지 패턴이 존재함을 알 수 있다. (아래의 9가지 패턴은 또한 흰색은 검은색으로 검은색은 흰색으로 바꿈에 따라 또 다른 9가지 패턴이 더 발생하게 된다.)



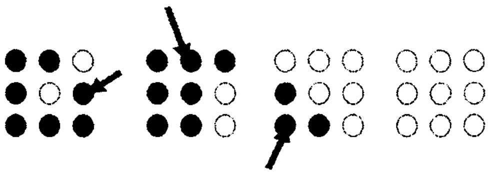
<그림 3>

그 이유를 설명하기 위하여 다음과 같은 한 가지 예를 생각해 볼 수 있다. 아래와 같이 초기 조건이 주어졌다고 가정하자.



<그림 4>

위의 상황은 위에서 제시한 18가지 패턴에 속하지 않는 경우다. 그런데 아래와 같은 순서로 클릭을 수행하면 세 번의 클릭으로 모두 흰색으로 만들 수 있게 된다.

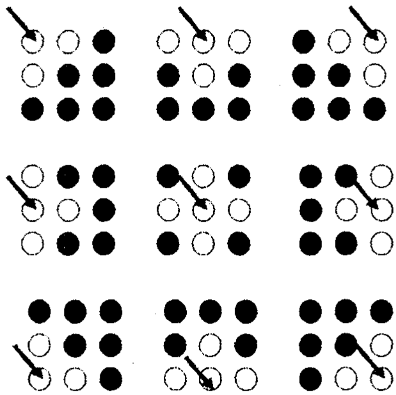


<그림 5>

실제 많은 게임을 수행한 결과 시행착오에 의해 시간이 걸렸을 뿐 단 한 번의 실패도 없었으며 따라서 흑백 (Blackout) 게임에서는 어떠한 초기조건에 의해서도 반드시 답이 존재하는 문제일 거라는 추측이 가능하다. 이 과정에서 게임 승리의 전략으로서 18가지의 기본적인 패턴이 존재한다는 것을 인식하는 것은 증명의 열쇠를 준다. 그리고 이 18가지 패턴이란 바둑알 하나하나를 클릭하는 것과 동일하다는 사실도 중요하다.

이제 이 게임의 해의 존재성에 대한 수학적 증명을 시도해 보자. 위 게임에서 클릭을 한다는 것을 수학적으로 표현하면 어떻게 될까? 우리는 수없이 바둑알들을 클릭 할 수는 있지만 서로 다르게 클

릭 할 수 있는 방법은 오로지 9가지뿐이다. 그 이유는 바둑알이 9개이기 때문이다. 즉 가장 기본적인 패턴은 다음의 9가지 밖에 없다는 것이다. 그리고 9가지 바둑알 중 하나의 바둑알을 클릭하면 주변의 바둑알들이 뒤집어지게(Toggle) 되는데 이것은 수학적으로 어떻게 표현할 수 있을까? 이것은 바로 색을 바꾸어 준다는 의미로 대응하는 "9가지 바둑알들의 집합"을 더하는 것으로 이해 할 수 있다. 즉, 바둑판을 3×3 행렬로 표현하고, 9가지 행렬을 2를 법으로 하는 잉여류의 합 (mod 2) 으로 문제를 해결 할 수 있다는 것이다.



<그림 6>

다시 말하면 흰 색이 1이고 검은색이 0 이라고 가정할 때 위의 9개의 바둑알 클릭이라는 동작은 아래의 9개 **"행렬을 더한다"**의 문제로 바뀌게 된다. 만약 초기 조건이 주어졌다고 가정하면 초기 조건에 해당하는 행렬에 9개의 바둑알 클릭을 수행함에 따라 **9개의 행렬 중에 몇 개를 더하는 것과** 같은 문제로 바뀐다는 것이다.

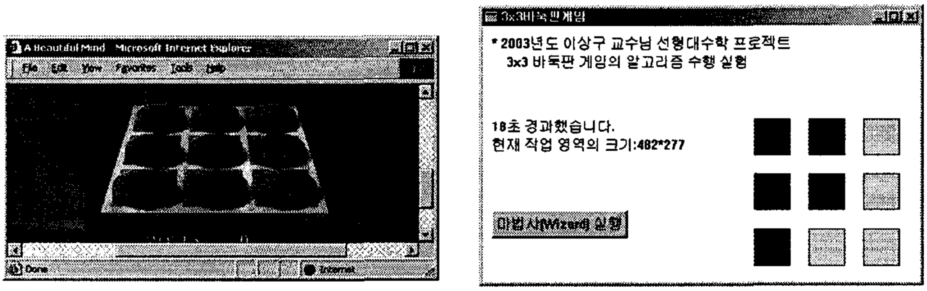
이렇게 더했을 때 과연 $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ 혹은 $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$이라는 결과를 도출할 수 있느냐는 것인데 위의 9개의 서로 다른 바둑알의 클릭은 다음 9개의 행렬 중 하나를 더한다는 것으로 대체가 된다. 이 때, 합은 대응하는 성분끼리의 2를 법으로 하는 잉여류의 합 (mod 2)으로 한다.

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

실제 한 가지 예를 들어 보자. 아래의 경우에는 3×3 바둑판에 흑색 돌(파란) 5개와 흰색 돌(붉은) 4개가 존재하고 있다.



<그림 7>

위의 경우 파란 돌을 0으로 붉은 돌을 1로 가정하면, 초기 조건에 해당하는 행렬은 $B=\begin{bmatrix}0&0&1\\0&0&1\\0&1&1\end{bmatrix}$으로 표현할 수 있고 여기에 마우스를 여러 번 클릭 한다는 의미는 아래의 행렬들을 더하는 것이다.

$$a\begin{bmatrix}1&1&0\\1&0&0\\0&0&0\end{bmatrix}+b\begin{bmatrix}1&1&1\\0&1&0\\0&0&0\end{bmatrix}+c\begin{bmatrix}0&1&1\\0&0&1\\0&0&0\end{bmatrix}+d\begin{bmatrix}1&0&0\\1&1&0\\1&0&0\end{bmatrix}+e\begin{bmatrix}0&1&0\\1&1&1\\0&1&0\end{bmatrix}+f\begin{bmatrix}0&0&1\\0&1&1\\0&0&1\end{bmatrix}$$
$$+g\begin{bmatrix}0&0&0\\1&0&0\\1&1&0\end{bmatrix}+h\begin{bmatrix}0&0&0\\0&1&0\\1&1&1\end{bmatrix}+i\begin{bmatrix}0&0&0\\0&0&1\\0&1&1\end{bmatrix}$$

즉, 위의 행렬을 더한 값이 $\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}$ 혹은 $\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$이 될 수 있느냐의 문제란 것이다. 여기서 $\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}$가 되게 한다는 것은 모든 바둑알을 모두 검은색 돌로 만든다는 것이고 $\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$란 것은 모두 흰색으로 만든다는 것을 의미한다. a 라는 계수가 의미하는 것은 $\begin{bmatrix}1&1&0\\1&0&0\\0&0&0\end{bmatrix}$이라고 하는 행렬을 a번 더하는 것을 의미하는 것으로서 마우스로 첫 번째 (1, 1) 바둑알을 몇 번 클릭 하는지를 말하는 것이다. b,c,d,e,f,g,h,i 역시 마찬가지 방법으로 해석될 수 있다.

문제를 간결하게 생각하기 위해 결과가 모두 검은색이 되게 만드는 게 목적이라고 가정하면 최종값이 $\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}$이면 된다는 것이고 위 게임을 그렇게 해석해 보자. 위 식을 묶어서 다시 쓰면 아래

의 방정식을 만족하는 $a, b, c, d, e, f, g, h, i$를 구하라는 문제가 된다.

$$a\underset{(1,1)}{\begin{bmatrix}1&1&0\\1&0&0\\0&0&0\end{bmatrix}}+b\underset{(1,2)}{\begin{bmatrix}1&1&1\\0&1&0\\0&0&0\end{bmatrix}}+c\underset{(1,3)}{\begin{bmatrix}0&1&1\\0&0&1\\0&0&0\end{bmatrix}}+d\underset{(2,1)}{\begin{bmatrix}1&0&0\\1&1&0\\1&0&0\end{bmatrix}}+e\underset{(2,2)}{\begin{bmatrix}0&1&0\\1&1&1\\0&1&0\end{bmatrix}}+f\underset{(2,3)}{\begin{bmatrix}0&0&1\\0&1&1\\0&0&1\end{bmatrix}}$$

$$+g\underset{(3,1)}{\begin{bmatrix}0&0&0\\1&0&0\\1&1&0\end{bmatrix}}+h\underset{(3,2)}{\begin{bmatrix}0&0&0\\0&1&0\\1&1&1\end{bmatrix}}+i\underset{(3,3)}{\begin{bmatrix}0&0&0\\0&0&1\\0&1&1\end{bmatrix}}+\underset{\text{초기값}}{\begin{bmatrix}0&0&1\\0&0&1\\0&1&1\end{bmatrix}}=\underset{\text{목표값}}{\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}}$$

$$a\underset{(1,1)}{\begin{bmatrix}1&1&0\\1&0&0\\0&0&0\end{bmatrix}}+b\underset{(1,2)}{\begin{bmatrix}1&1&1\\0&1&0\\0&0&0\end{bmatrix}}+c\underset{(1,3)}{\begin{bmatrix}0&1&1\\0&0&1\\0&0&0\end{bmatrix}}+d\underset{(2,1)}{\begin{bmatrix}1&0&0\\1&1&0\\1&0&0\end{bmatrix}}+e\underset{(2,2)}{\begin{bmatrix}0&1&0\\1&1&1\\0&1&0\end{bmatrix}}+f\underset{(2,3)}{\begin{bmatrix}0&0&1\\0&1&1\\0&0&1\end{bmatrix}}$$

$$+g\underset{(3,1)}{\begin{bmatrix}0&0&0\\1&0&0\\1&1&0\end{bmatrix}}+h\underset{(3,2)}{\begin{bmatrix}0&0&0\\0&1&0\\1&1&1\end{bmatrix}}+i\underset{(3,3)}{\begin{bmatrix}0&0&0\\0&0&1\\0&1&1\end{bmatrix}}=\underset{\text{목표값}}{\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}}-\underset{\text{초기값}}{\begin{bmatrix}0&0&1\\0&0&1\\0&1&1\end{bmatrix}}=-B$$

위의 연립방정식을 푸는 과정에서 주어진 9개의 3×3 행렬은 9×1 벡터들로 생각할 수 있으므로 실제로 $\mathbf{x} = [a\ b\ c\ d\ e\ f\ g\ h\ i]^T$에 관하여 풀어보면 9개의 방정식이 생성되고 이렇게 생긴 연립 방정식을 다시 행렬로 표현하면 아래와 같은 9×9 대칭행렬 A를 포함하는 연립일차방정식이 된다. 그림 7의 초기문제에 해당하는 행렬 B와 A를 이용하여 실제 $\mathbf{x}$를 구해보자.

$$A\,\mathbf{x} = -B \Rightarrow A\begin{bmatrix}a\\b\\c\\d\\e\\f\\g\\h\\i\end{bmatrix} = -\begin{bmatrix}0\\0\\1\\0\\0\\1\\0\\1\\1\end{bmatrix} \quad \text{이므로,} \quad A=\begin{bmatrix}1&1&0&1&0&0&0&0&0\\1&1&1&0&1&0&0&0&0\\0&1&1&0&0&1&0&0&0\\1&0&0&1&1&0&1&0&0\\0&1&0&1&1&1&0&1&0\\0&0&1&0&1&1&0&0&1\\0&0&0&1&0&0&1&1&0\\0&0&0&0&1&0&1&1&1\\0&0&0&0&0&1&0&1&1\end{bmatrix}.$$

와 같은 값을 갖게 된다. 매쓰메티카, 매트랩 또는 저자가 개발한 자바행렬계산기[3]등을 이용하여 실제로 위의 계산을 수행해 보면

3) http://matrix.skku.ac.kr/newMatrixCal/Test/html

$$\mathbf{x} = \begin{bmatrix} \frac{1}{7} & -\frac{4}{7} & \frac{1}{7} & \frac{3}{7} & \frac{2}{7} & -\frac{4}{7} & -\frac{6}{7} & \frac{3}{7} & -\frac{6}{7} \end{bmatrix}^T$$

가 되고 $\mathbf{x}$ 의 각 성분을 정수로 만들기 위해 $\mathbf{x}$ 에 7을 곱해주면

$7\,\mathbf{x} = [1\ -4\ 1\ 3\ 2\ -4\ -6\ 3\ -6]^T$

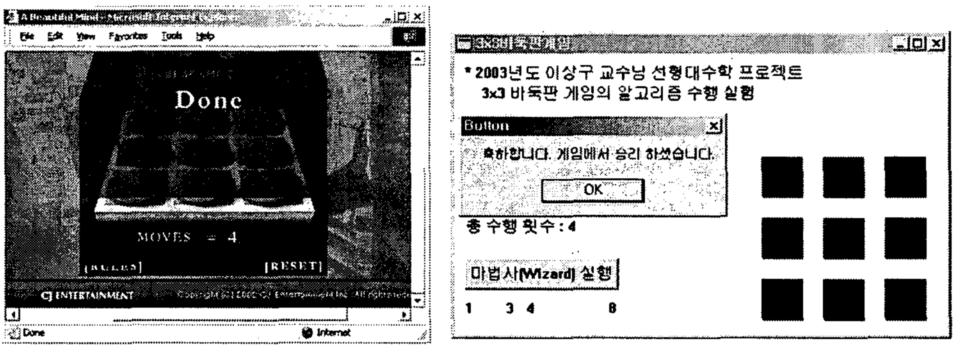
이 되며, 이것의 각 성분들은 2를 법으로 하는 잉여류의 합에 의해

$7\,\mathbf{x} \equiv [1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0]^T$ 이 된다.

위의 벡터를 다시 3×3 행렬로 표현하면 아래와 같이 쓸 수 있는데 우리가 알아 낸 것은 바로 이 행렬이 게임 승리의 전략에 대한 답을 준다는 것이다.

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

실제 1이라고 쓴 (1, 1) (1, 3), (2, 1), (3, 2) 성분의 바둑알을 순서대로 클릭 해본다. 이 성분을 9×1 벡터 성분의 위치와 대응시켜 생각하면 1번, 3번, 4번, 8번 위치를 클릭하면 분명히 4번 만에 게임은 끝이 날 것이다. 사실 클릭하는 순서와는 관계가 없이 게임은 끝난다. 단지 이 4개의 위치를 한번 씩 클릭 하는 것 만 중요한 것이다. 아래의 그림 8은 실제로 주어진 바둑판에서의 지정된 4개 바둑알을 클릭하자 바로 4번 만에 게임이 끝났음을 보여주고 있다. 그림 8에서 "총 수행 횟수 4"는 클릭(MOVE) 4번에 게임이 끝났음을 의미한다. "마법사(Wizard) 실행"이라는 명령을 그림 8에서 볼 수 있듯이 그 아래 숫자 "1 3 4 8"은 저자가 개발한 소프트웨어에서 최적의 해를 확인하기 위해 제공한 기능으로 마법사를 클릭하여 얻은 최적 해, 즉 클릭해야 할 돌의 위치를 보여준 것이다.



<그림 8>

어떻게 이런 일이 생길 수 있는가? 위의 결과는 3×3 흑백 게임에 등장하는 9개의 행렬이 일차결합으로 어떤 임의의 3×3 행렬도 생성하고 각각은 일차독립이라는 사실로부터 증명이 가능하다. 우선 9개의 행렬

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

은 임의의 3×3행렬의 기저(Basis)라는 것이 다음과 같이 쉽게 증명된다.

위의 각 행렬의 모든 성분들을 하나의 행으로 만들어 새로운 행렬 A를 만들고 이 A 행렬의 기약행사다리꼴(RREF)과 계수(rank)를 직접 또는 자체 개발한 웹상의 자바 프로그램[4]을 이용하여 구해본 결과 $\mathrm{RREF}=I$, $\mathrm{rank}A=9$로서 행렬

$$A=\begin{bmatrix}1&1&0&1&0&0&0&0&0\\1&1&1&0&1&0&0&0&0\\0&1&1&0&0&1&0&0&0\\1&0&0&1&1&0&1&0&0\\0&1&0&1&1&1&0&1&0\\0&0&1&0&1&1&0&0&1\\0&0&0&1&0&0&1&1&0\\0&0&0&0&1&0&1&1&1\\0&0&0&0&0&1&0&1&1\end{bmatrix}$$

의 9개의 열은 모두 일차독립임을 알 수 있다[5]. 즉, 위의 9개의 3×3행렬이 각각 1차 독립임이 확인 된다. 따라서 어떤 3×3 행렬도 이 9개의 행렬에 의해 생성(span)된다는 것도 역시 쉽게 유도된다. 이것만으로도 처음에 우리가 가지고 있었던 "어떤 초기조건에 의해서도 반드시 해결되는 문제인가?" 라는 의문에 대한 답은 얻어진다. 즉, 해의 존재성이 증명된 것이다.

실제 게임을 수행할 때는 7을 곱해주고 2를 법으로하는 잉여류의 합 연산을 수행하는데 그 이유는 다음과 같다. 애초 이 게임을 수식화 할 때 우리는 이 게임이 주어진 임의의 3×3 행렬에 대해, 모든 돌을 검은색으로 만드는 최소 클릭의 수와 클릭할 돌 즉, 최적해를 구하는 과정이라고 가정했었다. 하지만 실제로 바둑판 게임의 경우 그것의 해는 반드시 0 아니면 1 이 될 수 밖에 없는 게임으로서 초기에 이 조건을 빼고 계산했기 때문에 나중에 다시 정수로 만들어 주고 2를 법으로 하는 잉여류의 합 연산으로 계산한 것뿐이다. 이 알고리즘은 위수가 2인 유한체 즉, GF(2)에서 설명하는 것이 가능하지만 본 논문의 독자가 추상대수학을 배우지 않은 대학 1, 2 학년 학생들을 포함하여 선형대수학 수업 중에 예로 이용할 수 있도록 본 증명과정에서 선형대수학 입문의 지식만을 이용하고자 한다.

위의 내용을 종합하여 증명과정을 서술하면 다음과 같다. 마우스로 9개의 바둑알 중 한 개를 클릭한다는 것은 앞의 9개 행렬 중 한 개를 더한다는 것을 의미하고 이것은 이 게임을 해석하는데 매우 중요한 부분이다. 그리고 이것은 또 다른 흥미 있는 결론에 이르게 하는데 마우스의 클릭순서가 게임에서 이기는데 중요한 요소가 아니라는 것이다. 즉 "어떤 바둑알을 몇 번 클릭하느냐?"가 승패의 직접적인 영향을 주는 요소이고 문제가 해결가능한지 혹은 그렇지 않은지를 판가름하는 열쇠가 되는

4) http://matrix.skku.ac.kr/newMatrixCal/Test.html
5) 이상구 (2005), 현대선형대수학, 서울: 경문사.

것이지 어떤 바둑알을 클릭한 후 어떤 바둑알을 그 다음에 클릭해야 하는 것은 아니라는 것이다. 즉, "9개 기본 행렬의 일차결합"일 뿐이다.

앞서 서술한 내용을 정리하면 마우스를 클릭 하는 동작은 아래와 같은 수식으로 표현이 가능하다.

$$\underset{\text{초기값}}{\begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix}} + a\underset{(1,1)}{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}} + b\underset{(1,2)}{\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}} + c\underset{(1,3)}{\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} + d\underset{(2,1)}{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}} + e\underset{(2,2)}{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}}$$

$$+ f\underset{(2,3)}{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}} + g\underset{(3,1)}{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}} + h\underset{(3,2)}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}} + i\underset{(3,3)}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}} = \underset{\text{목표값 1}}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}} \text{혹은} \underset{\text{목표값 2}}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}$$

즉, 어떤 $a, b, c, d, e, f, g, h, i$ 에 대하여도 반드시

$$\underset{\text{목표값 1}}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}} - \underset{\text{초기값}}{\begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix}} \text{ 혹은 } \underset{\text{목표값 2}}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}} - \underset{\text{초기값}}{\begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix}}$$

가 mod 2에서 영 행렬이 되는 행렬이 언제나 존재하느냐 하는 문제로 귀결된다.

이것은 다시 아래 9개 행렬이 임의의 3×3행렬의 기저인가? 의 문제로 요약 할 수 있다. 그 이유는 기저의 필요충분조건이 "선형독립"과 "임의의 벡터공간의 생성(span)"이기 때문이다. 이때 "해의 존재성 문제"를 선형대수학적 개념으로 표현하면 아래 9개 행렬이 임의의 3×3행렬을 생성(span)하는가를 묻는 문제가 되는데, 따라서 이 행렬 벡터들의 집합이 기저임이 증명되기만 한다면 "임의의 벡터공간의 생성(span)유무 문제"도 자연스럽게 해결되는 것이다. 즉 다음의 9개의 행렬

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

이 임의의 3×3 행렬의 기저라고 한다면 우리는 위의 9개의 행렬만 가지고서 모든 3×3 행렬을 만들 수 있게 된다. 결국 어떠한 초기조건이 주어지더라도 우리는 해결 가능한 해를 얻게 되는 것이다.

해의 존재성에 대한 보다 수학적인 증명 과정은 다음과 같다. 연립방정식 $A\mathbf{x} = \mathbf{b}$가 해를 가질 필요충분조건은 $\text{rank}(A) = \text{rank}[A \mid \mathbf{b}]$이다[6]. 그런데 위의 행렬들의 1차결합인 행렬 A가 영행렬이면 바둑알이 모두 뒤집혀진 것 이므로 게임은 끝난다. 즉, 해의 존재성이 이미 보여진 셈이다. 만일 A가 영행렬이 아니라면

$\mathbf{0} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$, $\mathbf{j} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$ 일 때,

$A\mathbf{x} = \mathbf{0}$ 또는 $\mathbf{j}$ 각각에 대해

$\text{rank}(A) = \text{rank}[A \mid \mathbf{0}]$이고, 또

$\text{rank}(A) = \text{rank}[A \mid \mathbf{j}]$가 성립하므로 해의 존재성은 역시 증명된다.

그리고 실제 게임에서 최소 클릭을 위한 전략을 얻기 위해서는 아래의 알고리즘을 따르면 가장 적은 횟수의 클릭만으로 게임을 끝낼 수 있음을 알게 된다. 아래와 같이 행렬과 벡터 $A, \mathbf{x}, \mathbf{0}, \mathbf{j}, \mathbf{b}$를 잡자.

$$A = \begin{bmatrix} 1&1&0&1&0&0&0&0&0 \\ 1&1&1&0&1&0&0&0&0 \\ 0&1&1&0&0&1&0&0&0 \\ 1&0&0&1&1&0&1&0&0 \\ 0&1&0&1&1&1&0&1&0 \\ 0&0&1&0&1&1&0&0&1 \\ 0&0&0&1&0&0&1&1&0 \\ 0&0&0&0&1&0&1&1&1 \\ 0&0&0&0&0&1&0&1&1 \end{bmatrix}$$

$$\mathbf{x} = [a, b, c, d, e, f, g, h, i]^T,$$

$$\mathbf{0} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T,$$

$$\mathbf{j} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T,$$

$\mathbf{b} = [b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9]^T$는 초기조건에 해당하는 (0,1)행렬이라고 가정하면,

$$A\mathbf{x} + \mathbf{b} = \mathbf{0} \ (\text{또는 } \mathbf{j})$$

일 때 $\mathbf{x}$를 구하라는 문제가 된다. A^{-1}를 실제 계산하여 보면 아래와 같다.

6) 이상구 (2005). 현대선형대수학, 서울: 경문사.

$$A^{-1}=\frac{1}{7}\begin{bmatrix} -1 & 4 & -1 & 4 & -2 & -3 & -1 & -3 & 6 \\ 4 & -2 & 4 & -2 & 1 & -2 & -3 & 5 & -3 \\ -1 & 4 & -1 & -3 & -2 & 4 & 6 & -3 & -1 \\ 4 & -2 & -3 & -2 & 1 & 5 & 4 & -2 & -3 \\ -2 & 1 & -2 & 1 & 3 & 1 & -2 & 1 & -2 \\ -3 & -2 & 4 & 5 & 1 & -2 & -3 & -2 & 4 \\ -1 & -3 & 6 & 4 & -2 & -3 & -1 & 4 & -1 \\ -3 & 5 & -3 & -2 & 1 & -2 & 4 & -2 & 4 \\ 6 & -3 & -1 & -3 & -2 & 4 & -1 & 4 & -1 \end{bmatrix}$$

실제 $\mathbf{x}$를 구해보면 다음과 같다.

$\mathbf{x} = \mathbf{0}-A^{-1}\mathbf{b}$ (또는 $\mathbf{x} = \mathbf{j}-A^{-1}\mathbf{b}$) (단, $\mathbf{x}$의 모든 성분은 0 또는 자연수)

여기서 $\mathbf{x}'$을 실제 게임의 전략행렬이라고 하자. 즉, 구체적으로 클릭하는 바둑알의 위치를 나타내는 것으로 하자. 즉, 최소클릭을 의미하는 전략행렬 $\mathbf{x}'$을 구하기 위해서는 미리 구한 행렬 $\mathbf{x} = \mathbf{0}-A^{-1}\mathbf{b}$ (또는 $\mathbf{x} = \mathbf{j}-A^{-1}\mathbf{b}$)에 대해 적절한 보정이 필요하게 된다. 실제 바둑판 게임의 경우 그것의 해는 반드시 0 아니면 1 이 될 수밖에 없는 게임이다. 우리가 이 게임에서는 같은 바둑알을 여러 번 클릭하는 것은 규칙에 위배되지 않으므로 계산과정에서 나오는 분수를 정수로 만들어 주기 위해서 적당한 자연수를 곱할 수 있다. 즉, 어떤 바둑알의 $2n$번 클릭은 0번 클릭과 동일하다. 위 알고리즘에서 A^{-1}의 성분들을 모두 자연수로 만들기 위해 7을 곱하고, 2를 법으로 하는 잉여류 합의 연산을 수행하여 최소 클릭수를 찾게 된다. 결국 바둑판을 수식으로 표현하면 마우스의 클릭을 의미하는 2를 법으로 하는 잉여류의 합 연산을 통해 행렬 $\mathbf{x}$는

$$\mathbf{x} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_9 \end{bmatrix} = \begin{bmatrix} 2n_1+a' \\ 2n_2+b' \\ \vdots \\ 2n_9+i' \end{bmatrix} \equiv \begin{bmatrix} a_1' \\ a_2' \\ \vdots \\ a_9' \end{bmatrix}$$

이라고 쓸 수 있다. 반면 목표로 하는 바둑판 행렬에서 최초 바둑판 행렬을 뺀 것으로 표현되는 우변의 행렬의 경우는 계산과정에서 성분들이 음수값을 가질 수 있게 된다. 정수의 2를 법으로하는 잉여류의 합 연산에 대한 0 또는 1의 값으로 계산이 가능하므로 우리의 관심사인 (0,1)-행렬을 구할 수 있다.

이로서 최소한의 클릭으로 바둑알 색깔을 모두 동일하게 만들 수 있는 답을 얻을 수 있다. 즉, 어떤 초기 조건이 주어지더라도 해는 존재하며, 그 해를 구하는 최소 클릭의 방법을 구한 것이다. 그리고 이 알고리즘을 이용해 저자가 C++ 언어로 만든 프로그램으로 실험한 결과가 그림으로 첨부된 것이다. 이 프로그램은 http://matrix.skku.ac.kr/sglee/blackout_win.exe에서 다운 받아 사용할 수 있으며 이를 이용하여 새로 **제작한 자바프로그램은 웹 상에서 바로 사용할 수 있다.**

http://matrix.skku.ac.kr/BlackWhite2/BlackWhite.html

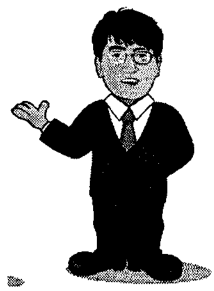
동시에 위의 도구는 본 논문에서 서술하고 증명한 알고리즘이 정확함을 스스로 증명해 준다.

같은 방법으로 $n \times n$ 크기의 바둑판에서의 흑백문제를 생각할 수 있었으며, 실제로 우리는 $n=3,4,\ldots,19$의 모든 경우를 분석하였다. 결과를 간단히 소개하면 $n=4$, 5, 9, 11, 14, 17, 19 경우를 제외한 나머지 경우는 모두 대응하는 행렬 A가 가역행렬이 되어 앞에서 소개한 내용과 똑같은 방법으로 모든 바둑판의 알을 같은 색으로 만드는 알고리즘이 존재하며, $n=4$, 5, 9, 11, 14, 17, 19 경우는 대응하는 행렬 A가 가역행렬이 아님을 보일 수 있었으며 따라서 적절한 제한 조건을 주어서 무수히 많은 해가 존재하도록 한 후, 적절한 알고리즘의 개발이 가능함을 확인하였다. 또 그에 따라 역시 관련 공학적 도구를 제공할 수 있음을 확인하였다. 위의 내용 연구는 교육대학원의 학위논문 수준으로 적절하다고 본다.

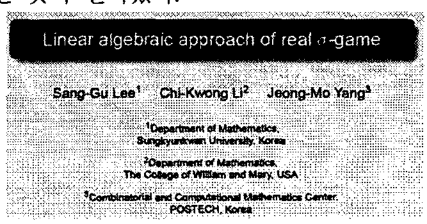
이 진행 과정에서 그간 잘 알려져 있지 않던 '흑백게임과 오토매타 이론' 사이의 관계에 대한 연구 결과를 발견하였다. Pelletier7)는 MERLIN이라 불리는 배터리로 작동되는 장난감에 이용되는 수학에 대하여 소개한 내용을 발표하였으며, Sutner8)는 MERLIN-타입 게임과 Wolfram9) 이 분석한 cellular 오토매타 사이의 밀접한 관계를 소개하였다. 이 분석은 선형대수학과 cellular 오토매타 이론에 근거하여 유향그래프를 이용하였다. 이 내용이 위의 연구와 직접적인 관련을 가지며 흑백게임 연구는 시그마-게임으로 확장된다.

III. 결 론

본 원고에서는 복잡해 보이는 게임도 수학적 문제 해결과 마찬가지로 게임의 기본 구성 원리를 찾아낸다면 문제 해결의 실마리가 보인다는 사실을 위의 흑백 게임의 경우에 적용했다.

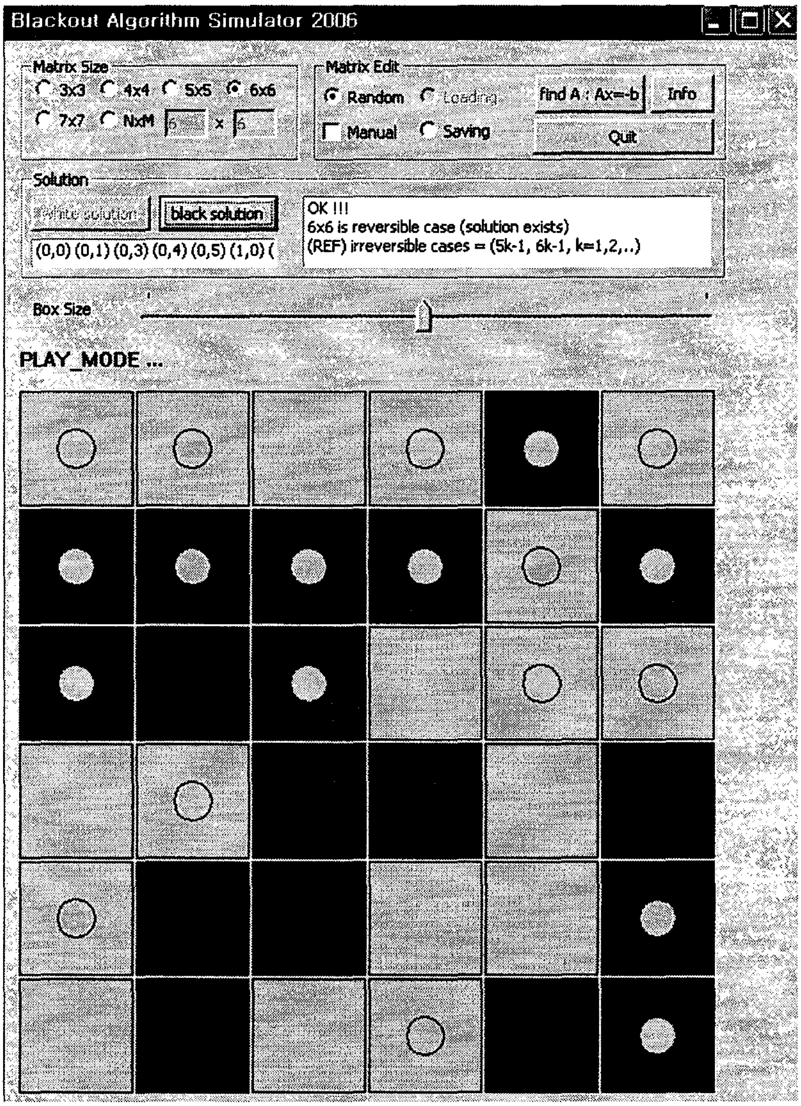


흑백게임의 경우 바둑알을 클릭 한다는 기본 조작(Operation)이 9가지뿐이라는 사실의 파악을 통해, 주어진 초기 조건에 대응하는 3×3 바둑판을 행렬에 대응시키고, 그 행렬에 각각 9가지 기본 조작에 대응하는 9개의 기본행렬을 더하는 것이 위의 9가지 연산과 대응한다는 사실을 인식한 것이 열쇠였다. 이를 통하여 주어진 문제를 연립방정식 문제로 바꾼 후 연립방정식의 해법을 이용하여 최적의 승리 전략을 구하는 알고리즘을 찾은 것이다. 그리고 이 알고리즘을 증명하기 위해 기본적인 선형대수학 지식을 이용했다. 마지막으로는 그 알고리즘을 이용하여 프로그램을 개발하고, 그



7) D. H. Pelletier (1987). Merlin's magic square, Amer. Math. Monthly, 94 pp.143-150
8) K. Sutner (1990). The σ-game and cellular automata, Amer. Math. Monthly, 97, pp.24-34.
9) S. Wolfram (1984), Geometry of binomial coe±cients, Amer. Math. Monthly 91, pp.566-571.

프로그램을 웹상에서 누구나 이용할 수 있는 도구로 제공하였다. 따라서 처음 제시하였던 5가지 질문을 완전히 답한 것이 된다. 이론적으로는 한 발 더 나아가 그 이상의 연구 가능성에 대하여 제시했다.



<저자와 박종빈군이 개발한 프로그램과 6차 흑백게임의 해의 예>