사과 게임의 NP-Hardness 증명

조찬우

선린인터넷고등학교 소프트웨어과

요약

사과 게임(fruit box)은 웹에서 즐길 수 있는 간단한 퍼즐 게임으로, 제한된 시간 안에 높은 점수를 얻는 것을 목표로 한다. 높은 점수를 얻으려면 빠른 속도뿐만 아니라 효율적인 전략의 수립 또한 중요하다. 본 보고서에서는 사과 게임에서 보드의 상태가 주어질 때 얻을 수 있는 최고 점수를 구하는 문제를 결정 문제로 바꾸고, 3-SAT 문제를 해당 결정 문제로 환원할 수 있음을 증명하여 사과 게임에서 최적의 전략을 찾는 문제는 NP-hard임을 보인다.

1. 서론

퍼즐 게임은 정해진 규칙에 따라 문제를 풀어나가는 종류의 게임을 통칭한다. 퍼즐 게임이 많은 사람에게 꾸준히 사랑받으려면 규칙이 쉬우면서도 게임의 진행이 단조롭지 않아야 한다. 게임을 수행하는 단순한 전략이 존재한다면 반복적인 경험을 통해 유저가 이를 파악할 수 있으며, 유저는 해당 게임이 지루하다고 느끼고 흥미를 잃게 된다.

이러한 이유 때문에 지뢰찾기, 테트리스, 슈퍼마리오, 스도쿠 등 오랜 시간 사랑받아 온 퍼즐을 푸는 문제는 NP-hard인 경우가 많다[1]. 어떤 퍼즐을 푸는 문제가 NP-hard라면 아직 해당 퍼즐을 푸는 효율적인 알고리즘이 발견되지 않았고, 존재하더라도 일반 유저가 발견하기는 불가능에 가깝기 때문이다. NP-hard인 퍼즐 게임들은 단순한 규칙 속에서도 매 게임마다 다른 상황이 일어나는 경향이 있다.

사과 게임(fruit box)[2]은 웹에서 즐길 수 있는 간단한 퍼즐 게임으로, 제한된 시간 동안 높은 점수를 얻는 것을 목표로 한다. 사과 게임을 여러 번 진행해 보면 더 많은 점수를 얻을 수 있는데도 시간이 부족한 경우도 있지만, 시간이 많이 남았는데도 더 이상 점수를 얻을 방법을 찾기 어려운 경우도 빈번히 겪게 된다. 이러한 차이가 일어나는 이유 중 하나는 사과 게임에서는 속도뿐만 아니라 효율적인 선택 또한 중요하기 때문이다.

---

1 https://www.secmem.org/blog/2020/12/20/np-complete-problems/ 접속 날짜 2022.06.11

2 https://www.gamesaien.com/game/fruit_box_a/ 에서 플레이해 볼 수 있다.

본 보고서에서는 사과 게임의 규칙을 명확히 정리하고 사과 게임의 임의의 상태에서 얻을 수 있는 최고점을 구할 수 있다면 풀 수 있는 결정 문제를 정의한 뒤, NP-hard라고 알려진 3-SAT 문제를 사과 게임의 결정 문제로 환원할 수 있음을 증명하여 해당 결정 문제가 NP-hard임을 보인다. 사과 게임을 플레이하는 최적의 전략이 존재한다면 사과 게임의 결정 문제를 풀 수 있으므로 사과 게임은 NP-hard이다.

2. 문제 정의

2-1. 사과 게임의 정의

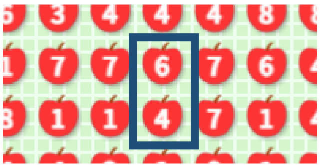
사과 게임은 아래와 같은 퍼즐 게임이다.

- 1~9 사이의 숫자가 무작위로 채워진 격자판이 주어진다.
- 유저는 각 변이 X축 또는 Y축과 평행한 직사각형을 그린다.
- 유저가 그린 직사각형에 포함된 칸에 쓰여진 수의 합이 정확히 10이라면 직사각형에 포함된 칸이 모두 지워지며, 지워진 칸의 수만큼 점수를 얻는다.
- 한 번 지워진 칸은 이후 그린 직사각형에 포함되더라도 수의 합을 구할 때와 지워진 칸의 개수를 구할 때 무시된다.
- 제한 시간 안에 높은 점수를 얻는 게임이다.

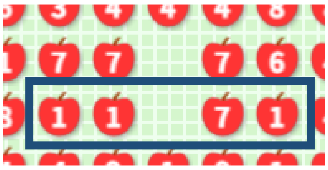
Figure 1. 사과 게임의 초기 상태

본 보고서에서는 설명의 편의를 위해 주어지는 격자판을 ‘보드’, 보드의 각 칸을 ‘사과’ 라고 부른다.

보드의 좌측 상단 일부를 잘라 보자.



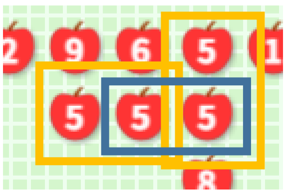
위 상태에서 푸른 박스와 같이 직사각형을 그리면 직사각형 내부의 숫자의 합이 10이 되므로 해당 사과 두 개는 지워지고, 2점을 얻는다.



6과 4가 지워지면 1-1-7-1을 선택해 10을 만들 수 있다. 이처럼 게임이 진행되고 사과가 지워지면 이전에는 불가능했던 직사각형을 그릴 수 있게 되기도 한다.

사과 게임에서는 같은 상태에서라도 어떤 사과를 먼저 지우는지에 따라 얻을 수 있는 점수가 달라지기도 한다.

위 그림에서는 5-5를 만들 수 있는 방법이 여러 가지인데, 아래 그림의 파란색 상자처럼 직사각형을 그린다면 2점을 얻지만 노란색처럼 두 번 직사각형을 그리면 4점을 얻을 수 있다.



이러한 국소적인 손익은 계산하기 쉽지만, 보드 전체를 고려하였을 때 가능한 최선의 수를 찾기는 쉽지 않다.

2-2. 결정 문제 정의

사과 게임은 보드의 상태가 주어질 때 얻을 수 있는 최고점을 구하는 최적화 문제로 생각할 수 있으며, 사과 게임의 결정 문제는 다음과 같이 정의할 수 있다.

- 초기 상태에서 0개 이상의 사과가 지워진 보드가 주어질 때, 현재 상태에서 k점 이상의 점수를 얻을 수 있는가?

사과 게임을 플레이하는 최적의 전략이 존재한다면, 보드의 상태가 주어졌을 때 최적의 전략을 시뮬레이션하여 얻을 수 있는 최고점을 계산할 수 있다. 수행할 동작이 순서대로 주어진다면 이를 시뮬레이션하는 과정은 보드의 크기에 대한 다항 시간임이 자명하므로, 사과 게임의 최적 전략을 찾는다면 사과 게임의 결정 문제를 풀 수 있다.

3. 3-SAT을 사과 게임으로 환원

3-1. 3-SAT의 정의[3]

- 불 변수나 그 역을 **리터럴**이라고 한다.
- 리터럴들이 논리합 '∨'로 연결된 것을 **절**이라 한다.
- 절들이 논리곱 '∧'로 연결된 것을 **CNF**라 한다.
- 각 절이 정확히 k개의 서로 다른 리터럴로 되어 있는 CNF를 k-CNF라 한다.
- 3-SAT은 다음과 같은 문제이다.
    - 입력: 3-CNF 형식의 부울식 ∮
    - 질문: ∮가 만족 가능한가?

3-2. 환원 과정

주어진 3-CNF 식의 변수의 개수를 N, 절의 개수를 M이라 하자. 보드의 상태를 Figure 2. 와 같이 구성한다. 이해를 위해 일부 칸을 색칠하고 설명을 넣었으며, 격자로 표시된 부분이 보드이다.

- 각 불 변수마다 5개의 열을 할당한다. 5개의 열 중 2번째 열은 참, 4번째 열은 거짓을 나타낸다. 1, 3, 5열은 다른 변수에 해당하는 칸과의 상호작용을 막는 벽 역할을 한다.
- 각 불 변수의 참 열과 거짓 열의 맨 위부터 7을 나란히 M개 배치하고, M개의 7 아래에 6을 하나씩 배치해 7의 아래를 막는다. 이후 두 6 사이에 4를 배치한다.
- 각 불 변수의 1,3,5열에 9를 최소 M개씩 채워 7의 양 옆을 막는다.
- M개의 절을 순서대로 나타낸다. 절을 나타내는 행 사이에는 최소 1행의 공백이 있어야 하며, 모두 변수를 나타낸 칸 아래에 위치해야 한다. 절을 나타내는 방법은 다음과 같다.
    - 절을 구성하는 세 리터럴에 해당되는 열들과 해당 절을 나타내는 행이 교차하는 칸

[3] 문병로,「쉽게 배우는 알고리즘」(한빛미디어) p439-441

에 3을 배치한다. 즉, 각 3들은 해당 절의 리터럴 하나와 대응된다.

- 해당 절을 나타내는 행 중 어떤 불 변수에도 할당되지 않은 왼쪽 끝 열과 오른쪽 끝 열에 2, 9를 순서대로 배치한다. 9는 2가 다른 행과 상호작용하는 것을 막는 역할이므로 마지막 절에는 없어도 무방하다.

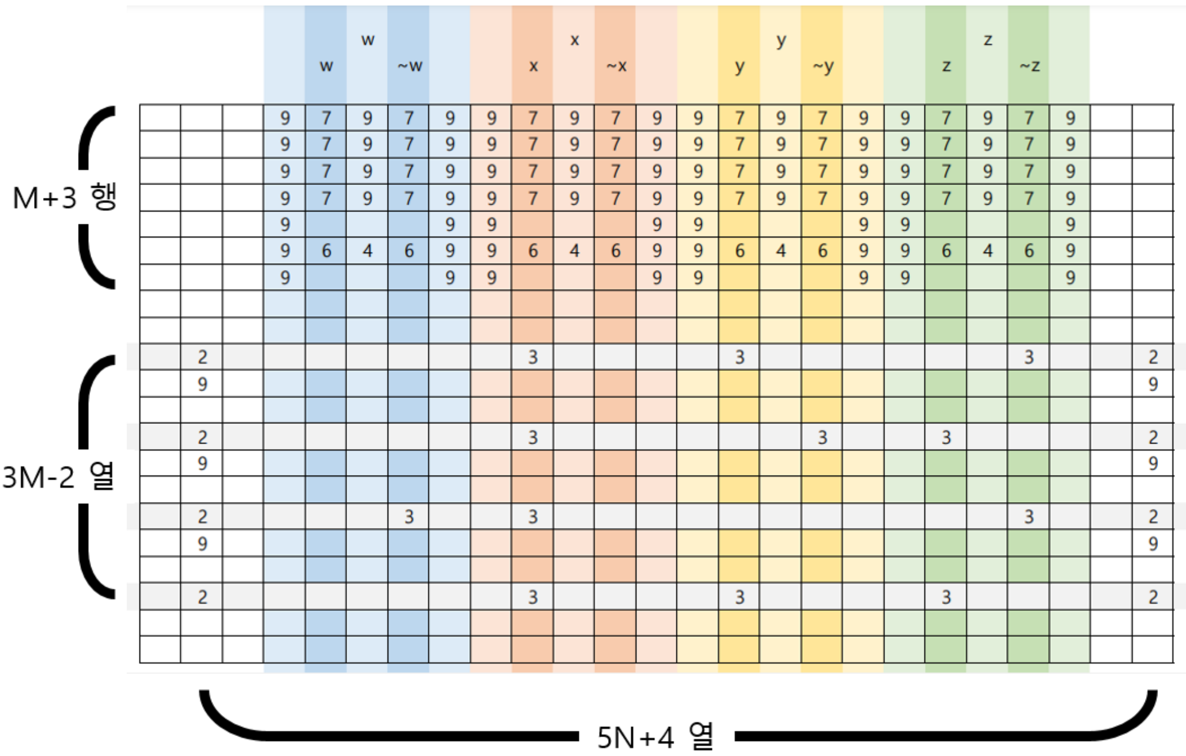


Figure 2. 3-CNF 식에 따른 보드의 구성 예시

Figure 2. 는 3-CNF 식 $(x \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (\bar{w} \vee x \vee \bar{z}) \wedge (x \vee y \vee z)$ 이 주어졌을 때의 보드 구성이다. 하나의 절에 같은 리터럴이 여러 개 있다면 그 중 하나만 바른 위치에 두고 아무 변수의 1,5열 중 하나에 나머지 중복되는 리터럴들을 나타내주면 된다.

이처럼 보드를 구성했을 때 취할 수 있는 최적의 전략을 알아보자.

우선 당장의 선택지는 N개의 6-4 조합이다. 이때 각 변수마다 참 열의 6, 거짓 열의 6 중 어떤 것을 지울지 선택해야 한다. 이는 3-SAT 문제에서 변수마다 참/거짓 값을 할당하는 것과 같다. Figure 3. 는 6-4 조합을 모두 없앤 상태로, w, x에는 거짓을, y, z에는 참을 할당한 모습이다.

6-4 조합을 없앴다면 절의 리터럴을 나타내는 3과 6이 없어진 열의 7을 직사각형으로 묶을 수 있게 된다. 6이 없어진 열은 해당 변수의 값을 의미하므로, 3과 7을 없애려면 3에 대응되는 리터럴의 값이 참이어야 한다.

각 절의 리터럴에 대응되는 3 중 하나만 없앨 수 있다면 해당 행을 직사각형으로 묶어 2-3-3-2 조합으로 10을 만들 수 있다. 즉, 각 절의 리터럴 중 하나가 참이라면 참인 리터럴에 대응되는 3과 7을 지우고, 나머지 리터럴들을 2-3-3-2 조합으로 모두 지울 수 있고, 6점을 얻게 된다.

			w	w	~w			x	x	~x			y	y	~y			z	z	~z			
		9	7	9	7	9	9	7	9	7	9	9	7	9	7	9	9	7	9	7	9		
		9	7	9	7	9	9	7	9	7	9	9	7	9	7	9	9	7	9	7	9		
		9	7	9	7	9	9	7	9	7	9	9	7	9	7	9	9	7	9	7	9		
		9	7	9	7	9	9	7	9	7	9	9	7	9	7	9	9	7	9	7	9		
		9				9	9				9	9				9	9				9		
		9	6			9	9	6			9	9			6	9	9			6	9		
		9				9	9				9	9				9	9				9		
2								3					3							3			2
9																							9
2								3							3			3					2
9																							9
2					3			3												3			2
9																							9
2								3					3					3					2

Figure 3. 각 변수에 값을 할당한 상태

임의의 절에서 2개 이상의 리터럴이 참이라도 하나의 리터럴만 7과 묶어 지우고 나머지는 2-3-3-2 조합으로 지우는 것이 이득이다. 또한 어떤 절에 속한 모든 리터럴의 값이 거짓이라면 해당 절을 나타내는 행에 속한 칸은 하나도 지울 수 없다. 따라서 어떤 절이 참이라면 6점을 얻게 되고, 거짓이라면 0점을 얻게 된다.

각 변수마다 6-4 조합으로 2점씩을 얻을 수 있으므로 2N점을 얻을 수 있고, 변수에 값을 적절히 할당하여 주어진 3-CNF 식을 참으로 만들 수 있다면 모든 절마다 6점씩을 얻을 수 있으므로 6M점을 더 얻을 수 있다. 따라서 구성된 보드에서 $2N + 6M$점 이상을 얻을 수 있다면 주어진 3-CNF 식을 만족시킬 수 있으며, 역도 성립한다. 보드의 크기는 부울식의 길이에 대한 다항식 형태이므로 부울식이 주어지면 다항 시간에 보드를 구성할 수 있고, 구성된 보드에서 $2N + 6M$점 이상을 얻을 수 있는지를 판단할 수 있다면 3-SAT 문제를 풀 수 있다. 즉, 3-SAT 문제를 다항 시간에 사과 게임의 결정 문제로 환원할 수 있으므로 사과 게임의 결정 문제는 NP-hard이다.

사과 게임의 결정 문제의 답이 YES라는 근거는 수행할 동작, 즉 그릴 직사각형이 순서대로 주어진 목록이다. 수행할 동작의 목록이 주어진다면 순서대로 시뮬레이션하는 과정은 자명히 다항 시간에 수행할 수 있으므로 사과 게임의 결정 문제는 NP이다.

즉, 사과 게임의 결정 문제는 NP-Complete이며, 최적화 문제인 사과 게임은 NP-hard이다.