

셀룰러 오토마타 상에서 과반수 문제의 상태전이 함수

박종우, 위규범
아주대학교 소프트웨어학과
e-mail:jwmt2@ajou.ac.kr

Finding Transition Rules for Majority Problem on Cellular Automata

Jongwoo Park, Kyubum Wee
Dept of Software, Ajou University

요 약

과반수 문제는 전역 문제(global problem) 문제이기 때문에 국소 정보(local information)를 이용해야 하는 셀룰러 오토마타(cellular automata; CA) 상에서 풀기 어려운 문제이다. 본 논문에서는 일차원 CA에서 과반수 문제를 푸는 CA를 찾는 방법을 제안한다. 상태전이 함수를 CA에서 일반적으로 사용되는 규칙 표(rule table)가 아닌 CMR(conditionally matching rules)로 나타내고 진화 알고리즘을 적용하였다. 제안한 방법으로 다수의 서로 다른 규칙들을 찾아낼 수 있었고 찾아낸 규칙은 이전 연구에서 찾아낸 규칙과 유사한 성능을 보여주었다. CA의 문제를 해결하는데 CMR에 진화 알고리즘을 적용하는 것이 효용성이 있음을 보였다.

1. 서론

셀룰러 오토마타(cellular automata; CA)는 규칙적인 격자 형태로 유한한 상태(state)를 가지는 셀(cell)들로 정의된다. 각 셀에 대한 이웃(neighborhood)은 그 셀에 대한 관계로 정의한다. 시간 $t=0$ 일 때, 각 셀의 상태의 조합을 초기 구성(initial configuration; IC)이라 한다. 현재 시각의 각 셀과 그 이웃들의 상태에 따라 고정된 전이 함수(transition function)에 의해 다음 시각의 셀의 상태가 정해진다. CA는 소수를 구하는 문제, 동기화 문제, 과반수 문제 등 계산적인 문제[1, 2], 자기 복제와 같은 생명체의 특성을 시뮬레이션 하는 인공 생명 분야의 문제[3, 4], 또는 물리적 현상을 설명하는 모델[5] 등의 문제에 사용되고 있다.

과반수 문제(majority problem)는 이진 상태(binary state)를 가지는 일차원 이산 공간에서 초기 구성이 주어졌을 때, 모든 셀의 상태가 초기 구성에서 더 많았던 상태로 수렴하도록 하는 규칙을 찾는 문제이다. 어떤 셀의 다음 상태를 결정할 때, 전체 셀의 상태를 인식하지 못하고 자신과 이웃하는 셀의 상태만 인식할 수 있기 때문에 정확하게 풀기 어려운 문제 중 하나이다. 특히 0과 1의 수가 비슷한 경우에는 과반수를 이루는 상태를 찾기 어렵다.

Mitchell은 셀과 이웃들이 가질 수 있는 모든 상태에 대한 다음 상태 값을 표로 나타낸 규칙 표(rule table)를 전이 함수로 사용하고, 진화 알고리즘을 이용하여 과반수 문제의 규칙 표를 찾아내었다[6]. IC에서 0과 1의 수가 비

슷한 경우에 약 76.9%의 정확도를 보이는 규칙 표를 찾아내었다.

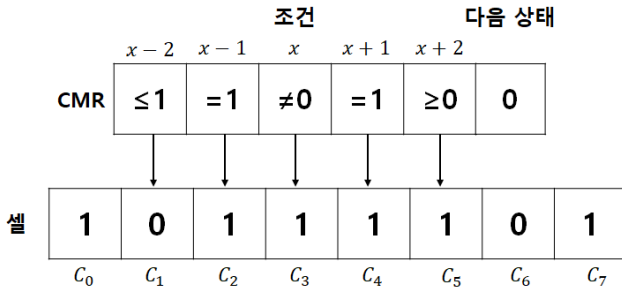
CMR(conditionally matching rules)은 CA 상에서 상태전이 함수를 나타내는 방법 중 하나이다. Bidlo는 CMR과 진화 알고리즘을 이용하여 일차원 CA 상에서 제곱을 계산하는 규칙과 이차원 CA 상에서 자기 복제를 하는 규칙을 효과적으로 찾아낼 수 있음을 보였다[7, 8, 9].

Bidlo의 연구는 일차원 CA에서 제곱함수를 계산하는 규칙을 찾음으로서 CMR의 유용성을 보였다. 본 연구는 일차원 CA에서 많이 연구된 대표적인 문제인 과반수 문제에 적용하여 CMR의 효용성을 보였다.

본 논문의 제 2장에서는 CMR에 대해서 설명하고 제 3장에서는 과반수 문제에 대한 CMR을 찾는 진화 알고리즘을 설명한다. 제 4장에서는 본 연구에서 찾아낸 CMR의 성능을 측정하고, 이전 연구의 결과와 비교한다.

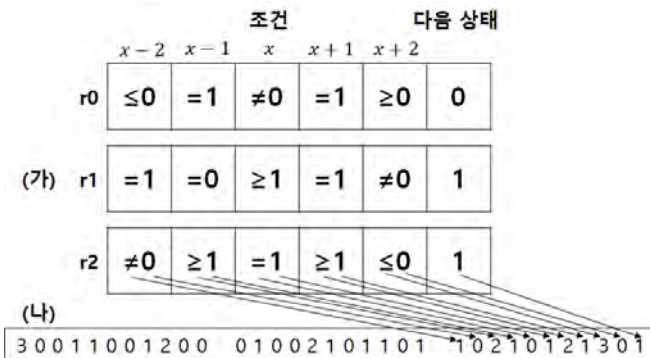
2. 일차원 셀룰러 오토마타 상에서 CMR

기존 CA의 전이 함수는 규칙 표가 사용되었다. 본 논문에서 사용한 CMR은 (그림 1)과 같이 셀의 상태에 관한 조건들과 다음 상태로 구성된다. 조건은 비교 연산자와 값으로 이루어져 있다. 비교 연산자의 자리에는 '같다', '같지 않다', '크거나 같다', '작거나 같다' 이 네 가지가 올 수 있다. 값은 CA 상에서 셀이 가질 수 있는 상태의 개수를 S 라고 했을 때, 0에서 $S-1$ 까지의 값을 가질 수 있다.



(그림 1) $N=8$ 인 CA에서 C_3 위치의 셀이 $r=2$ 인 CMR의 조건에 비교 되는 모습. C_3 의 다음 상태는 0이 된다.

다음 상태를 결정할 셀의 위치를 x 라 하고, r 을 지름이라고 하면 $x-r$ 부터 $x+r$ 까지 조건에 일치하는지 검사한다. 셀의 양 끝은 환형으로 연결된 형태이다. 따라서 셀 공간의 너비를 N 이라고 했을 때, (그림 1)과 같이 $N=8$ 인 CA 상에서 C_{-1} 는 C_7 의 값을 이용한다. 본 연구에서 $r=3$, $N=149$ 으로 설정하여 실험 하였다.



(그림 2) CMR(가)과 CMR을 데이터로 표현하는 방법(나)

CMR은 (그림 2)의 (가)와 같이 여러 개의 규칙으로 이루어진다. 어떤 셀의 다음 상태를 결정할 때, 목록에서 가장 먼저 조건을 만족하는 CMR의 다음 상태로 결정된다. 만약 목록에서 조건을 만족하는 CMR이 존재하지 않는다면 현재 상태를 유지한다. (그림 2)의 (나)는 CMR들이 진화 알고리즘에서 사용되는 데이터가 어떻게 표현되는지 보여준다. 조건 연산자는 각각 ‘같다’는 0, ‘같지 않다’는 1, ‘크거나 같다’는 2, ‘작거나 같다’는 3으로 일대일 대응된다.

3. 진화 알고리즘

CMR의 집단(population)에 진화 알고리즘을 적용하여 다수결 문제의 해(solution)를 이루는 CMR을 찾아내었다. 진화 알고리즘의 개체 수는 8로 설정하였으며, 첫 세대의 개체는 무작위로 8개가 생성된다. 개체 수가 많은 경우 한 세대를 계산하는데 오랜 시간이 걸리기 때문에 일반적인 진화 알고리즘보다 개체 수를 낮게 설정 하였다. 대신 총

분한 세대 동안 진화시켰다.

진화 알고리즘의 새로운 세대는 다음과 같이 생성한다. (가) 8개의 개체 중 무작위로 4개의 개체를 선택 후 그 중 적합도가 가장 높은 개체를 부모로 하여 하나의 자식을 생성한다. (나) 자식에 돌연변이를 적용한다. (다) 8개의 자식이 생성될 때까지 (가), (나)를 반복한다. (라) 진화에 성공한 개체를 발견한 경우 진화를 종료하고, 발견하지 못한 경우 5,000세대까지 (가)-(라)를 반복한다. 5,000세대까지 진화에 성공한 개체를 찾지 못하면 진화에 실패한 것으로 간주하고 진화를 종료한다.

CMR은 진화 알고리즘의 하나의 개체를 이룬다. 하나의 개체를 다수의 IC에 적용하여 각 IC에서 과반수를 이루어야 할 상태의 개수를 그 IC에 대한 적합도로 하고, 모든 IC에 대한 적합도의 합을 그 개체의 적합도로 정의한다. 더 자세히 설명하면, 각 IC를 $2N$ 번의 시간 단계(time-steps) 만큼 진행시켜 N 개의 셀 중에서 과반수 상태에 도달한 셀의 개수를 계산한다. 이 때 사용된 IC의 수를 $T=20$ 이라 하면 CMR이 가질 수 있는 최대 적합도는 $N*T=149*20=2980$ 이며, 이 적합도를 가지는 개체는 진화에 성공한 개체로 간주한다.

돌연변이는 다음과 같이 발생시킨다. 0에서 2까지의 정수를 무작위로 생성한다. 생성된 정수는 염색체에서 돌연변이를 일으킬 횟수이다. 그러므로 생성된 정수가 0인 경우는 돌연변이를 일으키지 않는다. 돌연변이에 의하여 선택된 정수가 CMR의 조건에서 값 또는 다음 상태를 의미한다면 0 또는 1 중 무작위로 선택된 값으로 교체하고, CMR의 조건 연산자가 선택되면 0에서 3까지의 정수 중 무작위로 선택된 값으로 교체한다.

4. 실험 결과

CMR의 적합도를 구하기 위하여 CMR을 적용하는 IC를 다음과 같이 생성한다. IC에서 상태가 0인 셀의 비율을 ρ 라고 하자. ρ 를 세 가지 방법으로 설정하여 IC를 생성하여 실험했다. 각 실험을 E_A , E_B , E_C 로 표기한다. E_A 는 $\rho=0$ 에서 $\rho=1$ 까지 0.05의 간격으로 IC를 한 번씩 생성한다. E_B 는 $\rho=0.5$ 로 IC를 20번 생성한다. N 은 홀수이기 때문에 정확히 $\rho=0.5$ 가 되지 않는다. IC를 생성할 때 각 셀의 상태를 1/2 확률로 정하여 0 또는 1로 설정한다. E_C 는 $\rho=0.4$ 에서 $\rho=0.6$ 까지 0.01의 간격으로 한 번씩 생성한다.

진화 알고리즘에서 찾아낸 개체를 I 라 했을 때 $P_m(I)$ 를 방법 m 으로 생성된 다수의 IC에서 개체 I 가 과반수를 정확하게 구한 비율이라 하고, 이를 I 의 성능이라고 한다. IC 집합을 생성하는 방법 a 는 $\rho=0$ 에서 $\rho=1$ 까지 0.1의 간격으로 각 ρ 값에 대해서 IC를 1,000번씩 생성한다. 방법 b 는 $\rho=0.5$ 로 하여 과반수를 이루는 상태를 찾기 어려운 IC를 10,000번 생성한다.

실험의 결과에서 E_A 는 모든 진화 알고리즘의 실행에서 진화에 성공한 개체를 찾았지만, 적합도를 올리기 쉽기 때문에 최대 성능이 E_C 보다 약 22% 떨어진다. E_B 는 적합도를 올리기 가장 어렵기 때문에 찾아낸 개체가 없었다. E_C 에서 두 번 중 한 번은 성공한 개체를 찾을 수 있고, 실험에서 발견된 개체는 비교적 높은 성능을 보였다. 따라서 E_C 의 ρ 를 설정하는 방법이 가장 적합하다고 할 수 있다. 각 실험의 진화 알고리즘은 20회 실행되었고, 성공 횟수는 20회 중 진화에 성공한 개체를 찾은 실행의 수이다. 최대 $P_b(I)$ 는 실험에서 찾은 개체의 성능 중 가장 높은 것이다.

<표 1> IC 생성 방법에 따른 결과

표기	ρ			ρ 당 IC 생성횟수	성공 횟수	최대 $P_b(I)$
	최소	최대	간격			
E_A	0	1	0.05	1	20	51.79%
E_B	0.5	0.5	-	20	0	-
E_C	0.4	0.6	0.01	1	11	73.35%

CMR 규칙의 수에 따라서 성능에 미치는 영향을 알아보기 위해 CMR 규칙의 수를 30, 40, 50개로 설정하여 실험하였다. 진화 알고리즘에서 IC를 생성하는 방법은 E_C 와 동일하다.

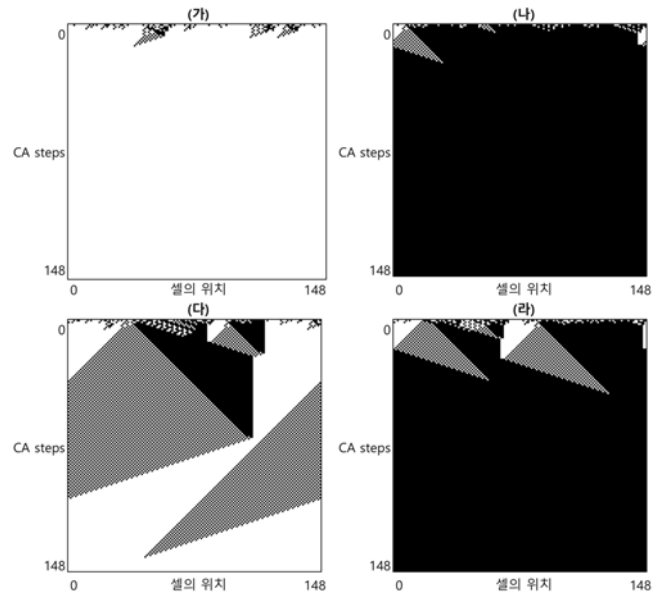
CMR 규칙의 수에 따라서 진화에 성공하는 개체를 찾아내는 비율은 유의미한 차이를 보이지 않는다. CMR의 규칙이 40개일 때 다른 두 개의 실험보다 높은 성능을 보이지만 CMR의 규칙 50개일 때와 큰 차이를 보이지 않는다. 또한 본 연구에서 찾아낸 개체는 Mitchell 이 찾아낸 규칙의 성능과 근접하여 상태전이가 과반수 문제의 규칙을 찾는데 효용성이 있음을 보여준다.

<표 2> 진화 알고리즘을 20회 실행했을 때 CMR 규칙의 수에 따른 결과와 Mitchell의 결과 비교

방법	성공횟수	최대 $P_a(I)$	최대 $P_b(I)$
CMR 30개	11	92.24%	73.35%
CMR 40개	12	96.16%	75.83%
CMR 50개	12	94.01%	75.13%
Mitchell	-	-	76.90%

* Mitchell의 연구에서 발견한 규칙은 $P_b(I)$ 에 대해서만 성능이 측정되어있다.

(그림 3)은 본 연구의 실험에서 찾은 개체들 중 가장 성능이 뛰어난 것을 ρ 값에 따라 실행한 것을 시각화 한 것이다. CA는 $2N=298$ 단계만큼 수행했으나, N 단계 이전에 이미 과반수 상태로 수렴하였으므로, (그림 3)은 $N=149$ 단계까지만 보여준다. (가)는 $\rho=0.25$ 일 때의



(그림 3) 실험 결과에서 찾은 개체 중 성능이 가장 뛰어난 것의 ρ 에 따른 과반수 문제의 실행 결과. 흰색은 0인 상태이고 검은색은 1인 상태이다.

실행 결과로 모든 셀이 0으로 수렴하는 것을 볼 수 있다. (나)는 $\rho=0.75$ 인 경우의 실행 결과로 모든 셀이 1로 수렴한다. (다), (라)는 $\rho=0.5$ 로 설정하여 각각 0이 많은 경우와 1이 많은 경우의 실행 결과이며 과반수 상태를 정확히 구하는 과정을 보여준다.

5. 결론

본 연구에서는 CA상에서 많이 연구된 잘 알려진 과반수 문제에 적용하여 CMR의 효용성을 보였다. 제시한 방법으로 실험한 결과 과반수 문제를 푸는 다수의 CA를 찾아낼 수 있었다. 찾아낸 CA가 기존의 연구에서 과반수 문제의 CA와 유사한 성능을 보여주었다.

적합도 함수를 개선하여 CA의 성능을 높일 수 있을 것으로 기대하여 연구를 진행 중이다. 또한 CMR은 이진 상태 CA보다 다수의 상태를 가지는 CA의 규칙을 표현하는데 적합하므로 다수의 상태를 가지는 과반수 문제로 확장하는 연구를 진행 중이다.

참고문헌

[1] S. Wolfram, "A new kind of science: A 15-year view," Complex Syst., vol. 26, (3), pp. 197-224, 2017.
 [2] R. Das, J. P. Crutchfield, M. Mitchell and J. E. Hanson, "Evolving globally synchronized cellular automata," in Proceedings of the 6th International Conference on Genetic Algorithms, 1995, pp. 336-343.
 [3] J. Von Neumann, "Theory of Self-Reproducing Automata," A. W. Burks, Ed. Urbana, IL, USA: Univ. Illinois Press, 1966.

- [4] C. G. Langton, "Studying artificial life with cellular automata," *Phys D Nonlinear Phenom*, vol. 22, (1-3), pp. 120-149, 1986.
- [5] T. Ostoma and M. Trushyk, "Cellular automata theory and physics," *Arxiv Preprint physics/9907013*, 1999.
- [6] M. Mitchell, "An Introduction to Genetic Algorithms," MIT press, 1998.
- [7] M. Bidlo, "On routine evolution of complex cellular automata," *IEEE Transactions on Evolutionary Computation*, vol. 20, (5), pp. 742-754, 2016.
- [8] M. Bidlo, "Evolving multiplication as emergent behavior in cellular automata using conditionally matching rules," in *IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 2732-2739.
- [9] M. Bidlo and Z. Vasicek, "Evolution of cellular automata with conditionally matching rules," in *IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 1178-1185.