

## MAX-CUT PROBLEM WITH PURE GENETIC ALGORITHM

### 맨앞자리팀

2019-1130	2021-1291	2021-1301	2021-1327	2021-1343
김영은	가상은	권수진	류경빈	방기연

## 1 유전에 대하여

[Table 1]은 유전에 대한 4 가지 질문이다. 이에 대한 팀원별 답변은 [Table 2]에 있다.

Table 1 유전에 대한 질문

1. 외모뿐만 아니라 나의 (생물학적인) 능력의 99.999% 이상은 유전자가 결정한다.
2. 내 가치관, 정치적 성향, 습관 등은 유전인가?
3. 감기에 걸려 항생제 5 일치 처방 후 복용 중 4 일차에 증상 호전 → 남은 약을 먹을 것인가 말 것인가? (의사의 조언은 구할 수 없다)
4. 상위 구조에서 하위 구조에서 갖지 못한 특성이 나타날 수 있나?

Table 2 유전에 대한 답변

[김영은] (1) 그렇다. (2) 그렇다. 다만 자신의 의지로 약간 바꿀 수 있다. (3) 의사의 지시가 없는 한 복약 지시를 지켜야 한다. (4) 가능하다. 부모의 모든 특성이 자식에게 유전되지 않는다.
[가상은] (1) 그렇다. (2) 유전될 수 있지만, 자유 의지나 외부 요소에 의해 바뀔 수 있다. (3) 의사가 처방해준 5 일치 약을 모두 복용해야 한다. (4) 그렇다.
[권수진] (1) 그렇다. (2) 그렇다. 하지만 환경적 요인이 함께 작용하여 변할 수 있다. (3) 남아 있는 바이러스가 오히려 면역이 생길 수 있으므로 남은 약을 모두 복용한다. (4) 그렇다.
[류경빈] (1) 그렇다. (2) 그렇다. 하지만 변수는 있다. (3) 처방해준 약을 복용하는 것이 맞다. (4) 충분히 가능하다.
[방기연] (1) 그렇다. (2) 그렇다. 하지만 환경적 요인으로 인해 바뀔 수 있다. (3) 먹을 것이다. 먹지 않을 경우 몸에 남은 바이러스가 약에 면역이 될 수 있다. (4) 나타날 수 있다.

## 2 문제 해결 설계

팀원별로 max-cut 문제를 해결하기 위한 순수 GA 구조를 설계한다.

### 2.1 해의 표현

Table 3 팀원별 해의 표현

[김영은] A, B 를 그래프 노드 수만큼 나열한 문자열로 표현. 문자열 인덱스와 노드 번호가 같고, 각 자리 글자가 노드가 속한 그룹을 의미함.
[가상은] 이진 인코딩 방식. 해의 길이는 그래프 정점의 총 개수가 되고, 그래프를 둘로 나누었을 때 정점 집합 S 에 속하면 1, 집합 S'에 속하면 0 으로 표현.
[권수진] 이진 인코딩 방식으로 노드 수만큼 나열.
[류경빈] 이진 인코딩 방식 선택. 그래프를 고정적으로 2 개의 서브그래프로 분할, 하나의 서브그래프는 0, 다른 것은 1 로 표현.
[방기연] 이진문자열. 0 또는 1 로 이루어져 있으며 그래프 노드 수가 문자열 길이가 되고 출력된 해에 포함된 숫자가 인덱스인 요소를 1, 아닌 요소를 0 으로 표현.

## 2.2 사용한 GA 구조

Table 4 팀원별 GA 구조

<p><b>[김영은]</b> (1) 노드 수만큼 A, B 선택해 해 생성, 유효성 검사와 함께 cost 계산하여 초기 부모 집합으로 사용. (2) cost 토너먼트로 각 부모 선택. 자식은 하나 생성, 해의 각 자리마다 50% 확률로 부모의 유전자 하나 선택, 돌연변이는 해의 각 자리마다 일정 확률로 값을 재선택. 자식이 무효하면 제거. (3) 일정량 생성된 자식은 각자 자신보다 cost 가 조금 작은 유전자를 대체, 대체할 유전자가 없다면 제거. (4) 한 세대에서 일정 비율 이상의 자식이 대체하지 못하고 제거되거나 제한 시간이 임박했을 때 보유한 최선의 해를 반환.</p>
<p><b>[가상은]</b> 1. 정점 개수만큼 0, 1 을 랜덤으로 뽑아 해를 생성, 각 해의 cost 는 정점의 가중치 합. 2. 토너먼트 방식을 2 번 진행해 두 부모를 선택, cost 가 큰 해가 이길 확률은 60%. 자식은 하나 생성, 교차점은 해 길이의 3/4 지점. 각 자식의 gene 마다 0.05% 확률로 변이 발생, 후속 세대 컨테이너에 추가. 3. 변이로 나온 해의 수가 한 세대를 이루는 양이 되면 통째로 세대교체. 4. 제한 시간이 다 되면 가장 우수한 해를 반환.</p>
<p><b>[권수진]</b> 노드 수만큼 0, 1 나열해 해를 구성, 가중치 총합이 클수록 우월하다 취급(우성). 룰렛휠로 두 부모 비복원 추출, 자식 유전자는 각 자리마다 6:4 확률로 우/열성 유전자 선택. 한 세대가 형성되면 자식풀을 새로운 부모풀로 일괄 대체. 가중치 총합이 더 이상 갱신되지 않거나 제한 시간이 임박하면 보유한 최고의 해를 반환.</p>
<p><b>[류경빈]</b> 1. 서브그래프의 수를 랜덤하게 생성, 그 개수만큼 인덱스 선택해 컨테이너의 뒤쪽으로 보내 초기 풀 형성, 각 해의 Cost 는 노드의 가중치 합. 2. 룰렛휠로 두 부모 선택. 교배 후 자식은 1% 확률로 하나의 gene 에 대해 변이 발생. 3. 자식은 하나 생성, 부모의 최악의 유전자와 비교해 우월하다면 그것을 대체, 아니라면 부모 유지. 4. 제한 시간이 다 되었을 때 마지막으로 도출된 cost 가 가장 높은 최선의 해를 반환.</p>
<p><b>[방기연]</b> 1. 랜덤으로 노드를 선택해 생성한 해 200 개가 첫 세대. Cost 는 해당하는 가중치의 합. 2. 토너먼트로 두 부모 선택. One Point 교차로 자식 생성, 각 노드에 0.01% 확률로 변이. 3. 생성된 자식으로만 세대 하나 구성하여 전 세대와 통째로 교체. 4. 1000 번째 세대가 되거나 시간 제한이 끝나면 존재하는 제일 좋은 해를 반환.</p>

## 2.3 사용한 연산자에 대한 설명

Table 5 팀원별 GA 연산자

<p><b>[김영은]</b></p> <ul style="list-style-type: none"> <li>* validation: 모든 해가 생성될 때 그것이 유효한지 검사.</li> <li>* selection: cost 를 기준으로 토너먼트, 뽑힌 cost 를 갖는 해 중에서 랜덤 선택.</li> <li>* crossover: 해의 각 자리마다 50% 확률로 두 부모의 유전자 중 하나 선택.</li> <li>* mutation: 해의 각 자리마다 일정 확률로 값을 다시 선택.</li> <li>* replacement: generational GA 방식 이용, 자신보다 cost 가 약간 작은 것과 대체한다.</li> </ul>
<p><b>[가상은]</b></p> <ul style="list-style-type: none"> <li>* selection: 토너먼트 선택 2 번, 우수한 해의 승률은 60%.</li> <li>* crossover: 해 길이의 3/4 지점에서 두 부모의 gene 을 swap, 우수한 부모의 해가 더 많은 자식이 변이 대상.</li> <li>* mutation: 해의 각 gene 마다 0.05% 확률로 flip, 변이된 해는 후속 세대 컨테이너에 추가.</li> <li>* replacement: 후속 세대 해의 수가 한 세대를 이루는 해의 수가 되면 통째로 교체.</li> </ul>
<p><b>[권수진]</b></p> <ul style="list-style-type: none"> <li>* Selection: 룰렛휠 방식 사용. 가중치 총합이 높을수록 선택될 확률 증가.</li> <li>* Crossover: 각 자리마다 6:4 비율로 우/열성 유전자 선택.</li> <li>* Mutation: 0.05 의 확률로 flip. 변이되는 자릿수는 노드 수에 비례.</li> <li>* Replacement: 한 세대가 생성되면 부모풀을 자식풀로 일괄 대체.</li> </ul>
<p><b>[류경빈]</b></p> <ul style="list-style-type: none"> <li>* selection: 서로 다른 서브그래프 사이의 가중치 합으로 cost 계산, Fitness 공식에 절대값 붙이고 정렬하여 룰렛휠 수행.</li> </ul>

- \* crossover: 랜덤 교차점에 따라 부모를 분할해 복제.
- \* mutation: 랜덤 변이점 지정해 확률에 따라 flip.
- \* replacement: Steady-state GA 사용. 최소 cost 유전자와 자식 유전자 cost 비교 및 교체.

#### [방기연]

- \* Selection: 토너먼트 방식, 우수한 해의 승률은 70%
- \* Crossover: One Point 방식, 교차점 전후로 두 부모의 유전자 잘라 붙임.
- \* Mutation: 자식의 각 노드 마다 0.01 의 확률로 flip.
- \* Replacement: generational GA 방식, 자식이 부모 세대를 통째로 대체.

### 3 문제 해결 시도

설계 및 구현한 팀원들의 GA 중에서 하나를 골라 문제 해결을 시도한다.

#### 3.1 최종 GA 선정

팀원 간의 논의로 최종 GA 는 [김영은]의 것으로 결정했다. 다음은 팀원별 GA 선택 사유이다.

[김영은] 성능이 안정적이고 오류 대비가 잘 되어 있었음

[가상은] 새로운 해가 생성될 때마다 실시하는 유효성 검사와 꼼꼼한 예외처리로 해의 다양성을 유지함과 동시에 해의 품질을 높였다.

[권수진] 유효성 검사와 치밀한 예외처리로 안정적인 알고리즘 수행이 보장된다. 해당 GA 의 replacement 방식이 다양성을 유지하며 점차 좋은 해로 수렴할 수 있는 조건이라 판단했다.

[류경빈] 해의 다양성을 유지하는 방식이 치밀해서 좋은 결과를 도출해낼 수 있는 알고리즘이라 생각하였다.

[방기연] 유일하게 해의 유효성 검사가 존재하고 꼼꼼하게 에러에 대비하였다. 해의 다양성과 성능의 안정성을 둘 다 놓치지 않았으며 또한 성능이 우수하다.

#### 3.2 Sample Test 결과

50, 100 개 노드를 갖는 그래프에 대해 2 ~ 5 내외의 표준편차로 안정적인 성능을 보이거나 500 개 노드를 갖는 그래프에 대해서는 표준편차가 80 이상으로 불안정한 성능을 보였다. 표준편차가 증가하는 이유는 찾아내지 못했으나, 단일 해에서의 돌연변이가 발생 횟수가 해의 길이에 비례하므로 해가 길면 발생 횟수가 증가하는 것이 원인일 것으로 짐작된다.

Table 6 Sample Test 통계

각 31 번 실행	최고 solution	최저 solution	평균	표준편차
unweighted 50	99	90	95.774194	2.704437
unweighted 100	354	337	344.903226	4.599673
weighted 500	3709	3440	3529.548387	84.007079

#### 3.3 Sample Test 결과 분석

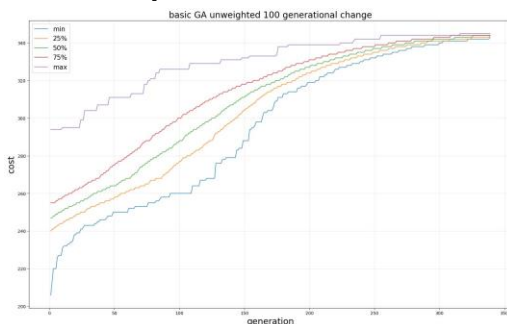


Figure 1 basic GA unweighted 100 generational change

[Figure 1]은 제출한 GA 의 테스트 run 중 unweighted 100 데이터의 31 번째 run 세대별 cost 변화이다. 가로축은 세대, 세로축은 보유한 해의 cost 이며 총 339 세대가 기록되었다. 초기 pool 의 cost 는 206 ~ 294 사이였고, 마지막엔 343 ~ 345 로 수렴했다. 그래프의 각 선은 아래서부터 세대별 min, 25%, 50%, 75%, max 값을 그린 것이다. 평균은 247.2715 에서 343.9210 으로 상승했다.

많은 구간에서 최댓값과 75% 값의 차이가 큰 것으로 보아 cost 가 큰 해는 소수라는 것을 알 수 있으며, 진화 초기에 빠르게 최솟값이 상승하는 점이 특징적이다.

## 4 DISCUSSION - 프로젝트 리뷰

팀원별로 프로젝트 진행 중 느낀 점, 잘 안 되는 점, 의외의 현상, 예상대로 된 점 등을 서술한다.

### 4.1 김영은

설계한 알고리즘을 코드로 썼을 때, 인덱스 오류 등 위치 파악이 어렵고 알고리즘의 논리와는 관계가 적은 버그가 발생하여 오류 대비 코드가 많아진 점, 해의 유효성을 검사하는 방식이 효율적이지 못해 노드 수가 많으면 시간이 오래 걸리는 점이 아쉽다. 마지막 제출 전 실행파일을 테스트해보다가 비주얼 스튜디오를 통해 실행할 때보다 실행 파일로 실행할 때가 훨씬 결과 도출 시간이 짧다는 것을 알게 되었다.

돌연변이 발생 알고리즘에 대해, 해의 길이와 무관하게 1 자리로 고정하면 길이가 길수록 돌연변이의 힘이 약해질 거라 판단해 해의 길이에 비례해 발생할 수 있도록 알고리즘을 작성했는데, 이로 인해 해의 길이가 길수록 결과의 편차가 증가할 수 있다는 것은 예상하지 못했다.

### 4.2 가상은

초반에 문제 이해와 세대나 해의 값을 어떠한 컨테이너에 담을지에 대한 전반적인 알고리즘 설계에 어려움이 있었다. 해의 다양성을 유지하는 것과 동시에 제한 시간 내에 가장 좋은 해를 도출해야 해서 이것들을 모두 고려하여 알고리즘 각 과정에서의 방식이나 조건, 확률 값 설정 또한 고민이 많았지만, 점점 해가 좋아지는 것을 볼 때마다 뿌듯함을 느꼈다.

### 4.3 권수진

고차원적인 인간 DNA에 비해 단순한 그래프에 GA를 활용하다 보니 구상했던 조건들을 사용할 수 없던 점이 아쉬웠지만, 알고리즘을 설계하고 값이 수렴해가는 과정을 확인할 수 있어 흥미로웠다. 급격하게 cost가 높게 진화하는 부분이 있어 놀랐고, 좋은 스키마를 지정해 그대로 물려주면 더 빠르게 수렴될 것이라 생각했다. 벡터 사용 관련 인덱스 오류 처리가 어려웠다.

### 4.4 류경빈

순수 GA 이다보니 빨리 값이 수렴이 안 되는 문제가 있었다. 따라서 Generation 의 stopping condition 지정이 어려웠다. 시간 단축이 힘들었다. 여러 자료구조를 사용해 보았고 알고리즘에 있어서 time complexity가 중요한 만큼 직접 알고리즘을 구성하는 경험은 뜻깊었던 것 같다.

### 4.5 방기연

자연적인 유전 현상을 알고리즘에 적용하는 것이 새로웠고 각 연산자 지정에 따라 성능에 큰 변화가 있다는 것이 신기했다. 자연어로 정리한 GA 구조를 코드로 구현하는 것과 갑작스러운 예외에 대한 처리와 디버깅 시간 조정에 있어서 어려움을 느꼈다. 사실 one point 연산자 사용에 있어서 성능의 향상이 별로 없을 거라고 생각했는데 그렇지 않은 점이 의외였다. cost를 구하는 함수가 예상대로 잘 작동해주어 좋았다.