

# 유전 알고리즘 기반 Max-cut 문제 해결

섬식 GA와 지역 최적화를 통한 해결방식 도입을 중심으로

20201015 컴퓨터공학과 최유림

## 1. 과제3 비교 수정사항 및 이유

### A. 코드 내 cost 용어를 위한 개념 재정립

기존의 코드를 작성할 때는 토너먼트 방식을 이용했기 때문에 sumWeight 값만 있으면 서로의 값을 비교할 수 있기 때문에 cost를 따로 계산하지 않았다. 그러나 해당 문제를 풀며 문제 공간 도시 해석을 위해 cost 값이 필요해졌기 때문에 cost값을 계산할 수 있도록 했다. Cost의 의미는 해를 구하기 위한 비용이므로 작을수록 좋은 해이다. 그러나 sumWeight는 클수록 좋은 해이므로 서로 반비례 관계를 가짐을 알 수 있다.

따라서 cost의 정의를 sumweight에 음수를 취한 값으로 정의하여 출력했다. 처음엔 1을 sumWeight로 나눈 값을 사용했으나 값이 너무 작아 그래프에 표현이 어려웠다. 이후 sumWeight에 log를 취한 값의 음수를 값으로 사용했으나 chimera 데이터의 경우에는 음수 값의 sumWeight를 가지는 경우가 있어 해당 계산 방식은 폐기했다.

### B. 섬식 GA로 수정

기존 코드는 500개의 해를 랜덤으로 발생시키고, 500개의 해를 generational GA를 통해서 해의 값을 발전시키는 방식을 사용했다. 그러나 이번 과제에서는 섬식 GA를 사용하였다. 섬 3개를 두고, 각 섬에서 100개의 해를 발생시킨다. 이후에 섬간 해를 공유하기 위해 모든 섬의 해를 하나의 섬으로 합쳐 300개의 해를 갖게 된다.

해당 방식의 문제는 자동으로 발생되던 좋은 해들에 대한 확률을 줄일 수 있다는 것이다. 한번에 500개의 해를 발생시키면 랜덤으로 좋은 gene을 가진 해가 발생할 확률이 더 높아진다. 그러나 각 섬마다 100개씩 발생되면 그 확률이 매우 줄어들 수 있다. 이후에 섬간 교류를 하더라도 기존의 확률로 발생된 해들이 합쳐지는 것이므로 좋은 gene의 발생이 어려워진다. 따라서 섬간 교류 방식을 도입했을 때 저번에 제출한 과제보다 더욱 좋지 않은 성능을 가지는 것을 알 수 있었다.

이에 대한 과정을 테스트하기 위해 각 섬마다 300개씩의 해를 가지고, 총 합쳐진 해의 개수를 900개로 조정했을 때 기존 과제의 성능처럼 나오는 것을 확인할 수 있었다. 그러나 해당 방식은 섬간 교류 방식의 장점과는 멀어지는 방식이기 때문에, 기존의 해 발생 개수를 늘리지 않고도 좋은 해를 출력할 수 있는 방식을 찾고자 했다.

### C. GA 내 중요 연산자

기존 코드로 실험했을 때, 시간 안에 원하는 sumWeight를 얻을 수 있었다. 따라서 연산자의 변화를 주지 않고 각 섬마다 그리고 마지막 교류 후 최종 섬에서의 GA에서도 동일한 연산자를 사용한다. 또한 섬식 GA와 local optimization을 추가한 경우의 성능을 비교하기 위해서 연산자에 변화를 주지 않았다.

따라서 selection에는 토너먼트 선택 방식을, crossover에는 Uniform Crossover를,

mutation은 non-uniform 방식, replace는 generational GA 방식을 그대로 사용했다.

## 2. 가장 좋은 GA 선택 과정 및 선택한 연산자

### A. 선택 과정

섬간 교류 과정의 결과에서 기존 일반 GA 방식보다 성능이 좋지 않은 것을 알 수 있었다. 그 이유는 다양성의 부족 문제로, 기존에 발생시키던 해의 수를 줄이게 되며 좋은 유전자의 발생 확률을 줄였기 때문이다. 따라서 local optimization을 도입해서 좋은 유전자를 강제로 도입하는 방식을 사용했다. 아래에서 선택한 local optimization의 과정을 설명하도록 하겠다. 섬간 교류 방식을 도입하는 장점은 자연 해 발생 시간을 줄일 수 있고 각 GA의 과정의 시간이 단축된다는 것이다. 이후 local optimization의 시간은 고려해야겠지만 GA 과정의 시간이 기존에는 매우 오래 걸린다는 것을 감안하면 큰 장점임을 알 수 있다. 그리고 최적화를 진행해보면 다양성에서의 장점 또한 파악할 수 있는데 3번의 그래프를 통해 설명하도록 하겠다.

### B. Local Optimization

Local Optimization의 적용 위치는 각 섬과 모든 섬의 해가 합쳐진 섬간 교류가 완료된 섬의 결과해이다.

Local Optimization을 적용한 최초의 해는 70번의 세대를 거친 GA 결과 해 중 가장 좋은 값을 선택한다. 이전에 작성한 과제에서 작성한 quick sort 방식을 이용해서 모든 결과의 해를 sorting하고 가장 첫 번째 해를 골라 local 최적화를 실행한다.

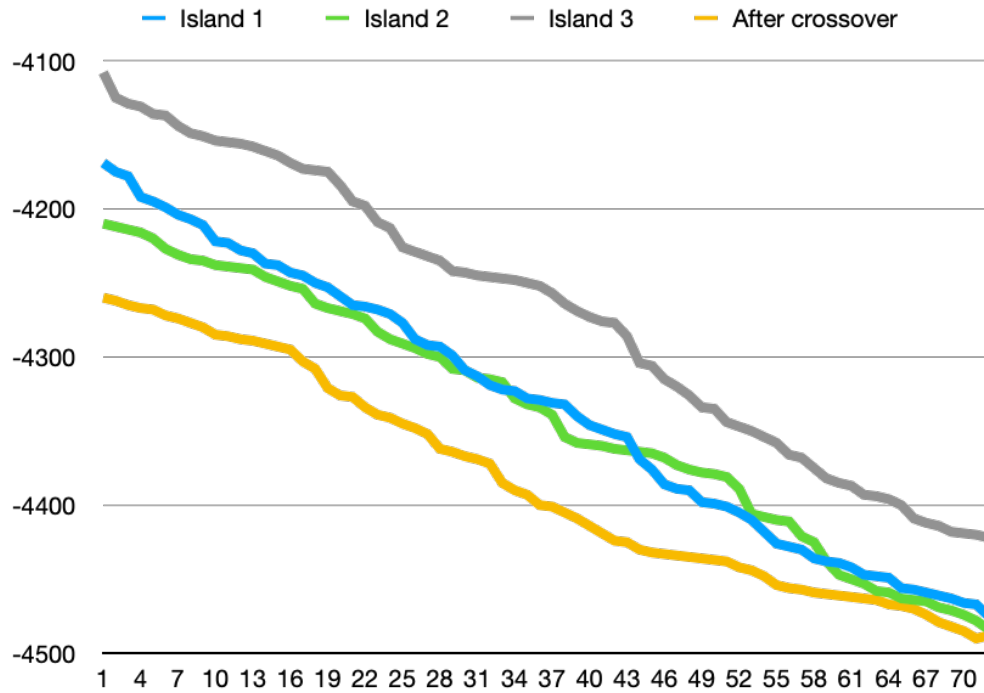
하나의 solution 해의 길이가 순열의 개수가 된다. 예를 들어 300의 길이를 가진 해는 총 300개의 경우가 생긴다. 첫 번째 gene부터 마지막의 gene까지 한 비트씩 바꾸기 때문이다. 기존 해에서 한 비트만 바뀌면 distance가 1이다. 로컬 최적화는 점점 좋은 해로 발전하기 때문에 한 비트를 바꿨을 때 기존의 해보다 좋은 sumWeight를 가지면 비트를 바꾼 해로 solution을 바꾼다. 바꾼 해의 sumWeight 값에서 기존 해의 sumWeight 값을 뺀 값이 0보다 크면 좋은 해로의 변화를 의미한다. 이후 발전한 solution을 기준으로 local optimization의 과정을 반복한다.

로컬 최적화를 진행하다 보면 distance가 중복되는 경우를 여러 개 가질 수 있다. 이를 정리하기 위해서 HashMap 데이터 구조를 도입했다. Hash Map에서 key는 distance가 되고 value는 해가 된다. Distance별 하나의 value를 가지는데, 이 value는 동일한 distance를 가지는 해 중 가장 sumWeight가 높은 값을 선택한다.

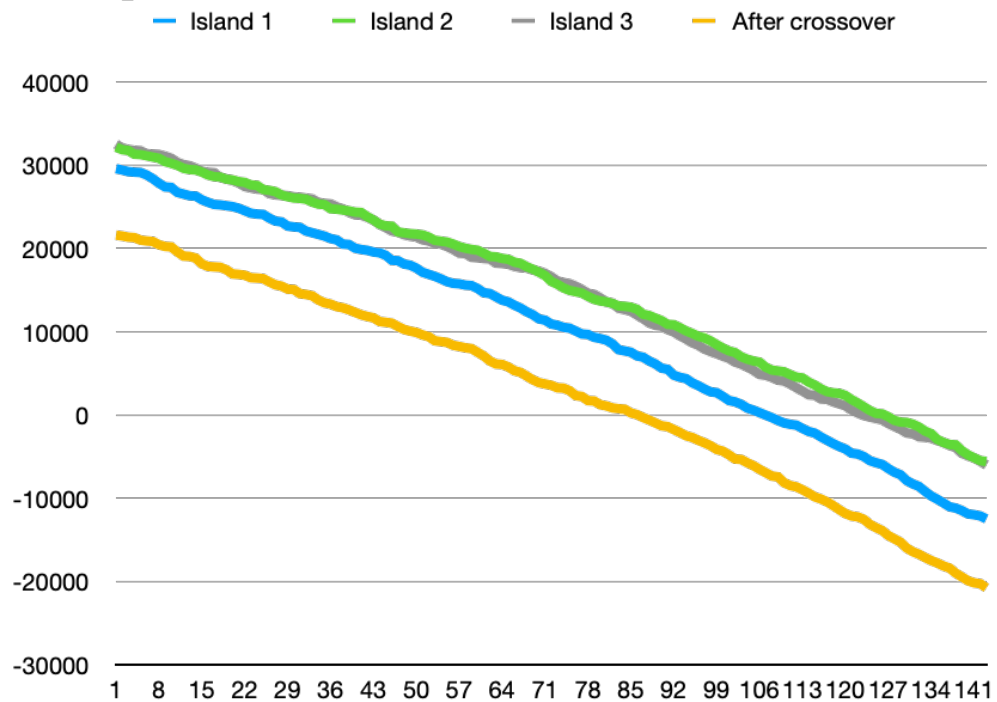
이후 distance 변화별로 sumWeight값의 변화를 보면 더 좋은 sumWeight를 가지는 것을 확인할 수 있다. 기존의 섬간 교류 방식만을 도입했을 때에는 모든 과정을 거쳐도 양수 값을 도출하지 못하는 것을 확인했는데 해당 local optimization을 통해서는 기존 섬에서도, 섬간 교류가 완료된 해에서도 양수 값의 sumWeight가 도출되는 것을 확인할 수 있다.

### 3. 문제 공간 도시와 해석

#### A. Weighted\_500을 이용한 문제 공간 도시 그래프



#### B. Chimerat\_946을 이용한 문제 공간 도시 그래프



#### C. 해석

각 섬별로 local optimization을 진행했을 때와 모든 섬을 교류한 결과를 기준으로 local optimization을 진행했을 때의 distance별 cost 변화이다. 그래프들을 모두 보면 local optimization을 진행할 때마다 cost가 낮아지는 것을 볼 수 있다. Cost는 낮아질

수록 좋은 해를 가짐을 의미하므로 local optimization이 좋은 해로의 결과를 이끌고 있다는 것을 확인할 수 있었다.

또한 해당 그래프를 보면 섬간 local optimization의 결과 사이에서도 큰 차이가 있음을 알 수 있다. 각 섬별로 local optimization이 진행되는 차이를 보면 특정 섬에서는 크게 좋은 해를 가진다. 이는 다양성을 가짐을 의미한다. 각자의 섬에서 독립적으로 해를 발전시키고 특히 좋은 해를 가지는 섬은 global optimization을 위한 다양성의 기반이 될 수 있다. 이후 섬간 교류가 발생하기 때문에 각자의 섬에서 발전한 최적의 해 또한 합쳐지고 합쳐진 해의 집합에서 다시 local optimization이 진행된다 보면 global optimization으로 갈 수 있는 확률이 늘어날 수 있다.

교류가 완료된 해들을 보면 기존 섬에서 발생한 해의 cost보다 훨씬 더 낮은 cost를 가지는 것을 확인할 수 있다. 이는 Global optimization으로 간다고 간주할 수 있다.

#### 4. Discussion

처음에 알고리즘 내에 섬간 교류 방식을 처음 도입했을 때는 성능을 보고 기존 방식보다 너무 좋지 않아 해당 방식의 도입 이유를 이해할 수 없었다. 특히 섬에서 발생하는 해가 적어 다양성을 보존하지 못하여 성능이 나빠지는 것을 이해하지 못했는데 이번 기회를 통해 직접 체감할 수 있었다. 섬에서는 100개의 해로 발생시켰을 때 그리고 총 해의 개수가 300개가 되는 교류 이후의 해 집합에서도 유전 알고리즘만으로는 chimera data의 양수 값을 얻어낼 수 없었다. 그러나 섬에서 300개의 해를 발생시키면 바로 양수 값을 얻어낼 수 있는 것을 확인한 뒤 유전 알고리즘에서 다양성 보존이 얼마나 중요한지를 체감할 수 있었다.

그리고 섬간 교류의 장점을 직접 그래프를 그려본 뒤 확인할 수 있었는데 각 섬에서 발생한 독립된 다양성이 생각보다 local optimization을 거친 뒤에는 큰 값의 차이를 보임을 알 수 있었다. 이는 결국 한 섬에서 독립적으로 발생한 해가 위치한 근처의 local optimization으로 이끈다고 이해했다. 이는 결국 여러 local optimization이 각 섬에서 발생하고, 이러한 섬이 여러 개 있으면 global optimization으로 갈 확률이 높아진다는 생각을 하게 되었다.

이러한 가정을 바탕으로 섬의 개수를 늘리면 기존에 발생하던 sumWeight 값보다 훨씬 큰 27000대의 sumWeight값이 더 자주 발생하는 것을 확인할 수 있었다. 특정 섬에서 발생한 다양성이 랜덤으로 발생했을 때의 경우지만, 발생하는 횟수를 확인했을 때 더 안정적으로 최적화되는 것을 확인할 수 있었다.

유전 알고리즘에는 정말 많은 연산자와 구조가 있는 것을 확인할 수 있었고, 이를 직접 적용해볼 때 결과값을 보고 어떤 연산과 구조가 좋은지 고민해보는 과정이 즐겁기도 했고 힘들기도 했다. 결과값은 단순히 연산자이나 구조의 옳고 그름이 아니라 테스트를 할 데이터에 따라서 어떤 방법을 적용하느냐에 따라 또 달라지는 것을 알 수 있었다. 작은 데이터의 경우에는 최적화를 돌렸을 때, 아무런 최적화가 일어나지 않는 것을 알 수 있었다. 이는 local optimization 없이도 유전 알고리즘만으로 최적의 해를 찾아낼 수 있음을 의미한다. 이렇듯 데이터에 따라 더 적합한 방식이 있음을 이해하고, 데이터에 따라 어떤 방식이 좋을지 고민하는 과정이 필수라는 것을 알 수 있게 되었다.