

MAX-CUT PROBLEM WITH PURE GENETIC ALGORITHM

정시공

2020-1028

2020-1039

2020-1043

2020-1047

김서진

성유진

심수림

유정선

1. 사용한 GA 구조

김서진	먼저 주어진 정점, 간선, 가중치 정보를 바탕으로 초기화를 진행했다. 정점을 임의로 선별하고, 정점별로 정점과 관계된 간선의 가중치 정보를 업데이트했다. 이를 바탕으로 선별된 정점 중 가중합이 작은 정점들을 교차했고, 돌연변이를 거쳐 최적해를 도출해내는 구조를 사용했다.
성유진	노드와 가중치를 모두 초기화하고, 데이터를 랜덤으로 골라 토너먼트를 하는 방식으로 선별과 교차를 하였다. 또한 임의로 돌연변이가 발생할 확률을 지정해서 돌연변이가 발생하였다면 그것을 랜덤하게 고른 데이터와 대치를 하는 GA 구조를 구성했다.
심수림	초기 해를 랜덤으로 선정하여 정점들을 둘로 나누어 0 또는 1로 표현하였다. 정점의 두 집합을 잇는 간선의 적합도(가중치 합)을 구하는 함수를 이용하여 적합도가 높으면 개체가 업데이트 된다. 토너먼트 방식으로 부모 개체를 선택하고 교차와 돌연변이 연산을 통해 최적해를 찾는 GA 구조를 사용하였다.
유정선	벡터를 만들어 인덱스마다 랜덤하게 0과 1을 주어 정점의 집합을 S와 S'으로 분류하였다. 랜덤하게 2개의 정점을 고르고 교차 연산을 수행하였다. 함수를 이용하여 두 정점 간의 가중치를 계산하고 토너먼트 방식을 사용하여 새로운 자식 개체를 만든다. 그리고 일정한 확률로 돌연변이 연산을 수행하여 변이된 개체를 생성하도록 하였다.

2. 해의 표현

김서진	개체는 'SELECT_V' 배열을 통해 나타낸다. 배열에 포함된 원소들이 선별된 정점들이고, 배열에 포함되지 않은 정점들과 구분된다. 이를 바탕으로 정점별 가중합, 선별된 정점들의 가중합을 계산하며 세대를 거듭한다.
성유진	개체는 POPULATION 벡터를 사용하여 나타냈다. POPULATION 개체 있는 정점들은 토너먼트를 통해 선별된 정점이며, 선택되지 않은 정점들과 구분하였다. 이를 통해 가중치를 계산하였고 POPULATION 개체에서 최적의 집합을 나타내는 벡터인 BEST 벡터를 만들어 최종 결과를 도출하였다.
심수림	GA의 각 개체를 'INDIVIDUAL' 구조체로 표현하여 구조체 내의 'CHROMOSOME' 배열로 해를 표현한다. 배열의 각 원소는 그래프의 정점을 나타내며, 두 집합으로 나누기 위해 0 또는 1의 값을 가진다. 1은 정점이 선택된 것을 의미하며 0은 선택되지 않은 것을 의미한다. 이진 문자열로 표현된 CHROMOSOME 배열은 선택된 정점들의 집합을 나타내고 이를 바탕으로 가중치 합을 계산한다.
유정선	개체는 PARTITION 벡터를 통해 표현했다. 벡터 PARTITION 원소에 0, 1 부여하여 정점들을 2개의 그룹으로 무작위 분할한다. 각 인덱스마다 0, 1의 값을 가지면 0의 값을 가진 정점들은 S 집합, 1의 값을 가진 정점들은 S'의 집합으로 분할된다. 벡터를 통해 집합을 분할한 후에 가중치 합을 계산하며 진행한다.

3. 사용한 연산자에 대한 설명

	SELECTION	CROSSOVER	MUTATION	REPLACEMENT
김서진	주어진 정점, 간선, 가중치 정보를 바탕으로 정점을 임의로 선발한다.	정점별로 관계된 간선의 가중치 정보를 초기화후, 이를 바탕으로 선발된 정점 중 가중합이 작은 정점들을 교차한다.	발생한 난수의 값이 임의로 정한 확률보다 작은 경우 돌연변이를 발생시킨다.	SELECTION, CROSSOVER, MUTATION 을 거쳐 최종적으로 정해진 최적해를 선발한다.
성유진	랜덤하게 데이터를 선발을 해서 토너먼트를 시행한다.	토너먼트를 시행해서 선발된 두 데이터가 있다. 토너먼트를 시켜 이긴 데이터들끼리 교차를 시킨다.	돌연변이를 만드는 확률을 임의로 정한다. 0.0 에서 1.0 사이의 랜덤 수가 미리 정한 임의의 돌연변이 확률보다 작다면 돌연변이가 발생한다.	돌연변이가 발생하였다면 돌연변이를 토너먼트 선발을 통해 뽑은 것들 중 하나랑 대치를 한다.
심수림	랜덤 함수를 통해 정점을 토너먼트 방식으로 선택하고 임의로 두 집합으로 나눈다.	랜덤하게 선택한 교차점을 기준으로 부모로부터 각각 정점 정보를 받아와 교차한다.	랜덤 수가 임의로 정한 확률 보다 작으면 개체의 비트를 반전시키는 돌연변이가 발생한다.	개체 선택, 교차, 돌연변이 함수를 수행하고 새로 생긴 개체 집단이 기존 개체보다 우수하면 최적해를 업데이트한다.
유정선	랜덤하게 인덱스를 선택하고 해당 인덱스에 해당하는 개체를 선택한다.	랜덤하게 두 개체를 선택하고 선택된 두개의 개체 중에서 적합도 높은 개체를 선택한다.	돌연변이는 0 과 1 사이의 난수를 생성하고 이 값에 따라서 발생한다. 이 값이 정해 놓은 MUTATIONRATE 보다 작으면 돌연변이를 적용한다.	서로 다른 두 개체를 교차해 새로운 개체를 생성하고 그 적합도를 계산하여 기존의 개체와 바꾼다. 돌연변이 적용시 그 개체의 적합도를 계산하여 기존의 개체와 바꾼다.

4. 함께 제공하는 세 개의 샘플 인스턴스 인스턴스에 대해 GA 를 각각 최소 30 번씩 수행하여 가장 좋은 결과, 평균 결과, 표준편차를 테이블로 기록

	가장 좋은 결과	평균 결과	표준편차
Unweighted_50	83	78.9	1.647
Unweighted_100	293	283.5	3.711
Weighted_500	3480	3389.9	24.41

5. 1~3 을 거치며 팀원의 의견을 하나로 모은 뒤, 최고의 GA 에 대한 팀 의견

최고의 GA : 주어진 입력 파일을 바탕으로 정점을 임의로 선발하고, 토너먼트를 통해 1 차적인 최적해를 선발한다. 개체들 중 임의로 두 개를 선발해 crossover 를 진행, 임의의 돌연변이 확률을 바탕으로 mutation 을 시행하며 개체 집단을 초기화 한다. 이들을 1 차 최적해와 비교해 최종적인 최적해를 선발한다. 해당 알고리즘이 max-cut 문제를 가장 잘 해결할 수 있는 ga 다. 이러한 기본 알고리즘을 바탕으로 개체 크기, 교차 방식 조정, 돌연변이 확률 조정 등의 매개변수 조정을 통해 최종적으로 알고리즘을 완성했다

6. 위에서 실행한 GA run 들 중 하나를 선택, 세대 진행에 따른 population 분석

initializePopulation 함수를 통해 1500 개의 개체 집단을 생성, 초기화 한다. 각 개체의 적합도를 계산하고 현재 개체가 최적해 보다 우수하면 최적해를 업데이트한다. 새로운 개체 집단을 생성한 후 토너먼트 선택을 통해 1 차적으로 개체를 선발한다. 이후 부모를 랜덤 선택해 crossover 를 진행하고, mutation 을 통해 생성된 자식 개체를 새 개체 집단에 업데이트한다. 위 과정을 while 문을 통해 제한 시간 동안 세대를 거듭하며 반복하며 population 이 변화된다.

7. Discussion

	느낀점	잘 안되는 점	의외의 현상	예상대로 된 점
김서진	이제껏 주어진 문제를 풀어내는 방식 위주의 학습만을 진행해와서 시간 내에 최적해를 도출하는 이번 과제가 낯설면서도 흥미로웠다. 같은 데이터를 바탕으로 하지만 코드에 따라 천차만별의 성능을 낸다는 것이 신기하다.	선발된 정점 중 관계된 가중합이 작은 정점들을 교차하면 성능 개선이 이루어질 것이라는 아이디어와 달리 성능 업데이트가 잘 되지 않았다. 오히려 가중합이 줄어들기도 했다.	기존의 코드에서 개체 집단의 크기, 교차 진행방식, 돌연변이 발생 확률 등 여러 요인들을 다양하고, 큰 폭으로 조정해봤음에도 불구하고 가중합에 드라마틱한 변화가 일어나지 않았다.	교차와 돌연변이가 초기엔 성능 개선에 큰 도움이 되지 않지만 결과적으로 다양성을 확보해 최적해로 수렴하게 해준다는 설명을 바탕으로 코드를 수정했고, 잘 진행되었다.
성유진	수업에서 구체적인 코드가 아니라 간단하게 말로 표현한 코드만 배우고 ga 에 대한 이론만 알고 있는 상황에서 maxcut 을 하는 ga 코드를 짜는 것이 어려웠다. 하지만 처음부터 하나하나 찾아보며 코드를 짜니 더 ga 에 대해 이해를 더 잘 할 수 있게 되었다.	혼자 ga 코드를 짤 때, 타임 에러가 나서 결과가 나오지 않았다. 팀원들의 도움을 받아 타임 에러를 해결할 수 있었다. 또한 처음에 코드의 방향을 잡는 것에 어려움을 겪었다.	팀원들 모두 돌연변이 확률을 살짝 다르게 설정하였음에도 불구하고 비슷하게 결과가 나와 의외였다.	입력 파일을 불러 파일의 데이터를 읽고, 출력 파일을 생성하여, max-cut ga 코드를 시행한 결과를 출력 파일에 정상적으로 출력하였다.
심수림	과제 수행을 위해 GA 에 대해 공부해보니 유전 알고리즘이 복잡한 문제 해결에 도움이 된다는 것을 알 수 있었으며, 과제를 통해 알고리즘을 직접 적용해보는 과정이 어렵기도 했지만 색다른 경험이었다.	초반에 어떤 방법으로 데이터를 선택하여 교차해야 할지 구상하는 과정에서 어려움을 겪었다. 다양한 방법을 시도해보았지만 자잘한 오류가 생겨 오류가 생기지 않던 토너먼트 방식을 선택하게 되었다	팀원들의 GA 코드가 각각 다름에도 불구하고 결과값의 치역이 서로 비슷하게 나온 점이 의외였다.	알고리즘이 초기 해의 적합도를 계산하고 그 적합도를 기준으로 개체를 선택하여 업데이트를 통해 점점 나은 최적해를 만들어나갈 것이라고 생각했는데 예상대로 잘 진행되었다.
유정선	수업 시간에 GA 알고리즘에 대해 설명을 들었을 때 개념과 과정이 재밌게 느껴졌다. 하지만 막상 직접 코드를 직접 짜고 최적해를 찾아보려고 하니	처음에 코드를 어떻게 짜야할 지 감이 잡히지 않아 시작하는게 어려웠다. 내가 진행할 교차 방법을 선택해서 그 방법에 맞게 조금씩 코드를 짜보니 어떻게 진행해야할 지 감이 잡혔다.	팀원들이 각자 코드를 짜서 GA 알고리즘을 실행하여 최적화 찾는 과정을 진행하였지만 특정 영역에 비슷한 값이 집중되는 현상이 발견되었던 것 같다.	세대가 진행되면서 점점 최적해의 가까워지고 성능이 향상되는 부분이 예상대로 진행되었다.

	<p>생각한 것과 다르게 어떻게 시작해야 할지 몰라 당황했던 것 같다.</p> <p>GA 알고리즘에 대해 공부하면서 차근차근 알고리즘을 짜내려 가보니 알고리즘의 흐름에 대해서 이해할 수 있었다.</p>	<p>또한 예외처리 문제가 발생하여 결과가 출력되지 않는 문제가 발생하여 진행되는 과정에서 어려움을 겪었다.</p>		
--	--	--	--	--

8. 9 강 강의 중 생각해 볼 내용에 대한 팀원 별 답변 정리 표

김서진	9 주차 강의에서 GA 에 대한 전반적인 설명을 들었고, 기본적인 틀에 대한 이해는 잘 되었다. 하지만 심화 내용에 대한 이해가 많이 부족하고, 룰렛 휠과 토너먼트 선발 외 다른 개념들에 대해서 공부할 필요성이 느껴진다.
성유진	9 주차 강의 중 다양한 GA 선발과 교차 방법을 배웠다. 그 중 LOCAL OPTIMUM 을 사용하는 방법들에 대해 다음 과제를 위해 다시 한번 복습하고 생각해봐야 할 것 같다.
심수림	9 강 강의에서 GA 의 개념과 여러 방법에 대해 공부하였다. GA 의 기본적인 컨셉은 개략적으로 이해하였지만 강의 후반의 LOCAL OPTIMUM 이나 다른 방법론에 대한 이해가 많이 부족하여 다음 과제를 수행하기 위해 복습이 필요하다고 느꼈다.
유정선	9 주차에는 GA 의 개념에 대해서 배웠고 마지막에 GA 알고리즘을 적용할 수 있는 여러가지 응용 분야에 대해서 알려주셨다. GA 알고리즘을 적용할 수 있는 분야에 대해서 좀 더 찾아보고 어떻게 적용해 보는지 공부해보면 좋을 것 같다.