

## <Genetic algorithm의 correctness와 efficiency>

### **Correctness**

- 유전자 알고리즘은 초기 집단 생성 단계에서 그래프의 정점을 임의로 선택하여 모든 가능한 해가 고르게 선택될 수 있도록 초기 개체를 선택한다. 이후 선택된 개체의 적합도를 평가한다. 각 집단의 가중치를 평가하여 선택된 집단에 대한 정확한 평가를 한다. 선택, 교차, 돌연변이, 대체 연산자를 사용하여 유전자 알고리즘의 기본 원칙을 수행하는데 이때 모든 연산자는 최적 해를 찾는 과정을 유지하며 적합도 기준에 따라 해의 공간을 탐색한다. 따라서 유전자 알고리즘은 정확하게 동작한다.

### **Efficiency**

- 초기 인구 생성 함수는 각 정점 집합을 무작위로 선택하는 과정이며 해당 함수의 시간복잡도는  $O(n \log n)$ 이다. ( $n$ 은 정점의 수) crossover함수의 시간복잡도는  $O(n)$ 이다. Mutate함수의 시간복잡도는 개체 내의 한 정점을 추가하거나 제거하는 데 필요한 시간이므로 상수이고 selection함수의 시간복잡도는  $O(n)$ 이다. 마지막으로 genetic\_algorithm함수는 세대별로 기능을 반복하므로 시간 복잡도는 세대수( $g$ ), 개체수( $n$ ), 유전 연산자의 수( $m$ )에 따라 결정된다. 따라서 해당 유전자 알고리즘의 전체 시간 복잡도는  $O(g * n * m)$ 이다.

## <해의 표현(chromosome design)>

해당 유전자 알고리즘에서 각 개체(해)는 그래프의 정점을 나타낸다. 해는 선택된 최종적으로 선택된 정점들의 리스트로 구성된다. 해는 각 정점들의 번호를 표시하는 형식으로 나타내어지며 이 리스트는 해당 그래프에서 선택된 최적의 해를 말한다.

## <사용한 연산자에 대한 설명(selection, crossover, mutation, replacement)>

1. Selection : 'roulette\_wheel\_selection'함수가 선택 연산을 담당한다. 해당 함수는 유전자 알고리즘의 다음 세대를 형성하기 위해 부모 개체를 선택하는 과정에 사용된다. 각 부모 개체의 선택 확률은 해당 개체의 적합도에 비례하며 선택된 부모는 교차 및 돌연변이 연산에 사용된다. 높은 적합도를 가진 개체는 다음 세대에 자신의 유전자를 전달할 확률이 더 높아진다. 전체 개체의 적합도 합을 계산한 후 각 개체의 선택 확률을 그 개체의 적합도가 전체 적합도 합에 차지하는 비율로 설정한다. 이후 이 확률 분포에 따라 무작위로 부모를 선택한다.
2. Crossover : 'crossover'함수가 교차 연산을 수행한다. 해당 함수는 선택된 부모 개체 사이에서 교차를 수행하여 자손 개체를 생성한다. 두 개의 부모 개체를 선택하고, 두 부모 개체 사이에서 교차 지점을 선택하여 교차를 수행한다. 두 부모 개체에서 임의의 절단점을 선택한 후 이 점을 기준으로 유전자의 일부를 교환해 두 개의 자손을 생성한다.
3. Mutation : 'mutate'함수는 돌연변이 연산을 담당한다. 개체의 유전자 중 일부를 무작위로 변형시킨다. 돌연변이 확률에 따라 정점을 추가하거나 제거하는데 추가하는 경우엔 현재 개체에 존재하지 않는 정점 중 하나를 선택하여 추가하고 제거하는 경우에는 현재 개체

의 정점 중 하나를 무작위로 선택해 제거한다.

4. Replacement : Elitism방식을 사용하여 각 세대의 가장 우수한 개체를 다음 세대로 직접 전달한다. 각 세대를 평가한 후 가장 적합도가 높은 상위 개체를 다음 세대의 일부로 대체한다. 새로 생성된 자손 개체는 이전 세대의 부모 개체들과 함께 다음 세대의 개체 집단을 형성한다.

#### <세 개의 샘플 인스턴스에 대한 GA수행 결과>

	50	100	500
좋은 결과(최댓값)	90	291	3325
평균 결과	80.1	280.6	3394
표준 편차	3.8	5.3	22

#### <GA run – 50의 세대 진행에 따른 population 분석>

해들의 평균 품질	최고 품질	표준 편차
79.4	90	2.27

GA은 초기 population에서 시작하여 selection, crossover, mutation, replace 과정을 통해 새로운 population을 생성한다. 이 과정에서 세대를 거듭할수록 해의 품질이 점진적으로 개선되게 된다. 엘리티즘을 적용하여 각 세대에서 가장 좋은 해들이 다음 세대로 직접 전달되어 알고리즘이 최적 해에서 멀어지는 것을 방지한다. 30번의 실행을 거친 결과 평균 최적해의 가중치 합은 79.34으로 세대가 진행됨에 따라 평균 해의 품질이 점진적으로 향상됨을 확인할 수 있다. 이는 알고리즘의 유전적 연산 과정이 효과적으로 좋은 유전자 조합을 발굴하고 개선하고 있음을 나타낸다. 최고 최적해의 가중치 합은 90으로 최적에 가까운 해를 발견해내고 있다. 이 해당 GA가 우수한 해를 탐색하고 이를 유지 및 발전시키고 있음을 나타낸다. 위의 결과는 평균 가중치와 최고 가중치 사이의 일정한 차이가 존재하고 이는 실행마다의 결과에도 변동성이 유지되고 있음을 나타낸다. Population 내에서 해의 다양성이 유지되고 있음을 알 수 있다. 다양성을 통해 해당 GA는 local optima에 빠지지 않고 더 넓은 해공간을 탐색할 수 있도록 해주며 이를 통해서 알고리즘은 다양한 가능성을 탐색하며 global optima에 접근할 수 있다.

## **<Discussion>**

수업 시간에 강의를 들었을 때는 유전자 알고리즘에 대해 정확하게 이해했다고 생각했다. 그러나 생각했던 것보다도 더 직접 구현하는 일에 있어 어려움을 많이 느꼈다. 제일 처음에 작성했던 코드로는 선택된 개체들의 가중치 합이 70초반대로 나와서 더 좋은 결과를 도출해내기 위해 mutation 확률을 0.1에서 0.05로 낮추었다. 이전보다 더 좋은 결과를 반환했으나 생각했던 만큼의 효과가 있진 않았다. 이후 population을 늘려도 보았으나 여전히 비슷한 결과를 반환했다. 다른 자료들을 참고하다 crossover 함수를 정점 1개만 교차하는 것이 아니라 2개를 교차하는 것으로 수정하고 엘리트주의 방식을 사용했다. 이후 weighted\_50파일을 기준으로 30번의 run을 실행하면 평균적으로 70중반이 나오던 알고리즘이 수정 후에 80대로 향상되었다. 예상보다 더 과감하게 교체를 하는 자세가 더 좋은 결과를 가져왔음을 알 수 있었다. 생각보다 지정된 변수들이 끼치는 영향보다 연산자의 구현이 끼치는 영향이 더 큼을 알게 되었다.