3-1 8-queen 문제

**염색체 구성 \_\_init\_\_**

텍스트이(가) 표시된 사진

자동 생성된 설명

방법 1 : 1 ~QUEEN의 숫자에서 랜덤으로 숫자를 골라 퀸의 수만큼의 정수배열을 만들어 준다면 행(또는 열)의 QUEEN끼리의 충돌을 제거하면서 초기화가 가능하다 .

방법 2 : 또한 같은 초기화에서 같은 index의 숫자가 들어가는 것을 막게 된다며 행열로 겹치는 것을 방지하면서 초기화가 가능하다.

이렇게 된다면 초반에 적합도가 높은 유전자를 생성 가능하므로 값을 찾는게 빨라진다.

**적합도 계산 함수**

텍스트이(가) 표시된 사진

자동 생성된 설명

방법 1 : n개의 queen이 있을 때 모두 충돌하는 경우는 n+n-1 +…1이다 . 즉 (N\*N-1)/2 를 max\_value로 지정

방법 2 : 정수배열로 만들 시 겹치는 경우가 (1,2) (2,1)이렇게 2개씩 나오는 것을 방지하기 위해 set으로 제거 후 마지막에 max\_value에서 겹치는 경우의 수를 뺀 것이 유전자의 적합도로 정함.

따라서 적합도 함수는 (N\*N-1)/2 – h 이다.

**Crossover 함수**

텍스트이(가) 표시된 사진

자동 생성된 설명

방법 1 : crossover시에 겹치는 문제를 방지하기 위해 2 point crossover를 실행

서로 부모의 유전자 배열을 가지고 crossover을 하면서 나머지 것들의 유전자는 부모 유전자에서

유전된 것을 뺀 나머지 순서 그대로 들어가는 방식이다.

방법 2 : 방법 1을 수행할 시에 index1 과 index2가 같거나 차이가 1이 나는 경우에 다시 index를 설정하게 정함

**Mutation 함수**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

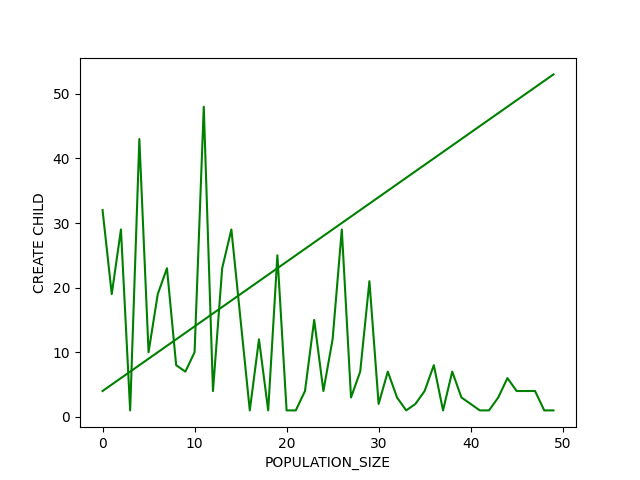
방법 1 : mutate시에 유전자의 각 요소마다 확률 이하의 값을 갖고 있다면 변형시키도록 정함

방법 2 : mutate는 유전자에서 2개의 index를 정해 swap을 해주는 방식으로 진행했다.

이때 index가 같은 경우 변형이 나지 않으므로 index가 다를 때까지 반복한다.

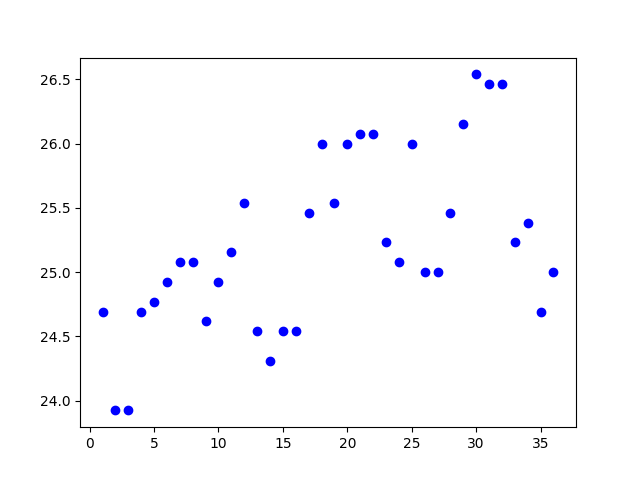
그후에 변형이 끝나면 유전자의 적합도를 다시 계산하도록 진행했다.

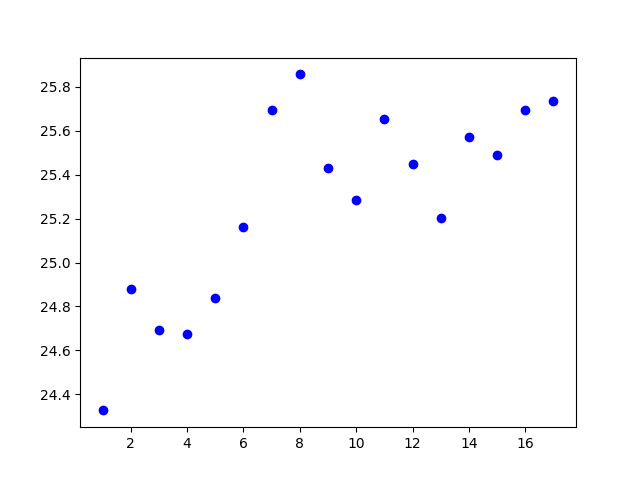
**8-queen의 세대 수 마다 유전자의 생성함의 차이**



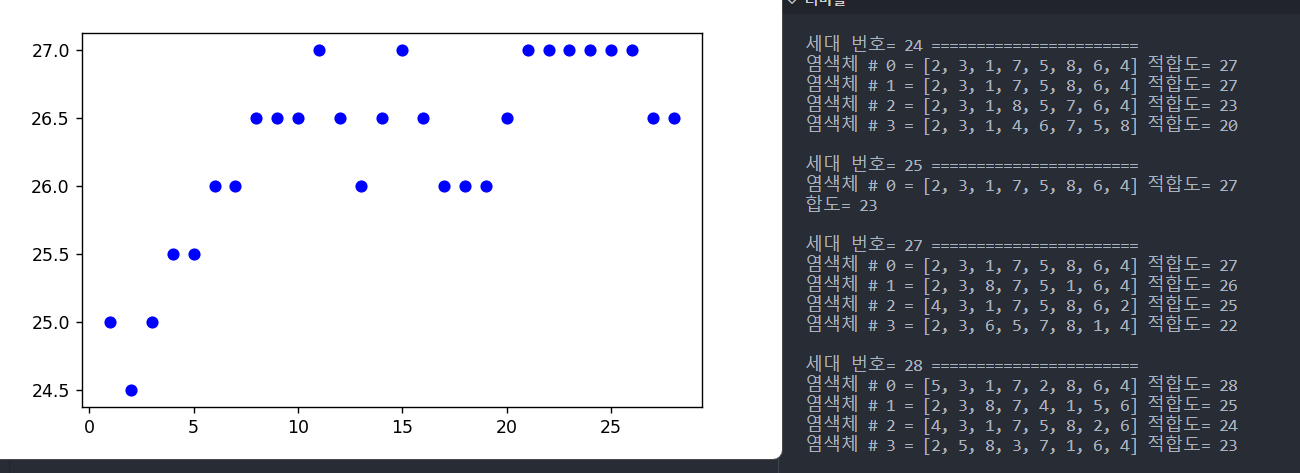
확실히 8-queen은 Population size가 커질수록 분열하는 세대수가 점점 줄어드는 모습을 보인다. 이는 즉 세대수세 설정할 때 분열되는 자식의 숫자 또한 더 많고 더 높은 적합도를 가질 확률이 높기 때문이라고 생각한다.

그리고 또한 유전 알고리즘에 있어서 세대가 분열될수록 적합도를 본다면 아래 그림과 같이 모호하게 나올수도 있고





위 그림처럼 증가하는 경향을 보일수도 있다. 위 그림들은 분열할 때 적합도가 증가하는 경향으로 나타난 scatter 그래프이다.



Ex) 세대수 4로 적용했을 경우의 적합도 추이 그래프와 수행결과 화면 캡쳐

**3-2 TSP 문제**

**염색체 구성 \_\_init\_\_**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

방법 1: 맨처음 csv에서 data를 가져와 city\_list를 만들어준다. 이때 서울은 처음과 마지막에 와야하므로 서울을 뺀 나머지의 city들을 가져온다.

방법 2: 이때 city배열에서 셔플을 통해 유전자를 생성한다.

**적합도 계산 함수**

텍스트이(가) 표시된 사진

자동 생성된 설명

방법 1 : 유전자에는 서울을 뺀 나머지 도시들만 존재한다 . 따라서 적합도를 서울부터 모든 도시를 거쳐 다시 서울로 돌아오는 것을 적합도로 설정하고 이때 적합도 함수는 이 거리를 계산한다. 그리고 모든 도시를 거쳤을 때 제일 작은 길이는 단순히 아래 그림과 같이 완전탐색으로 검색하고 이후 이 거리가 나오면 분열을 멈추도록 했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**Crossover 함수**

**위** 8-queen문제와 같이 중복을 제거하기 위해 2point crossover를 사용했다. 똑같이 crossover의 point들이 같거나 1차이가 난다면 계속 다른 값을 찾도록 지정했다.

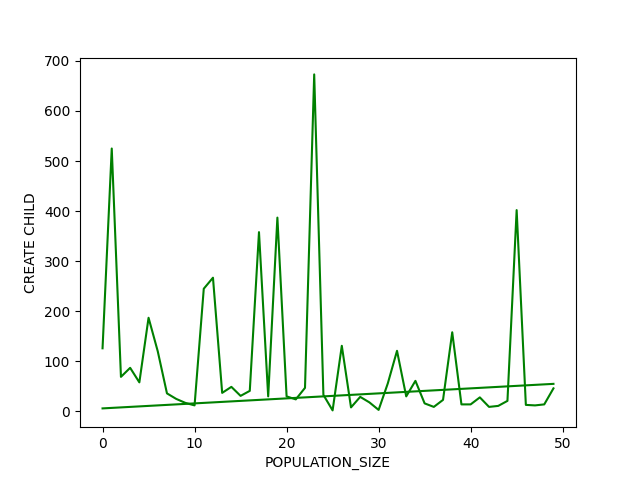
**Mutation 함수**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

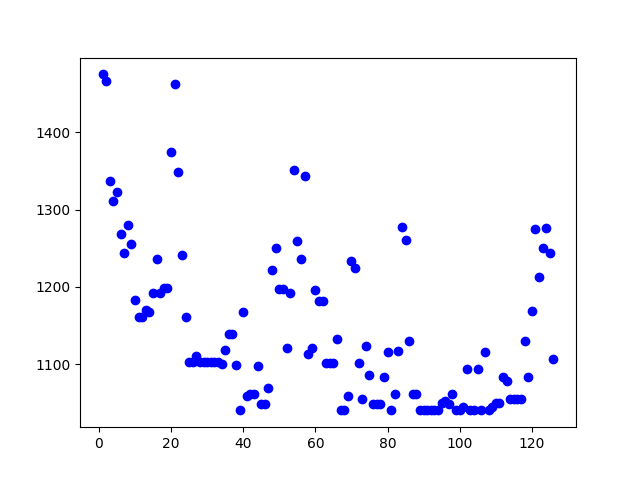
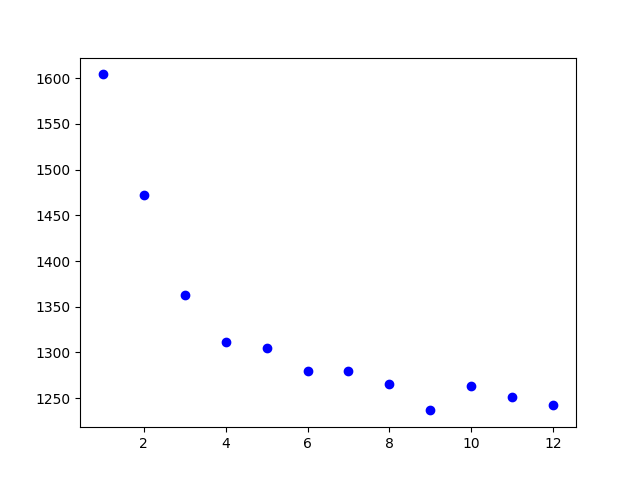
8-queen과 같이 random으로 rate보다 확률이 낮다면 다른 index와 변형해주도록 지정했다.

**TSP의 세대 수 마다 유전자의 생성함의 차이**

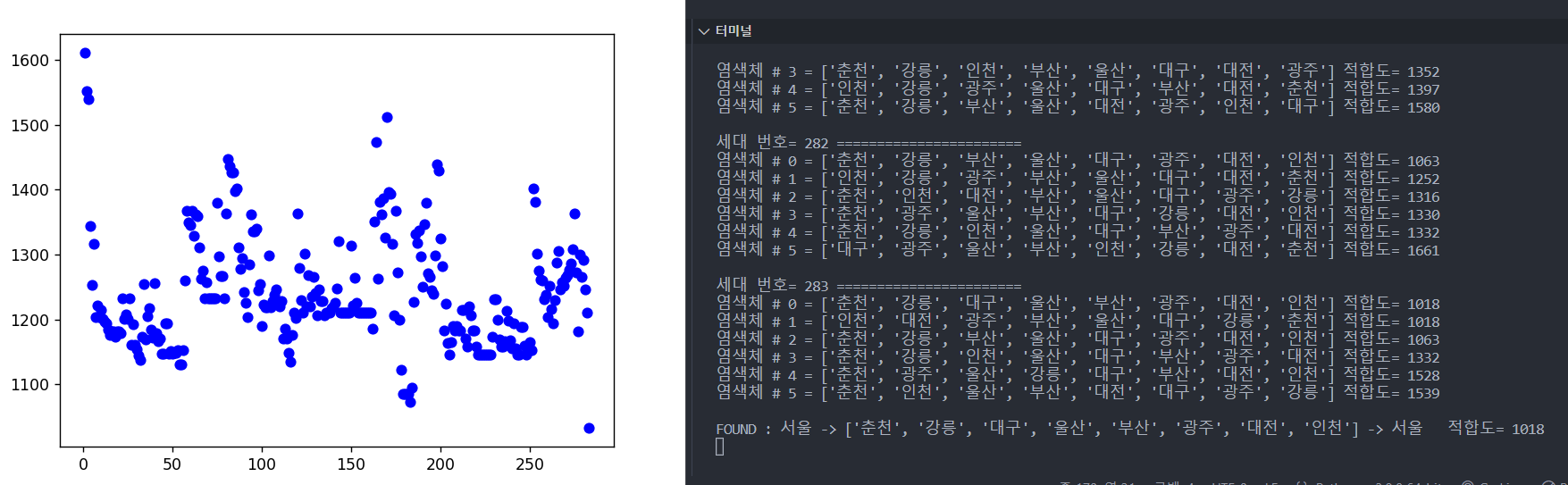


8-queen보다 극명하게 효과가 드러나지는 않는다. TSP는 경우의 수가 많게 초기조건을 설정했기 때문에 계속해서 유전자 분열이 일어나도 값을 찾는데 더 많은 분열이 필요 할 수도 있다는 것을 뜻하는 것이라고 생각한다.

이렇게 분열횟수로만 보면 적합도가 어떻게 달라지는 지 알수가 없기 때문에 적합도를 보자.



위 그림은 세대수가 16개일 때 6개일 때 분열하는 횟수를 보여준다. 확실히 세대수가 적은 유전자일수록 더 많이 분열하는 경향이 있는 것과 적합도가 점점 목표에 맞게 가까워져 가는 것을 볼 수 있다.



Ex\_ 세대수 : 6으로 했을 경우 유전자 적합도의 추이 그래프