

유전알고리즘을 이용한 N-queen problem 해결

정보컴퓨터공학부 201524530 이승준

1. 실험 목표

기존에 존재하는 알고리즘 외에 유전 알고리즘을 이용하여 N-queen problem의 optimal solution을 추출한다.

이 보고서에서는 N의 값에 5를 대입하여 5-queen problem을 해결했다.

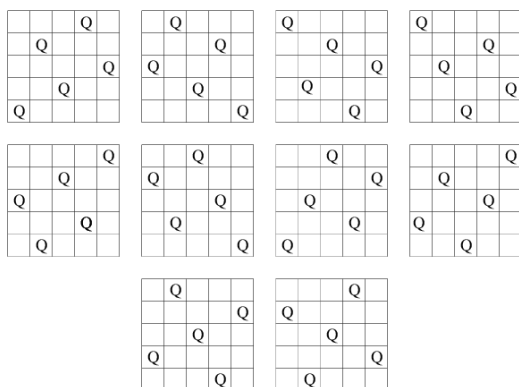
2. 배경 지식

유전 알고리즘(Genetic Algorithm):

자연세계의 진화과정에 기초한 계산 모델로서 **전역 최적화** 기법으로, **최적화 문제**를 해결하는 기법의 하나이다. **생물의 진화**를 모방한 **진화 연산**의 대표적인 기법으로, 실제 진화의 과정에서 많은 부분을 차용(채용)하였으며, 변이(**돌연변이**), 교배 연산 등이 존재한다. 또한 **세대**, **인구** 등의 용어도 문제 풀이 과정에서 사용된다. 일반적으로 1) Chromosome design, 2) Initialization, 3) Fitness evaluation, 4) Selection, 5) Crossover, 6) Mutation, 7) Update generation, 8) Go back to 3) 으로 총 8가지 단계를 실행한다.

5-queen Problem :

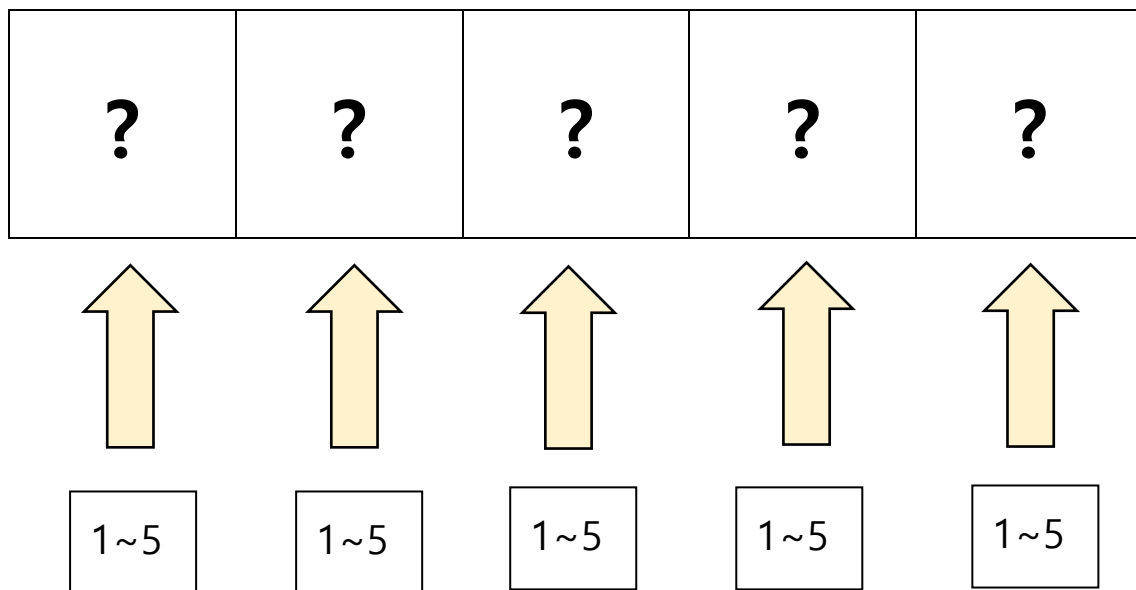
5x5크기의 체스판에 **퀸**을 5개 배치하는 문제이다. 이 문제를 일반화하면 NxN 크기의 체스판에 **퀸**을 N개 배치하는 N 퀸 문제가 된다.



3. 과정

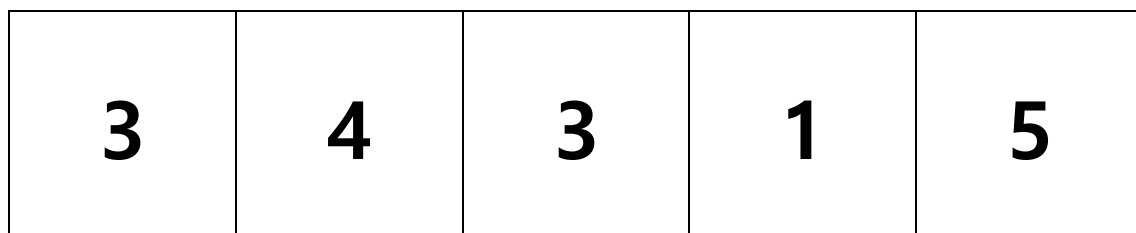
1) Chromosome design

Chromosome을 설계하는 단계. 5-queen problem의 경우 체스를 5개 놓는 경우이므로 일차원배열로 그림1-1과 같이 나타낼 수 있다.



<그림1-1. Chromosome design>

배열의 인덱스는 체스판 가로 열, 각 배열의 값은 퀸이 놓여진 세로 열을 의미한다. 체스가 세로 열에는 중복 돼서 는 안되지만 가로 열에는 1~5개의 체스가 랜덤 하게 올 수 있으므로 그림1-1과 같이 나타낼 수 있다.



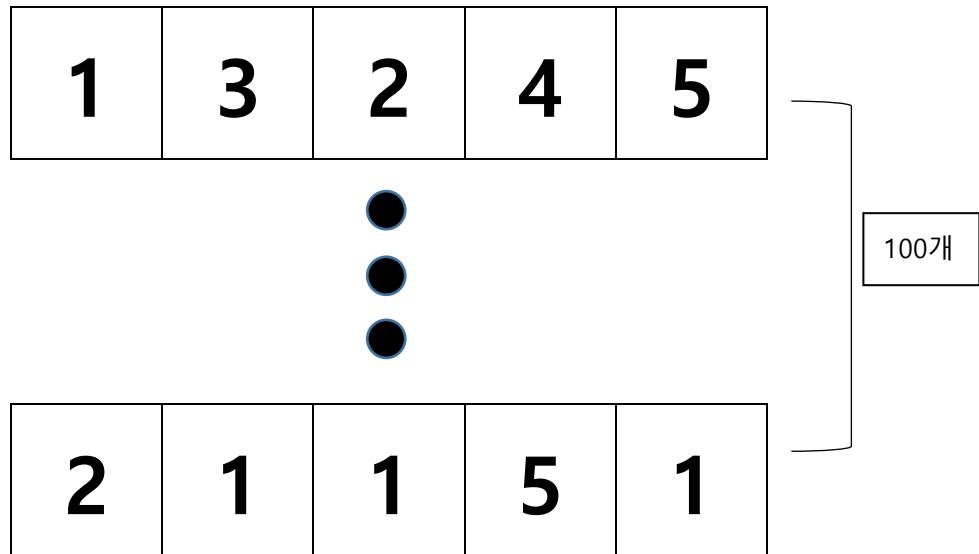
				Q
	Q			
Q		Q		
			Q	

<그림1-2. Chromosome design example>

그림1-1의 예시로 그림1-2를 나타냈다. 1번 인덱스는 3이므로 실제로는 체스판 1번 가로 열이며 3번 세로 열을 의미하고 2번 인덱스는 4이므로 2번 가로 열 4번 세로 열을 나타낸다. 나머지 인덱스도 같은 방식으로 적용된다.

2) Initialization

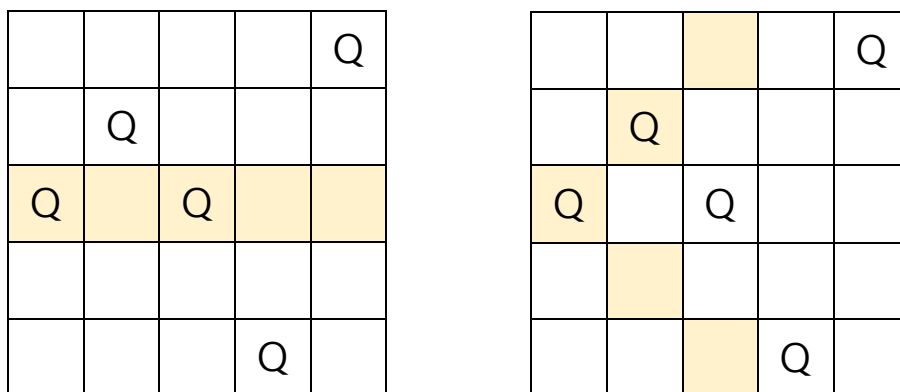
Chromosome의 초기 generation을 생성한다. 1번 단계에서 설계한 Chromosome을 일정한 개수만큼 랜덤 하게 생성한다. 이 실험 예서는 모집단의 크기를 100개로 설정하였고 그림2-1과 같이 나타냈다. 이 때 Chromosome 집합을 0세대로 지정한다.



<그림2-1. Chromosome Initialization >

3) Fitness evaluation

현재 세대의 Chromosome 적합도를 측정한다. 여기서 적합도 란 solution에 가까운 정도를 의미하는데 5-queen problem의 solution은 Queen이 서로 대각선과 가로에 겹치지 않아야 하므로 이 실험에서는 겹점의 개수 기준으로 적합도를 측정했다. 따라서 겹점이 적을수록 적합도가 높다.



<그림3-1. 1번 인덱스 Fitness evaluation >

				Q
	Q			
Q		Q		
			Q	

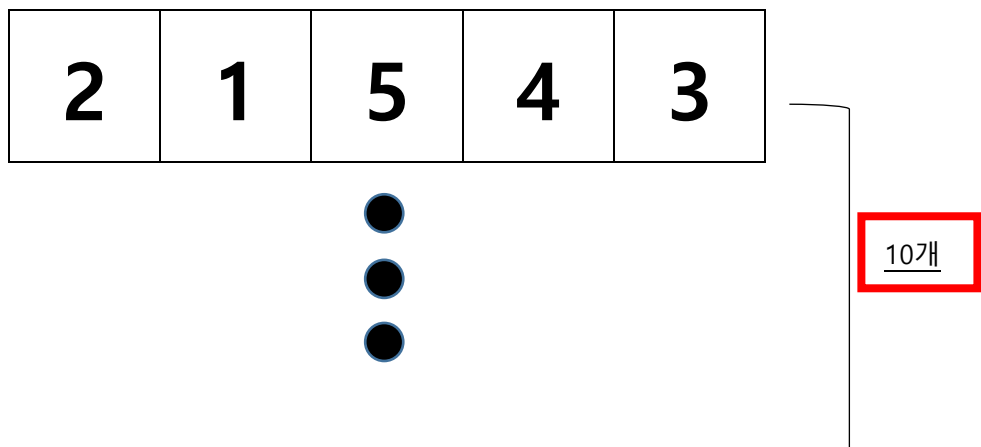
				Q
	Q			
Q		Q		
			Q	

<그림3-2. 2번 인덱스 Fitness evaluation >

그림 3-1과 3-2는 그림 1-2의 chromosome에 Fitness evaluation을 1, 2번 인덱스 까지 적용한 모습이다.왼쪽 테이블은 현재 인덱스의 Queen 기준으로 가로에 접점이 있는지 찾고, 오른쪽은 대각선에 접점이 있는지 찾는다. 그림3-1에서는 2개의 접점을 발견하였고 3-2에서는 1개의 접점을 발견하여 총 3개 이다. 이 과정을 1번부터 4번 인덱스까지 차례대로 자기보다 큰 인덱스와 접점이 있는지 찾으며 접점의 개수를 저장하여 4번 단계 selection에 사용한다.

4) Selection

Fitness evaluation에 의해 산출된 적합도를 기준으로 Selection 방법에 따라 chromosome을 선택한다. 이 실험에서는 Ranking Selection을 사용하여, 현재 세대 중 접점이 작은 순서로 10개를 추출한다. 만약 여기서 접점이 0인 optimal solution을 찾는다면 프로그램은 종료된다.



3	4	3	5	2
---	---	---	---	---

<그림4-1. Chromosome Initialization >

5) Crossover

4번 단계에서 선택된 chromosome으로 자손을 생성한다. One point crossover를 사용했으며 division point는 랜덤 하게 설정했다

2	1	5	4	3
---	---	---	---	---

3	4	3	5	2
---	---	---	---	---

<그림 5-1. Before Crossover, division point>

3	1	5	4	3
---	---	---	---	---

2	4	3	5	2
---	---	---	---	---

<그림 5-2. After Crossover >

그림 5-1은 crossover를 하기 전 chromosome이다 1~4 중 랜덤 하게 division point를 설정 할 수 있다. 그림 5-1로부터 division point가 1이 설정 되었다면 crossover를 한 경우 그림 5-2와 같이 유전자가 교차되어 자손이 생성된다.

6) Mutation

Chromosome의 인덱스를 일정한 확률로 변화시킨다. 이 실험에서는 crossover를 통해 생성된 자손을 mutation 했으며 mutation 의 발생 확률은 20%로 설정했다. Mutation에 의해 변하는 인덱스는 하나이고 랜덤 값이 설정된다. 4단계 에서 생성된 10개의 자손은 optimal solution에 근접하므로, 이 자손은 유지하고 5,6단계를 통해 생성된 자손은 4단계 에서 생성된 10개의 자손을 바탕으로 90개가 될 때까지 이 과정을 반복한다

3	1	5	4	3
---	---	---	---	---

<그림 6-1. Before Mutation >

3	4	5	4	3
---	---	---	---	---

<그림 6-2. After Mutation >

그림 6-1은 crossover에 의한 자손으로 mutation 하기 전 상태이다. 만약 mutation에 20% 확률로 선택된 chromosome이 그림 6-1이고, 2번 인덱스가 4로 설정되게 랜덤 값이 발생한다면, 그림6-2로 나타난다.

7) Update generation

6번 단계 이후 모집단의 크기만큼의 자손이 생성되면 generation을 1증가 시킨다.

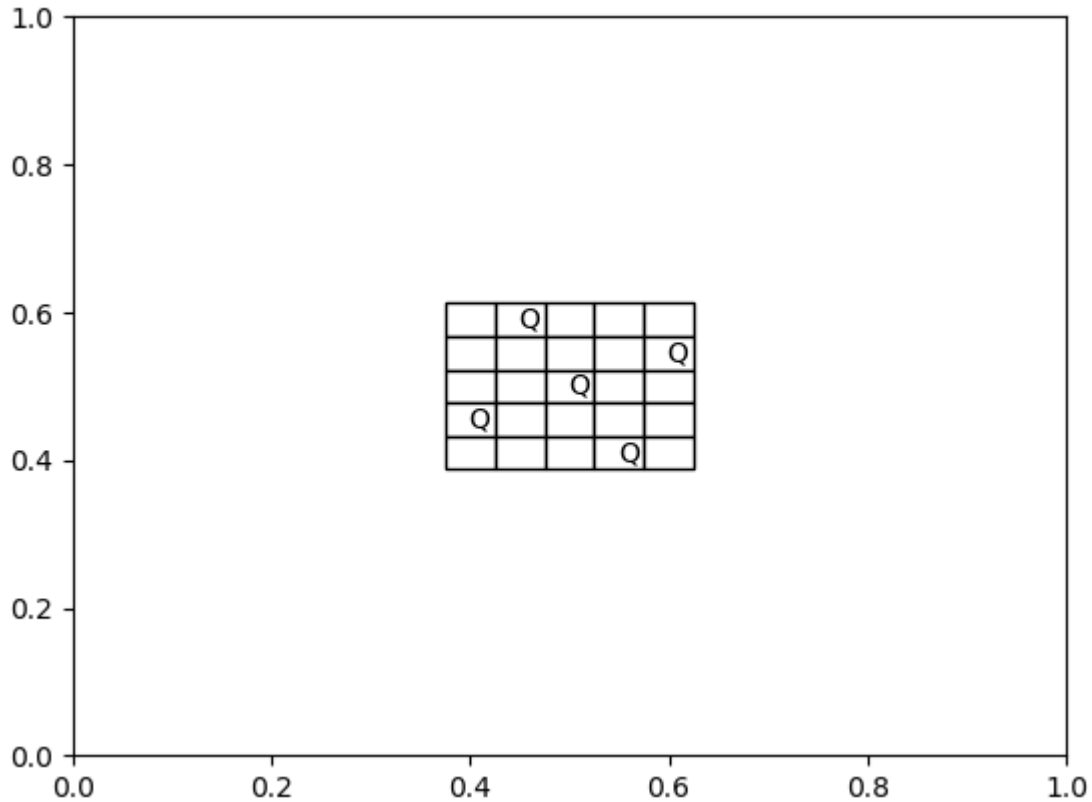
8) Go back to 3

3번 단계 Fitness evaluation 으로 돌아가서 Optimal solution인 접점0을 찾을 때까지 이 과정을 반복한다.

4. 결과

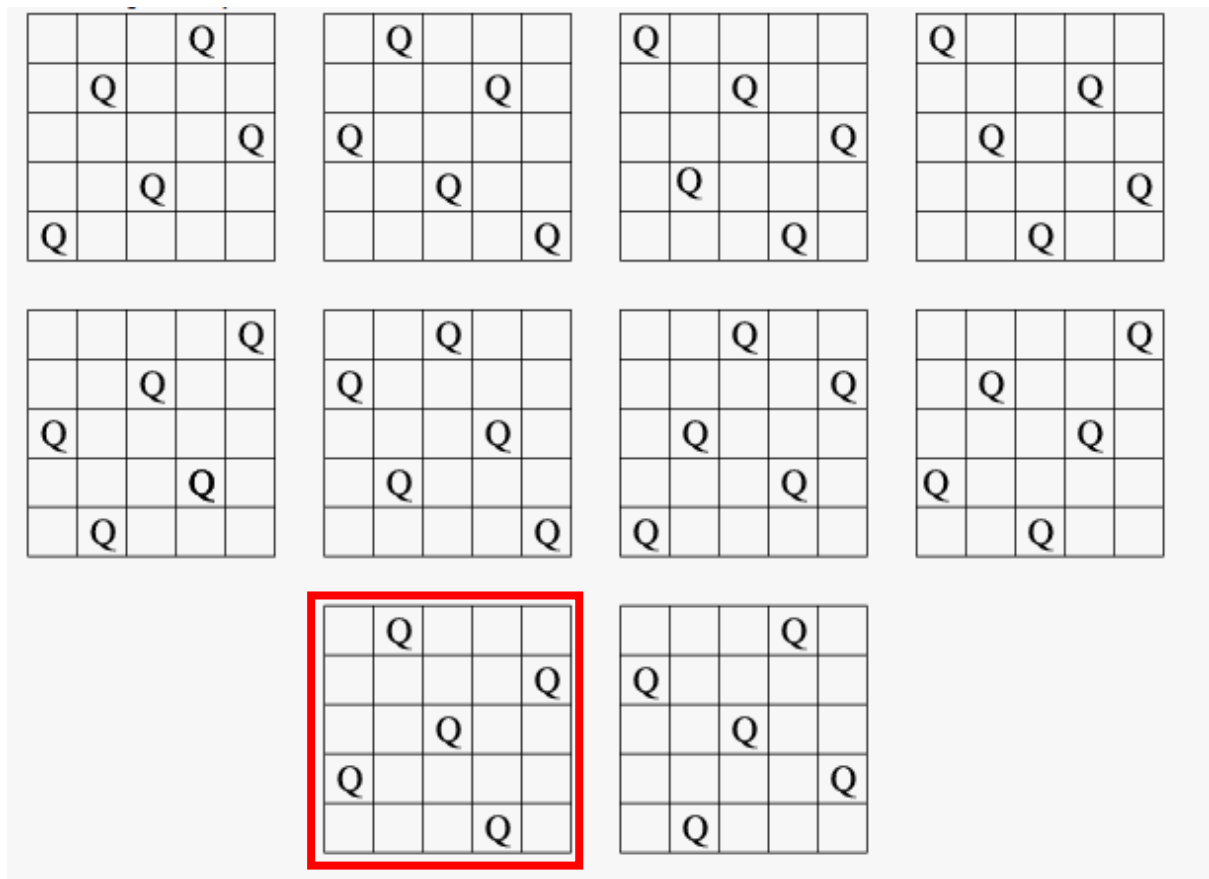
```
C:\Windows\system32\cmd.exe
I found 5 -Queen solution
current generation : 1
[2, 5, 3, 1, 4, 0]
```

<그림7-1. 5-queen cmd결과>



<그림7-2. 5-queen 결과를 table로 시각화>

유전 알고리즘은 랜덤 값에 의존하므로 결과는 실행함에 따라 다르게 나타난다. 하지만 평균적으로 2세대 만에 optimal solution을 찾아내었고, 간혹 5세대가 넘어가는 경우가 발생했다. 실험 결과는 다수의 결과 중 하나를 추출하였다. 그 결과, 그림 7-1에서 프로그램이 optimal solution을 1세대 만에 찾았으며 [2,5,3,1,4] 이라는 해를 발견했다.



<그림7-3. 5-queen solution>

이 해는 그림7-3의 빨간색 박스 안에 queen과 동일하므로 명백한 solution 이다. 그림 7-2는 그림7-1의 결과를 table로 시각화한 자료이다.