

# Genetic Algorithm with Tree Search for Traveling Salesman Problem (TSP)

Team#2

고준영 이우윤 이혜원



## **0 ) 프로젝트 목표**

- 유전 알고리즘 기법을 활용한 TSP 문제 최적화**
  - 최적해를 찾기 위한 비선형 최적화 기법의 활용**
  - 지역적 최적 트리 구조의 이해**



# 1 ) Introduction

## 〈 유전 알고리즘 기법을 활용한 TSP 문제 최적화 〉

- 유전 알고리즘이란?

- 최적화 문제를 해결하는 전역 최적화 기법의 일종.
- 생물의 진화를 모방한 진화 연산의 대표적인 기법.
- 변이(돌연변이), 교배 연산 등의 과정을 차용하여 최적의 해를 유전적으로 도출하는 알고리즘.

- TSP문제란?

- 도시의 개수와 각 도시 쌍 간의 거리들이 주어질 때, (input)
- 모든 도시를 한 번씩 방문하고 (조건 1)
- 여행을 시작한 원래 도시로 돌아올 수 있는
- 최단 거리 경로(Shortest distance Path)를 구하라 (Goal)

# 1 ) Introduction

## 유전 알고리즘 연산 수행 과정

- 1) 초기, 염색체 **모델(경로)**을 생성하여 적합성을 판단
- 2) **선택 연산**을 통해서 **부모 개체를 선택**
- 3) **교차 연산**을 통해서 부모 개체들의 교차로 **자식 개체를 생성**
- 4) **변이 연산**을 통해 **돌연변이 값을 생성하여**, 개체의 유전적 다양성 확보
- 5) 각 염색체의 **적합성을 판단**
- 6) 적합하지 않으면, 2 – 5 연산을 반복  
    || 적합하면 종료

## 2) Experiment Condition Setting

### I. 연산순서별 알고리즘 조사

- **선택 연산** : Roulette wheel selection,  
Tournament Selection
- **교차 연산** : One Point Crossover, Multi-Point Crossover,  
Uniform Crossover, Order Crossover,  
Partially Matched Crossover(PMX)
- **변이 연산** : Swap Mutation,  
Scramble Mutation,  
Inversion Mutation
- **변수 설정** : Threshold, Population Size, Generation Count

## 2) Experiment Condition Setting

### II. 실험 계획 수립

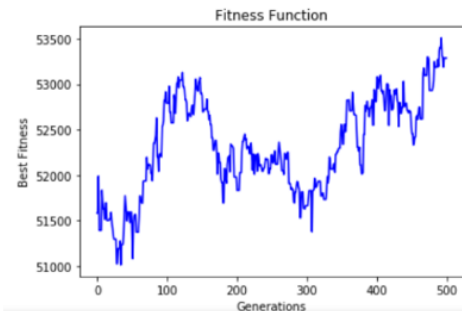
1. 각각의 (1)연산별 알고리즘들과 (2)기타 변수의 값들을 조정해가며, 적합도를 비교하는 실험을 진행
2. 연산별로 알고리즘들을 비교.  
(나머지 요소들의 값은 대조군으로 설정하여 변화를 주지 않음)
3. 최종 알고리즘 수행 전까지 실행 단계에서는,  
Population Size는 10, Generation Count는 500으로 설정

# 3) Experiment \_\_ (1) Selection

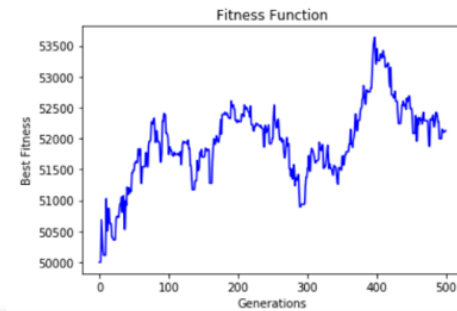
: 선택(Selection) 연산이란, 교차(Crossover) 연산에 쓰일 2개의 부모해를 고르기 위한 연산을 의미

## 1. Roulette wheel selection

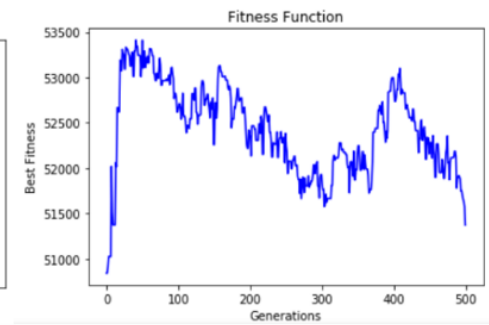
Total Best Fitness : 51012.1  
Total time : 82.46493744850159



Total Best Fitness : 50002.02  
Total time : 77.85994672775269



Total Best Fitness : 50841.97  
Total time : 77.94318532943726

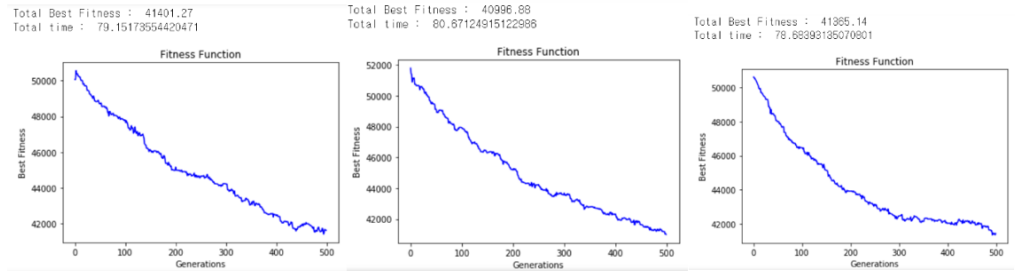


Roulette wheel selection 을 통해 유전 알고리즘을 3 회 진행한 결과, 평균 실행시간은 약 79.42 초였고, 평균 Best Fitness 는 약 50618.67 로 나타났다. ↴

# 3) Experiment \_\_ (1) Selection

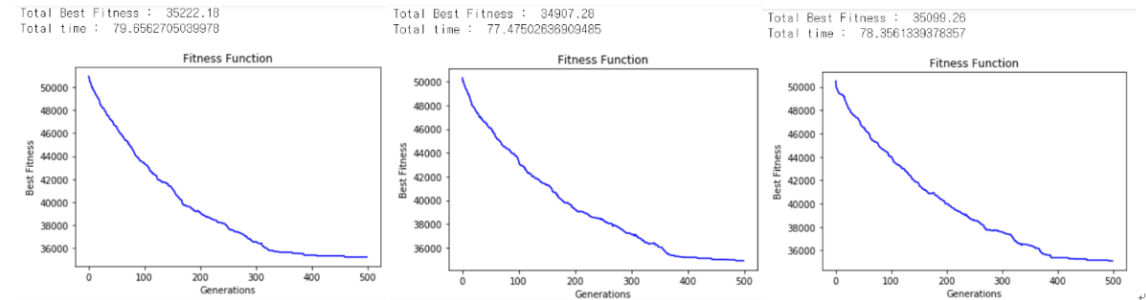
## 2. Tournament selection

### i. 염색체 2 개 선택



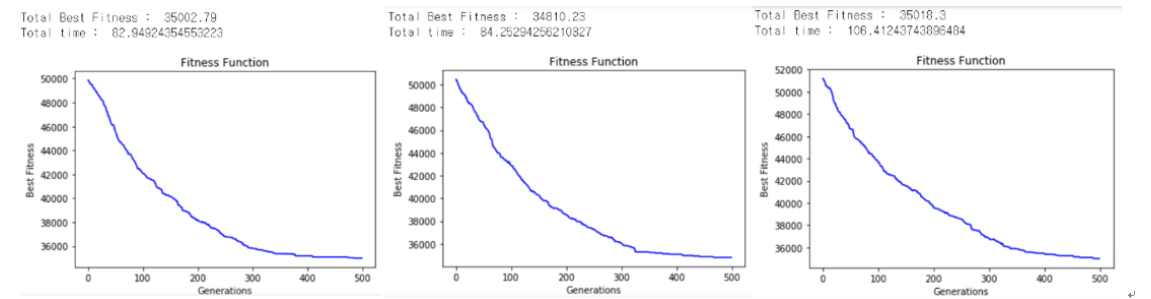
염색체를 2 개 선택해 토너먼트를 진행했을 경우, 평균 실행시간은 약 79.50 초였고, 평균 Best Fitness 는 약 41254.43 으로 나타났다.

### ii. 염색체 10 개 선택



염색체 10 개를 선택해 토너먼트를 진행했을 경우, 평균 실행시간은 약 78.50 초였고, 평균 Best Fitness 는 약 35076.24 로 나타났다.

### iii. 염색체 15 개 선택



염색체 15 개를 선택해 토너먼트를 진행했을 경우, 평균 실행시간은 약 91.20 초였고, 평균 Best Fitness 는 약 34943.77 로 나타났다.



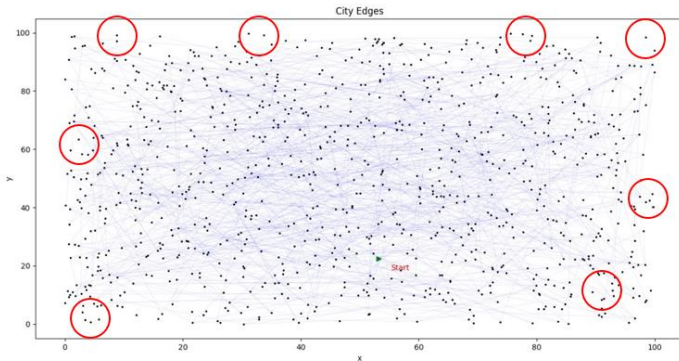


# 3) Experiment \_\_ (2) Crossover

: 교차 연산이란 선택된 두 개의 부모해들의 특정위치를 기준으로 해의 일부를 서로 교차하여 다음 세대의 해들을 생성하는 연산을 의미.

## 1. One Point Crossover

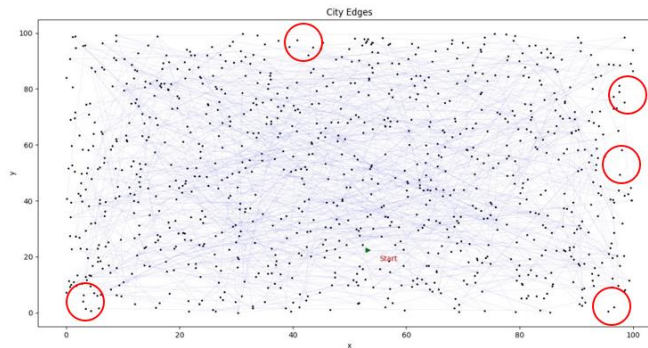
두 부모 염색체에서 한 점을 통해  
두 개로 각각 분리 후에  
두 부분을 교차하는 방식  
: 구현 간단



: 미 방문 도시 발생으로 해당 방법 제외

## 2. Two-Point Crossover

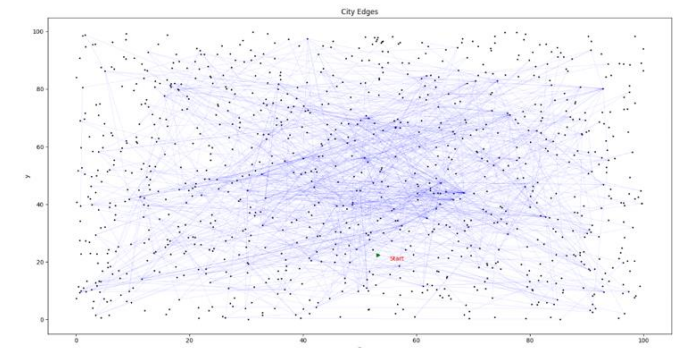
두 부모 염색체에서 두 개의 점을  
설정하여 분리 후에 일정 부분을  
교차하는 방식  
: 넓은 공간 탐색에 더 유리



: 미 방문 도시 발생으로 해당 방법 제외

## 3. Uniform Crossover

자름선을 이용하지 않고 임계확률  $P$ 를 이용.  
1) 각각의 유전자 위치에 대하여 난수를 발생  
2) 이 값이 임계확률  $P$ 보다 높으면  
2-1) 첫 번째 부모의 유전자를 가져오고,  
2-1) 낮으면 두 번째 부모의 유전자를 가져온다.

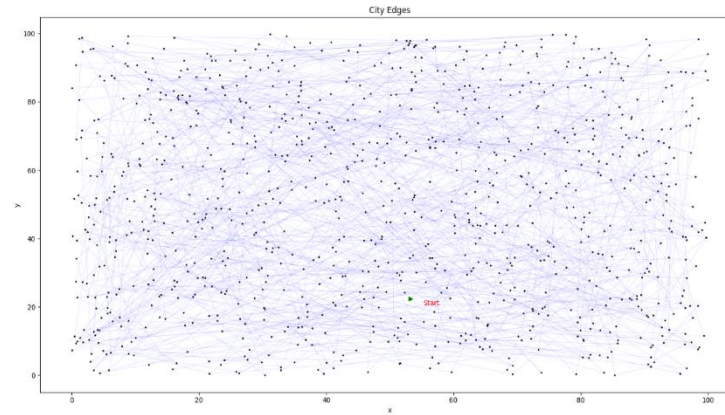
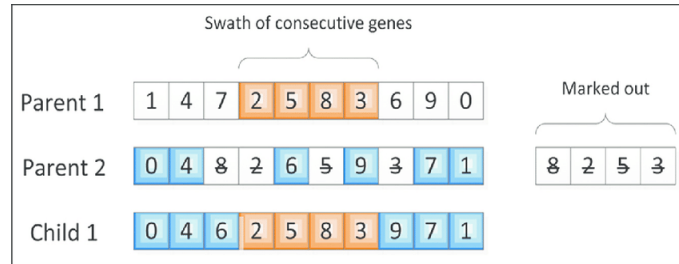


: 중복 방문 도시 발생으로 해당 방법 제외

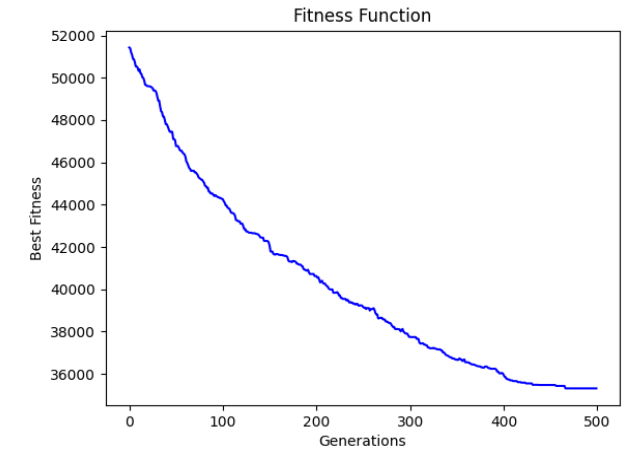


# 3) Experiment \_\_ (2) Crossover

## 4. Order Crossover.



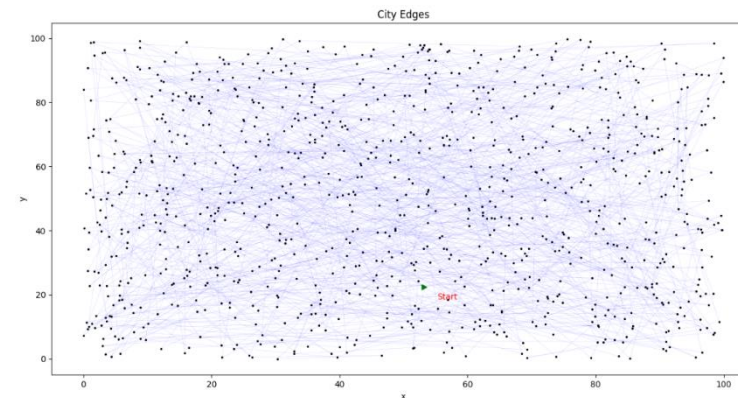
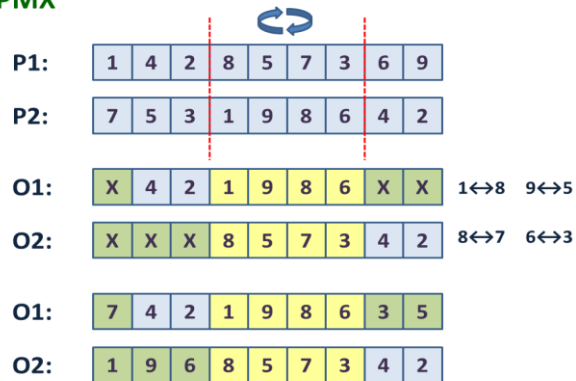
: 미 방문 도시 X, 중복 방문 도시 X



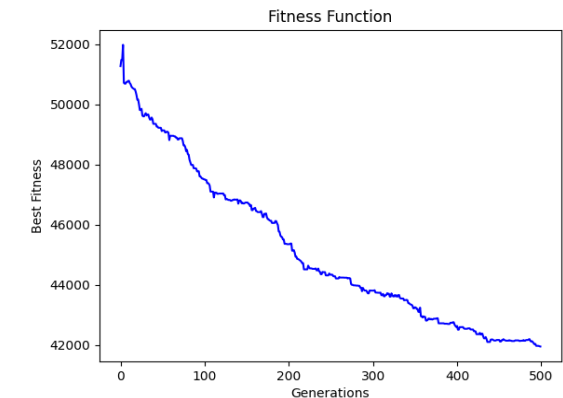
: 최종 Fitness: 35318.79

## 5. Partially Matched Crossover(PMX)

PMX



: 미 방문 도시 X, 중복 방문 도시 X

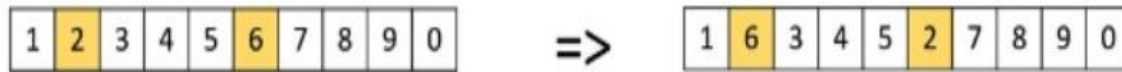


: 최종 Fitness: 41951.89



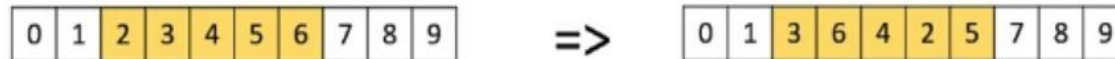
# 3) Experiment \_\_ (3) Mutation

## 1. Swap Mutation

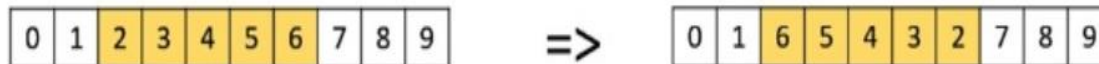


1. Order CrossOver

## 2. Scramble Mutation



## 3. Inversion Mutation

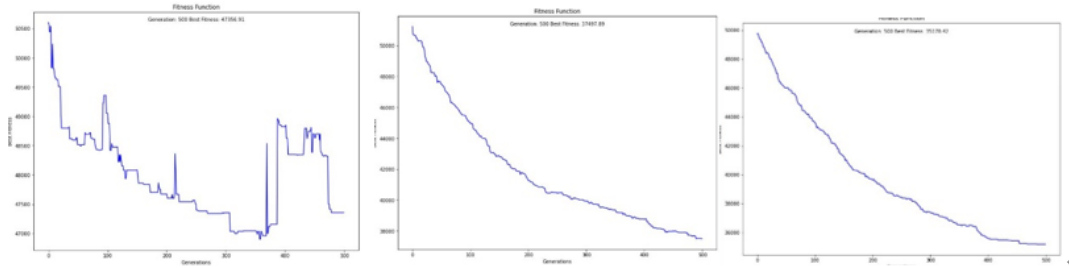


2. PMX CrossOver

# 3) Experiment \_\_ (3) Mutation

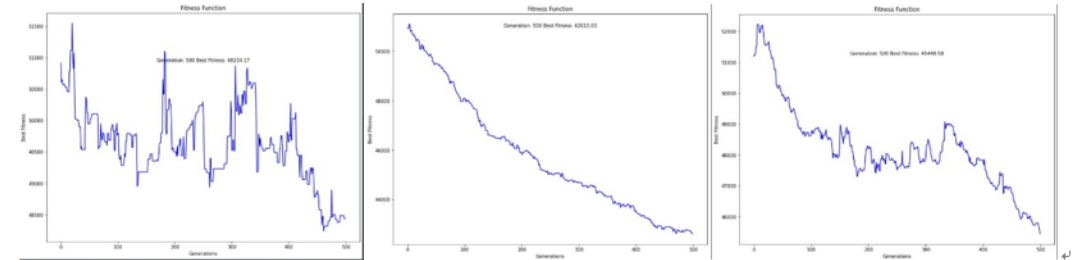
## 1) Order & PMX 에 따른 각 Mutation 적합도 비교

### I. Order Cross Over



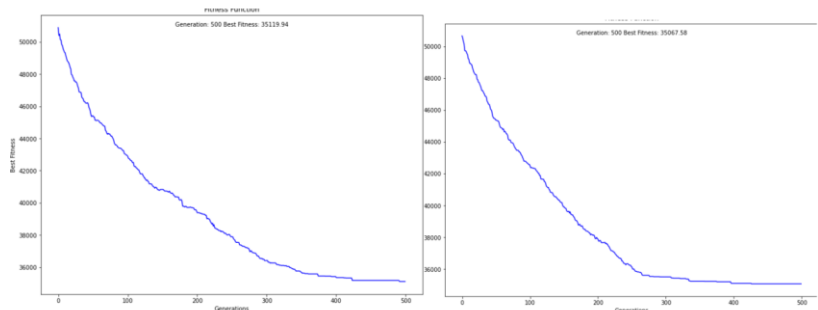
Scramble Mutation	Swap Mutation	Inversion Mutation
47358.91	37491.89	35178.42

### II. PMX Cross Over



Scramble Mutation	Swap Mutation	Inversion Mutation
48234.17	42610.03	45448.58

## 2) Swap Mutation & Inversion Mutation 적합도 추가 비교 ( Population 10 → 15 )



SwapMutation의 Best Fitness는 35119.94, InversionMutation의 BestFitness는 35067.58 로 Inversion Mutation이 조금 더 나은 성능을 보이는 것을 확인할 수 있었다.

## 3) Experiment \_\_ (4) Variables

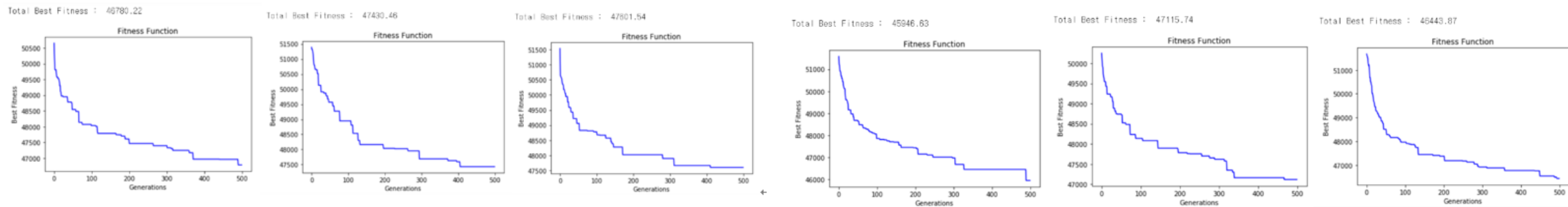
- 1) 임계점을 나타내는 **THRESHOLD**
- 2) 모집단의 크기를 나타내는 **population\_size**
- 3) Generation의 수를 나타내는 **Generation\_Count**

# 3) Experiment \_\_ (4) Variables

## 1) THRESHOLD: 유전체의 적합도를 판단하는 기준인 임계값

(1) THRESHOLD = 50000

Generation 초반 적합도: 10회 평균 약 51794.92 => 초기 THRESHOLD는 50000으로 설정



( THRESHOLD 미설정 / 평균 적합도 : 47270.74 ) ↵

( THRESHOLD = 50000 / 평균 적합도 : 46502.08 ) ↵

# 3) Experiment \_\_ (4) Variables

## 1) THRESHOLD: 유전체의 적합도를 판단하는 기준인 임계값

(2) THRESHOLD = [ 40000 , 50000 )

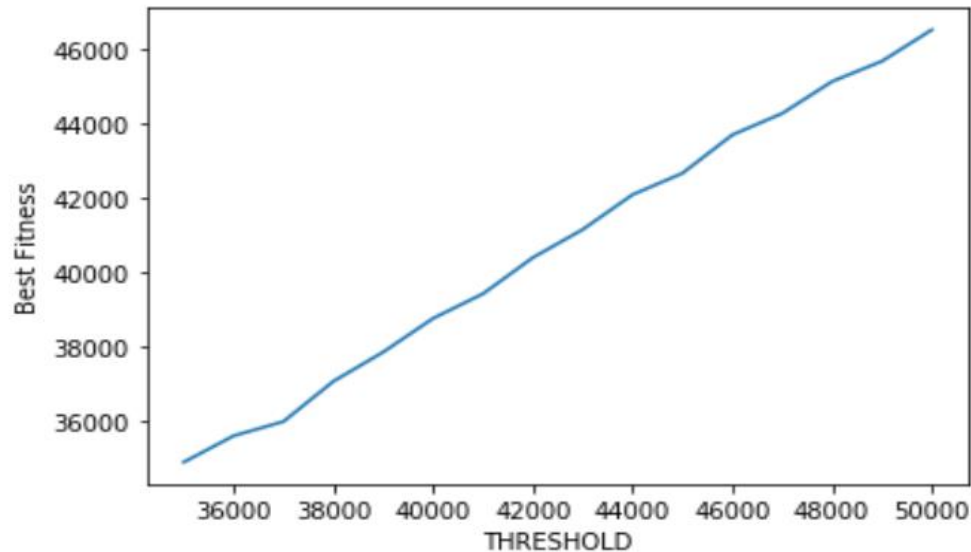
THRESHOLD	Average FITNESS	THRESHOLD	Average FITNESS
49000	45660.93	44000	42085.04
48000	45117.19	43000	41150.51
47000	44261.61	42000	40397.35
46000	43687.05	41000	39427.94
45000	42658.61	40000	38769.48

(2) THRESHOLD = [ 35000 , 40000 )

THRESHOLD	Average FITNESS
39000	37866.77
38000	37082.67
37000	36004.96
36000	35622.39
35000	34916.22

### 3) Experiment \_\_ (4) Variables

1) **THRESHOLD:** 유전체의 적합도를 판단하는 기준인 임계값



임계값 더 낮게 설정 => 좋은 적합도

but

- 1) 모집단 다양성 하락,
- 2) 더 많은 유전 개체의 제거 -> 시간 증가

-> Threshold : 35000 임의 설정



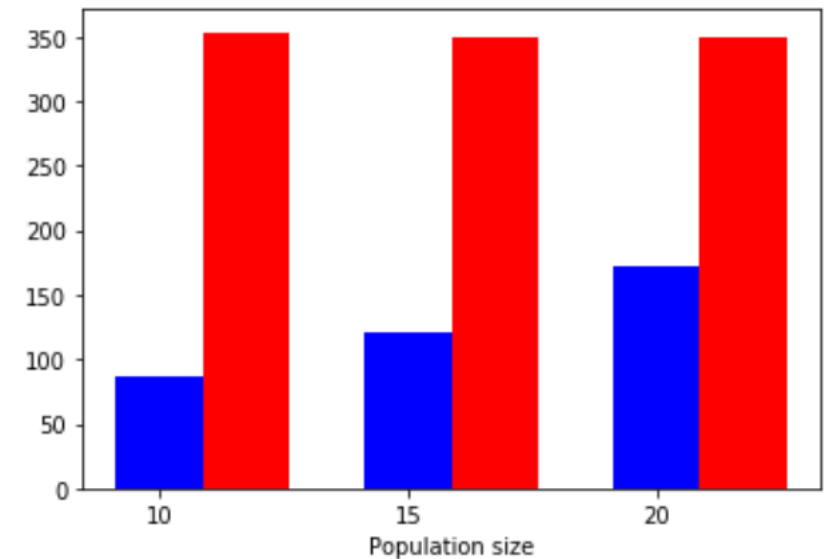
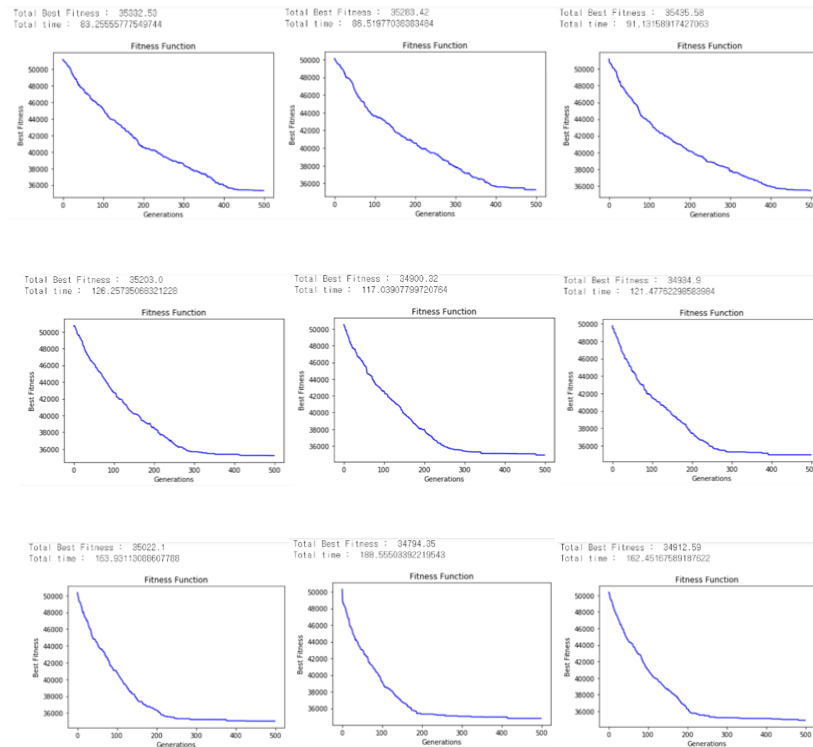
# 3) Experiment \_\_ (4) Variables

## 2) Population Size

(1) Population Size = **10**  
- Time cost: 86.97 sec  
- Best Fitness: 35350.51

(2) Population Size = **15**  
- Time cost: 121.59 sec  
- Best Fitness: 35012.74

(3) Population Size = **20**  
- Time cost: 171.65 sec  
- Best Fitness: 34909.68



(BestFitness \* 0.01)

Time : BestFitness

=> Pop Size : 15

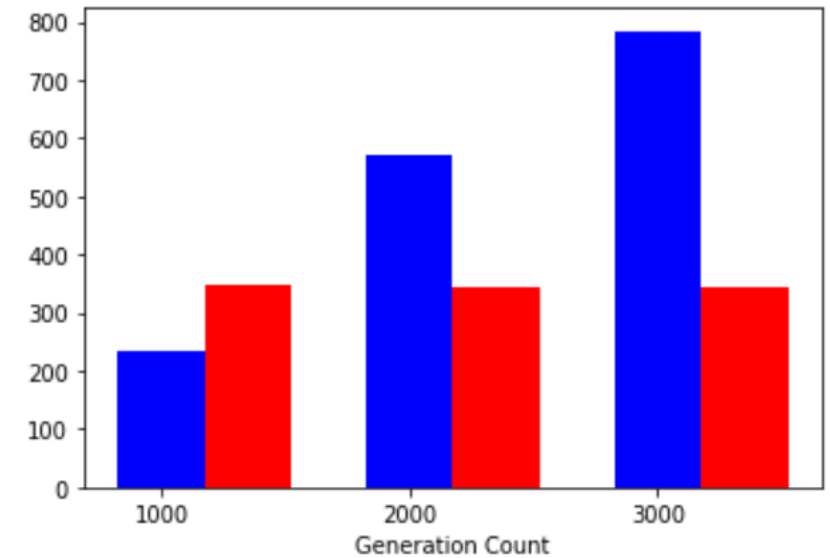
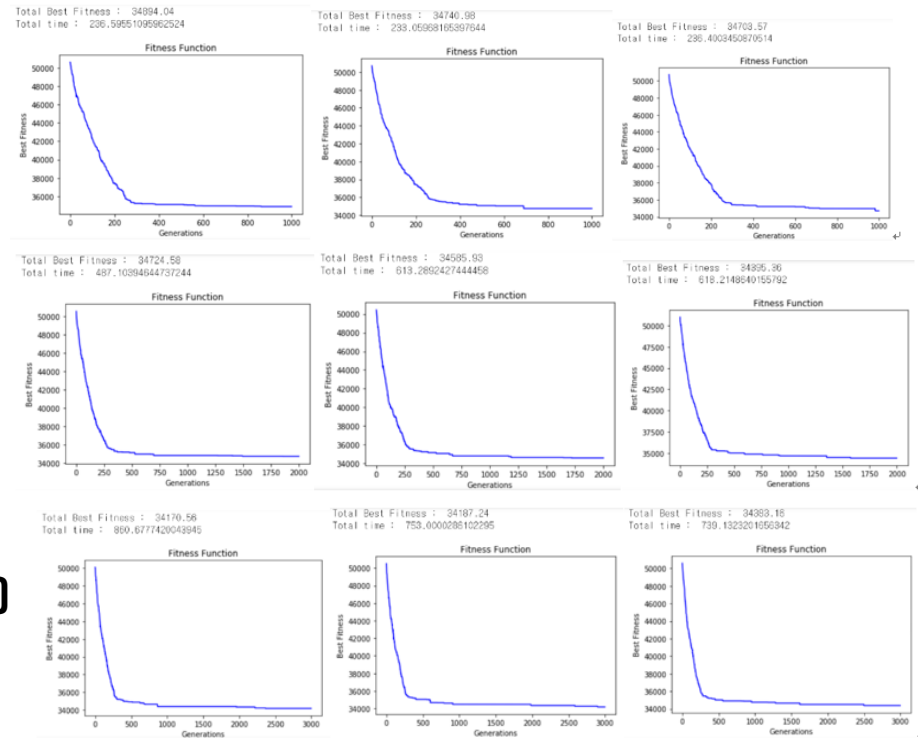
# 3) Experiment \_\_ (4) Variables

## 3) Generation Count

- (1) Generation Count: **1000**
- Time cost: 235.35 sec
  - Best Fitness: 34779.53

- (2) Generation Count: **2000**
- Time cost: 572.87 sec
  - Best Fitness: 34568.62

- (1) Generation Count: **3000**
- Time cost: 784.27 sec
  - Best Fitness: 34246.99



(BestFitness \* 0.01)

Time : BestFitness

=> Generation Count : 5000

## 4) Discussion & Conclusion

선택 연산 : Roulette wheel selection,  
**Tournament Selection**

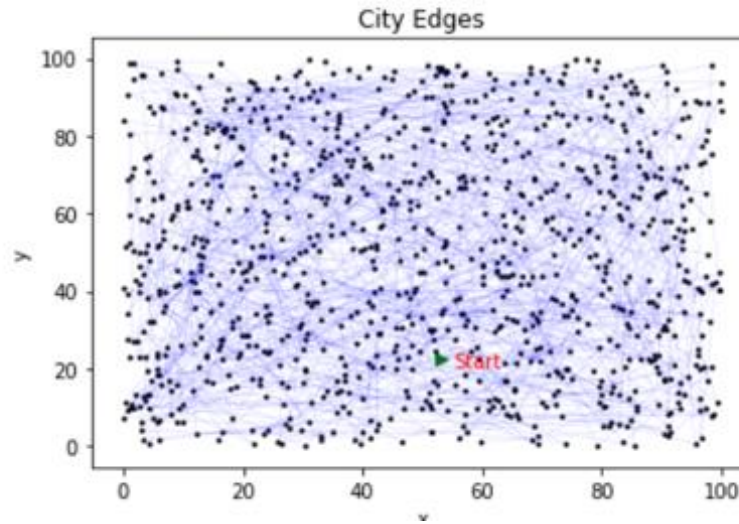
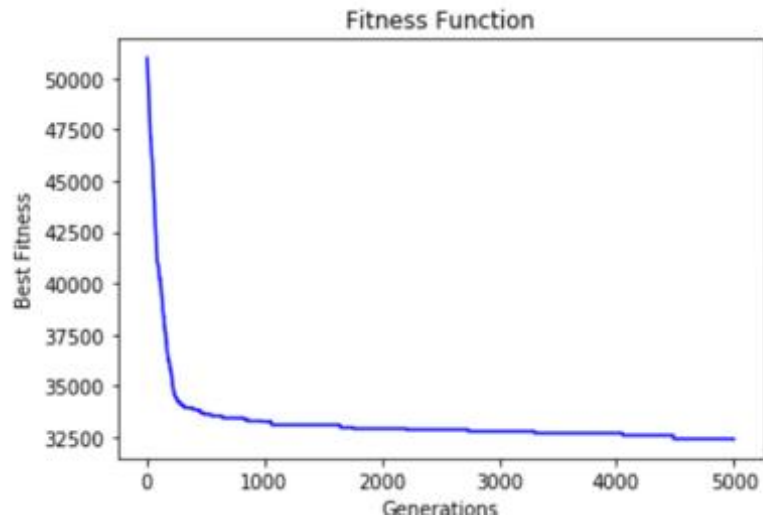
교차 연산 : One Point Crossover, Multi-Point Crossover,  
Uniform Crossover, **Order Crossover**,  
Partially Matched Crossover(PMX)

변이 연산 : Swap Mutation,  
Scramble Mutation,  
**Inversion Mutation**

변수 설정 : Threshold = **36000**, Population Size = **15**, Generation Count = **5000**

## 4) Discussion & Conclusion

Total Best Fitness : 32407.21  
Total time : 1721.507606267929



- 선택 연산 : Tournament Selection
- 교차 연산 : Order Crossover
- 변이 연산 : Inversion Mutation
- 변수 설정 : Threshold = 36000, Population Size = 15,  
Generation Count = 5000

**Total Best Fitness : 32407.21**  
**Total Time: 약 1721.508 sec**

## 4) Discussion & Conclusion

- **완전 무작위 탐색** :  $O(N-1!)$ 의 시간복잡도 소요  
1000개의 도시  $\Rightarrow$  999!의 경로 탐색
- **유전 알고리즘** : 선택, 교차, 변이의 과정을 거쳐, Fitness값을 비교적 빠르게 계산하여, 비교적 최적의 해를 보다 빠르게 도출 가능
- **실험 내용 요약** : 각 연산 단계의 알고리즘, 변수를 다양하게 조정하며 연산단계에서 좋은 Fitness를 얻기 위하면서도, 해의 다양성을 잃지 않기 위해 노력하였음.
- **실험의 한계** : 임의로 제한된 실험환경, 실험횟수 내에서 진행  
 $\Rightarrow$  다른 데이터에도 진행 가능하다는 일반화를 위한 더 많은 근거 필요

# 5) References

## References

- 문병로, 『쉽게 배우는 유전 알고리즘 : 진화적 접근법』, 한빛 아카데미(2014)
- Chudasama, C., Shah, S. M., & Panchal, M. (2011, December). Comparison of parents selection methods of genetic algorithm for TSP. In International Conference on Computer Communication and Networks CSI-COMNET-2011, Proceedings (pp. 85-87).
- Pandey, H. M. (2016). Performance evaluation of selection methods of genetic algorithm and network security concerns. Procedia Computer Science, 78, 13-18.
- Rexhepi, A., Maxhuni, A., & Dika, A. (2013). Analysis of the impact of parameters values on the Genetic Algorithm for TSP. International Journal of Computer Science Issues (IJCSI), 10(1), 158.
- Romit S Beed, Sunita Sarkar, Arindam Roy, & Shubham Chatterjee (2017). A Study of the Genetic Algorithm Parameters for solving Multi-Objective Travelling Salesman Problem. Bhubaneswar, India.