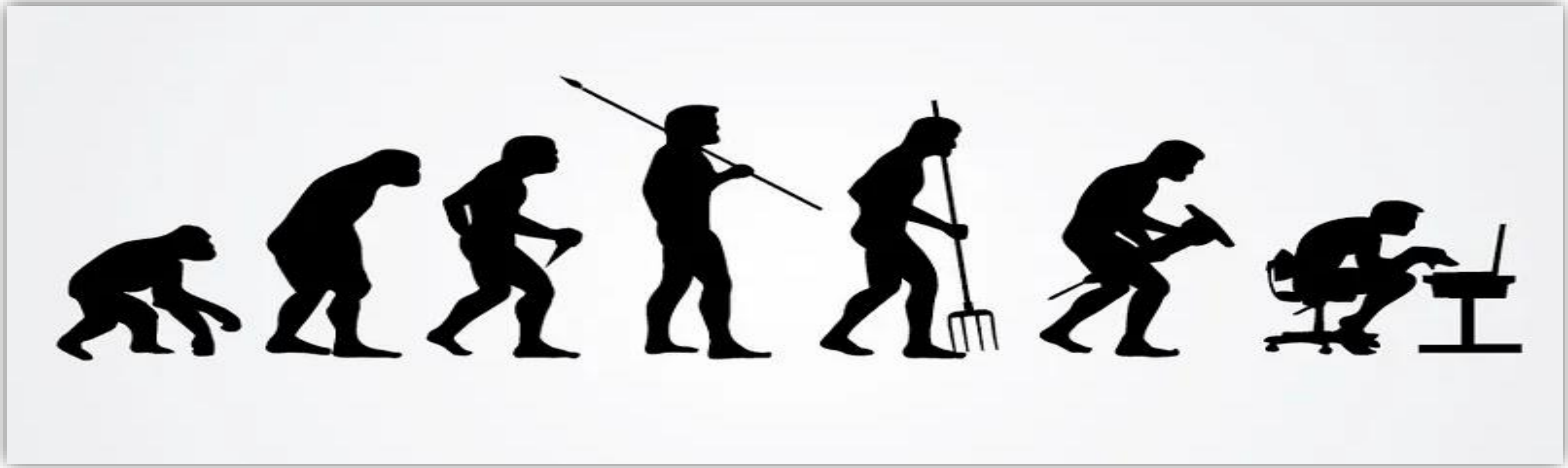
OPTIMIZATION OF TRAVELLING SALESPERSON PROBLEM BY GENETIC ALGORITHM

PRESENTED BY – PRANAB KUMAR PAUL



INTRODUCTION

- The **travelling salesman problem** asks the following question: "*Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city*?"
- **Modelled** as an undirected weighted graph, G=(V,E) that has a nonnegative integer cost c(u,v) associated with each edge (u,v) Є E such that cities : vertices, paths : edges, and a path's distance : edge's weight.
- It is a ***minimization problem*** starting and finishing at a specified vertex after having visited each other vertex exactly once.

MOTIVATION

- two approaches to solve a TSP: **Brute Force methods** and **heuristic methods**.
- TSP is a NP complete problem requires $O(n!)$ time for brute force solution.
- By *dynamic programming* it is solved in $O(2^n n^2)$ time which is still exponential time complexity.
- So here we use *Genetic algorithm* which is a heuristic approach to solve TSP because of its flexibility and robustness and to get a near by optimal solution with in less time.

IDEA BEHIND GENETIC ALGORITHM

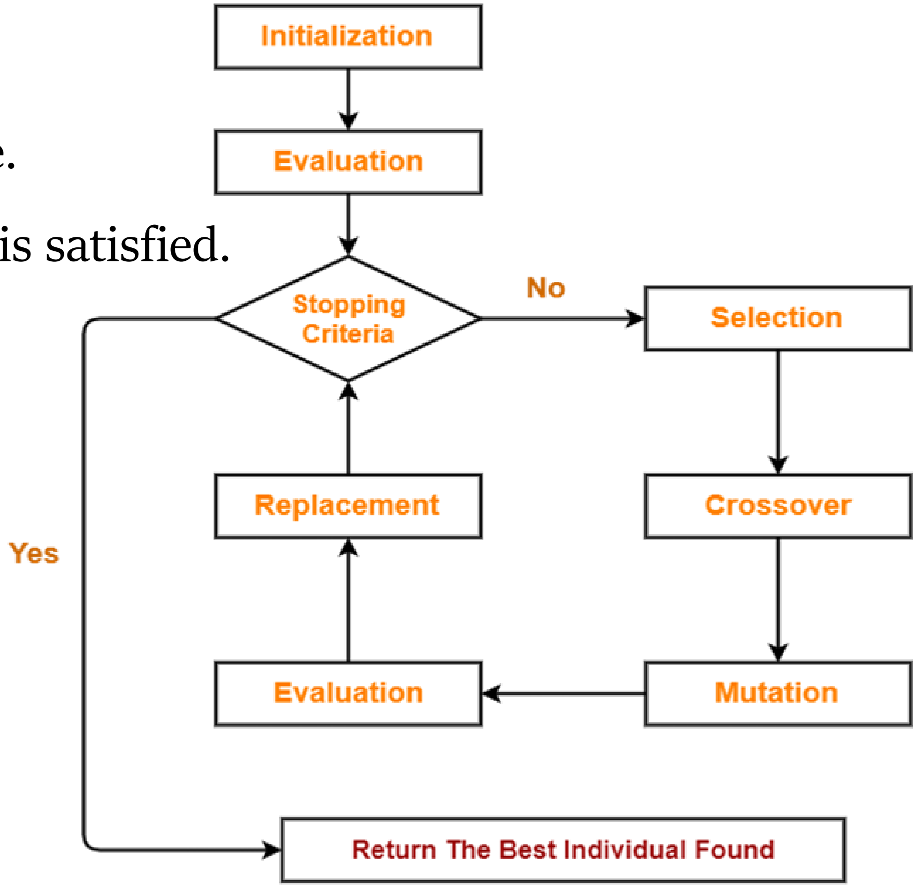
- Based on Darwin's Principle of ***Natural Selection***.
- Darwin's principle of Natural Selection can be stated as "**Survival of the fittest**".
- Genetic algorithms use techniques inspired by **evolutionary biology** such as *inheritance, mutation, selection,* and *crossover*.
- The evolution process stops when some predefined **termination condition** is satisfied. At each intermediate stage, the old generation is replaced by the new generation.

THE ALGORITHM

Procedure GA:

- **Step 1.** Initialize the population: **current population &** Evaluate.
- **Step 2.** Repeat step 3 through step 5 till **termination condition** is satisfied.
- **Step 3.** Apply **selection** to obtain the mating pool.
- **Step 4.** **Crossover** and **Mutate** the mating pool to generate

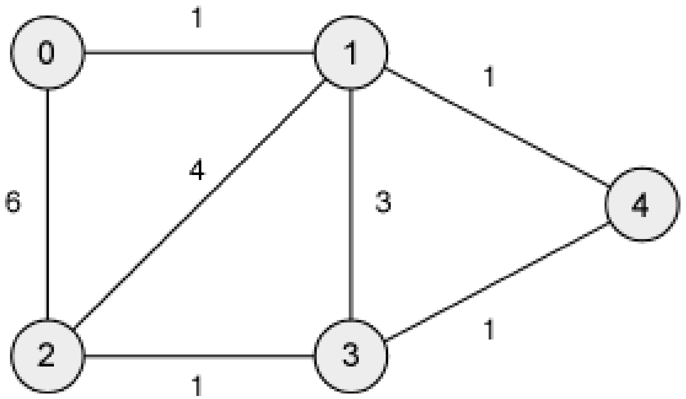
  The new population.
- **Step 5.** **Replace** the current population by the new population.
- **Step 6.** Return the best solution of the current population.



MAPPING WITH TSP

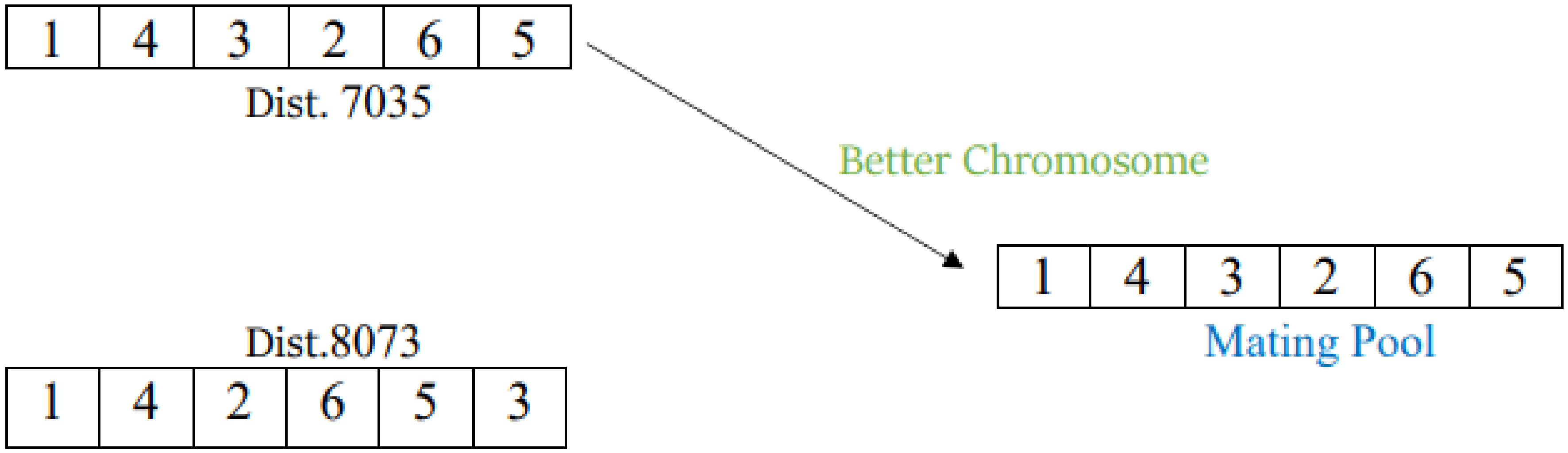
There are certain **features** associated with a GA. These are,

1. The **chromosomes**.
2. **Fitness function** to evaluate each solution.
3. **Population size** i.e., Initial population.
4. The **mating pool**, i.e., the set of chromosomes after selection.
5. GA operators, e.g., **selection**, **crossover**, and **mutation**.
6. Various GA parameters, e.g., **crossover probability** (pc), **mutation probability** (pμ).
7. **Termination condition**.



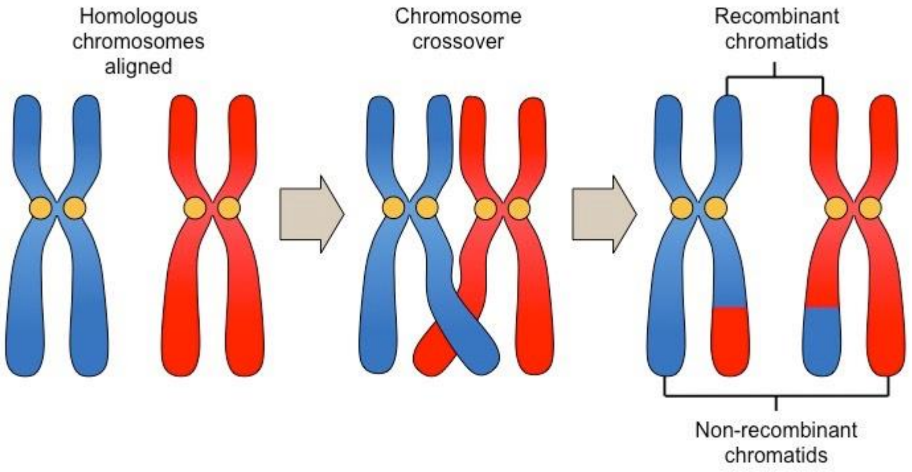
SELECTION

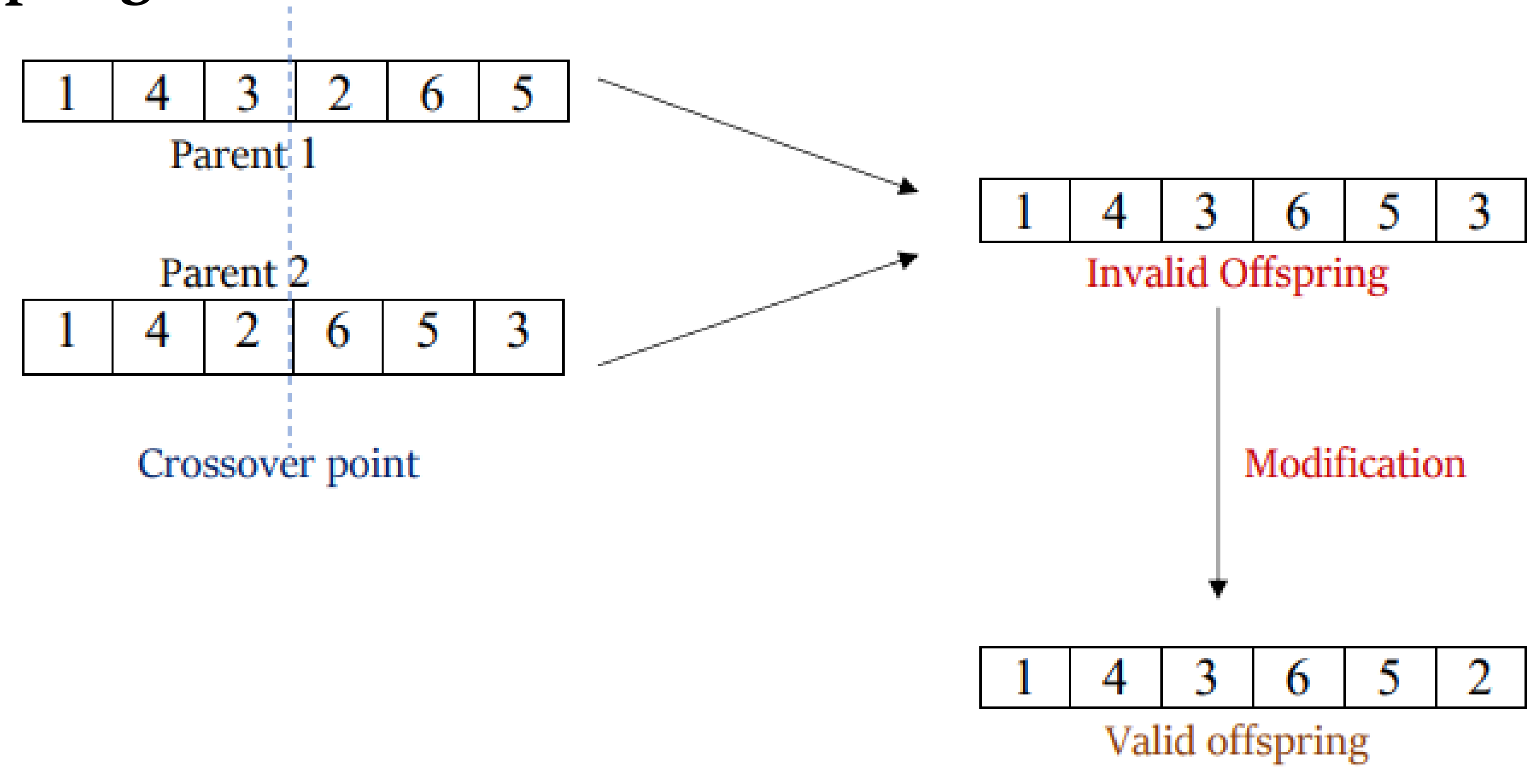
- Tournament selection.
- Selection based on Fitness value.



CROSSOVER

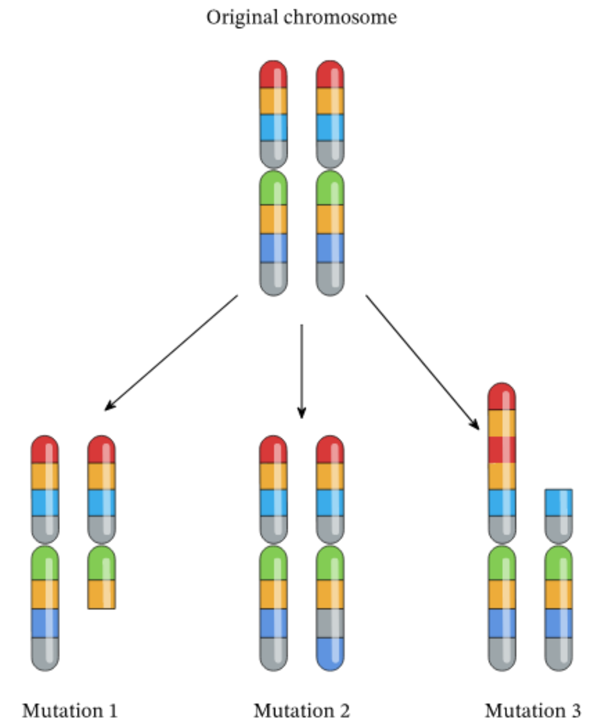
- Here we use the **Single point crossover** [6]method.
- may produce some **invalid offspring.**





MUTATION

- randomly selecting any two cities and **flipping** [3] [5] their cities in between them.



TERMINATION CONDITION

- Here I implemented terminating condition as follows, if the *minimum fitness value of the population does not change for the 100 iterations*, then the algorithm will converge and give the solution as the best tour length and the city sequences i.e., best chromosome.

RESULTS

- I've implemented the above-mentioned solution in MATLAB 2021a with the system configuration: The algorithm is implemented in MATLAB programming language in a computer with Intel(R) Pentium(R) CPU G4400 @ 3.30GHz, 3300 MHz, 2 Core(s), 2 Logical Processor(s), Windows 10 Operating System and 4 GB RAM.
- Error Rate = $\frac{Best\ Solution - Optimal\ Solution}{Optimal\ Solution * Cities}$

RESULTS(CONT….)

- Inputs are taken from these TSP Libraries:
- TSP - Data for the Traveling Salesperson Problem (fsu.edu).
- CITIES - City Distance Datasets (fsu.edu).
- National Traveling Salesman Problems (uwaterloo.ca).
- jorlib/jorlib-core/src/test/resources/tspLib/tsp at master · coin-or/jorlib · GitHub

RESULTS(CONT….)

Problems From TSPLIB	Cities	Population	Generation	Best solution	Optimal Solution	Error Rate(%)
P01	15	100	100	291	291	0.00
GR17	17	100	146	2085	2085	0.00
FRI26	26	400	280	937	937	0.00
WI29	29	3000	378	26703	26703	0.00
DJ38	38	5000	454	6656	6656	0.00
DANTZIG42	42	300	347	699	699	0.00
SWISS42	42	1000	500	1366	1273	0.17
ATT48	48	3000	846	33889	33523	0.02
ELI51	51	1000	1002	442	426	0.07
ST70	70	1000	1341	786	675	0.23
kroA100	100	1500	2508	24386	21282	0.14
pr124	124	5000	1586	66286	59030	0.12
pr152	152	5000	2342	85962	73682	0.11
QA194	194	1000	5369	11563	9352	0.19
kroA200	200	5000	3659	41205	29368	0.20

GRAPHICAL REPRESENTATION

X-axis: Generation **Y-axis:** Minimum Distance Plotting
Orange: Average Fitness value Green: Minimum Distance

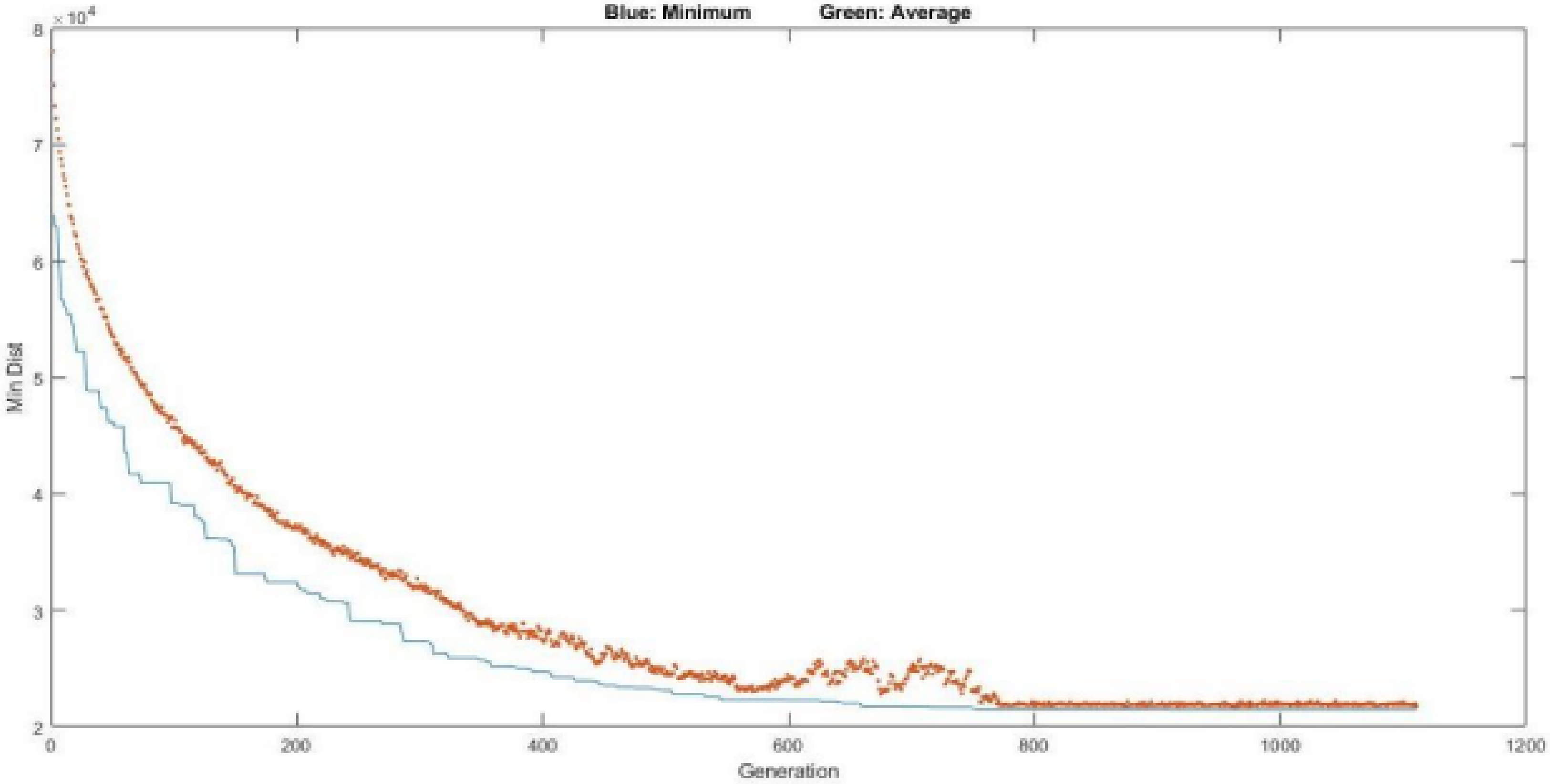


Fig: Graphical Representation of Average and Minimum Distance

BAR CHART

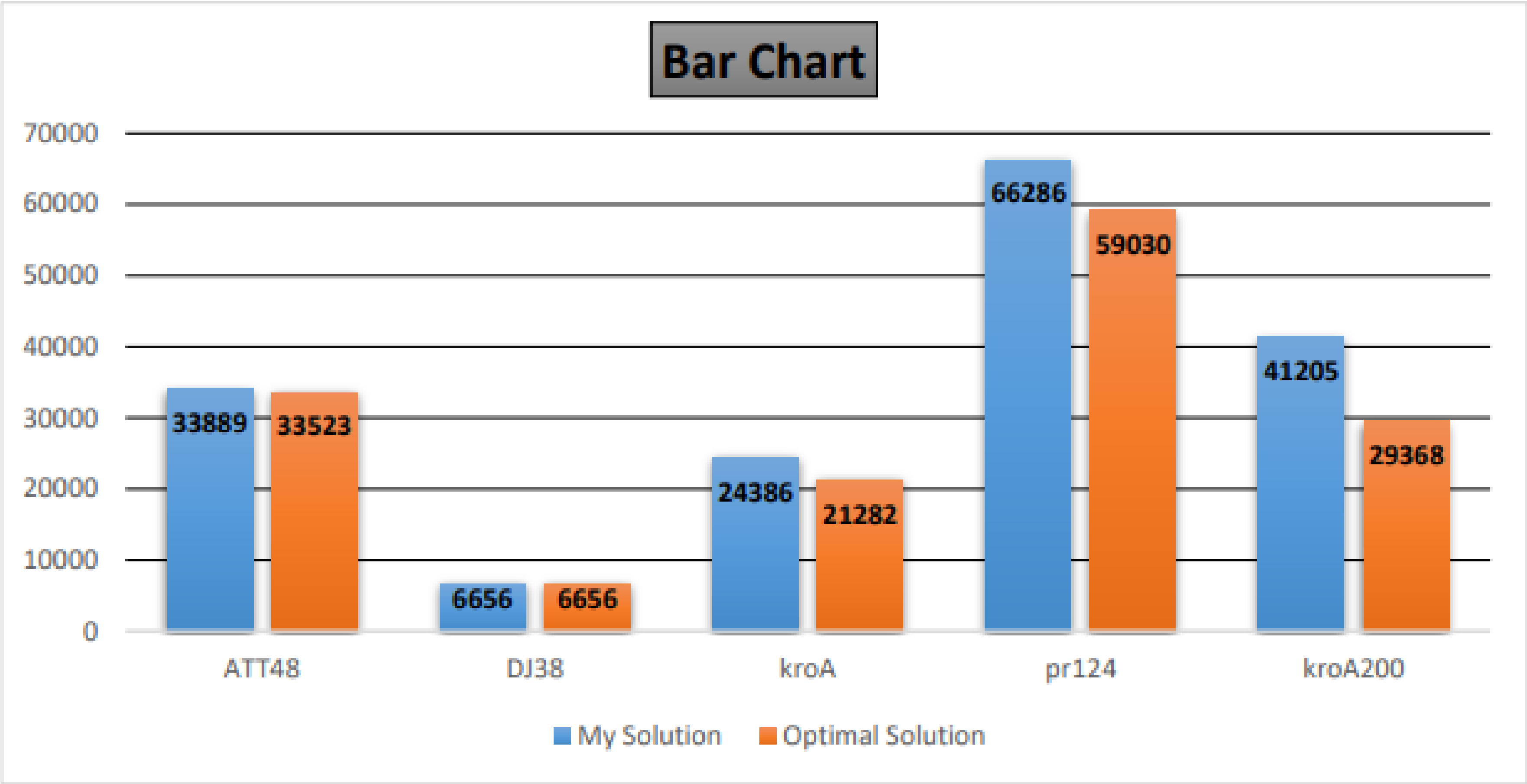


Fig: Bar Plot of some instances of Optimal solution and My Solution

Y-Axis: Minimum Distance (in km) X-Axis: Problem Instances

LINE PLOT

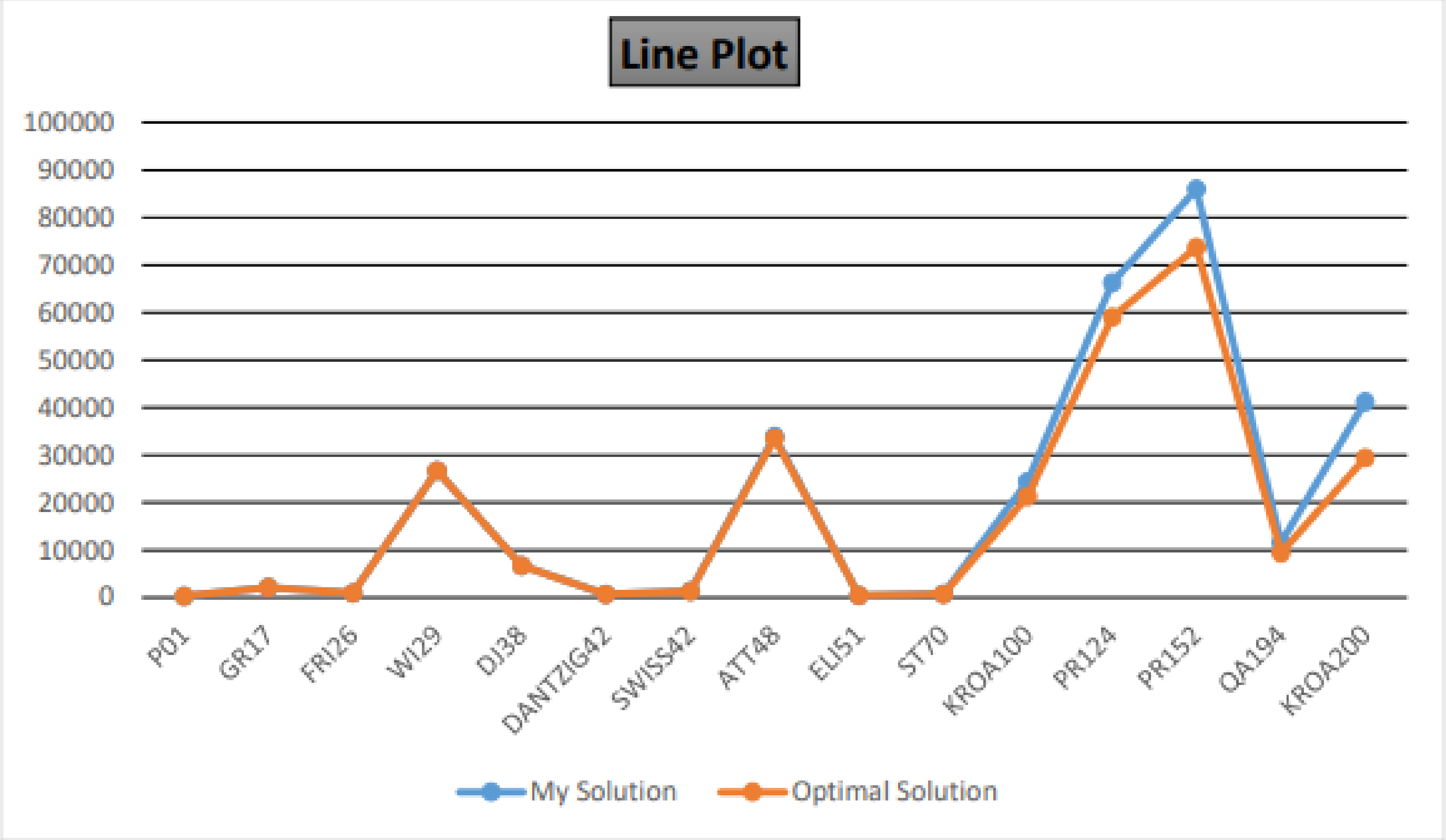


Fig: Line Plot of all instances of Optimal solution and My Solution.

CONCLUSION

- By combining the knowledge of GAs for solving TSP is an optimistic approach. Depending on the manner, the problem is encoded and which crossover and mutation methods are used, genetic algorithm find fine solutions for the travelling salesman problem. Through this paper **our objective** is to give a more effective process for solving TSP by using the genetic algorithm. In this paper we have solved the symmetric TSP but **in future** we would like to optimize this solution to more accurate and try to solve asymmetric TSP as well.

ACKNOWLEDGEMENT

- I want to sincerely thank my supervisor **Prof. Priya Ranjan Sinha Mahapatra** and **Dr. Soumen Atta** and **Prof. Anirban Mukhopadhyay** for encouraging me and helping me out at each and every step of this work. Again, I want to thank to permit me in Research LAB, KU for supporting this project.

REFERENCES

- [1] (PDF) "Solving Travelling Salesman problem by New Optimization Algorithm" (researchgate.net).
- [2]Introduction-to-soft-computing-neuro-fuzzy-and-genetic-algorithms-by-samir-roy-udit-chakraborthy.
- [3] An Introduction to Genetic Algorithms:Method and Implementation (Anirban Mukhopadhyay).
- [5] Adewole Philip, AkinwaleAdioTaofiki and OtunbanowoKehinde, "A Genetic Algorithm for Solving Travelling Salesman Problem", International Journal of Advanced Computer Science and Applications, 2011, Vol. (2), No. (1).
- [6] R. Poli and W. B. Langdon. "Genetic Programming with One-point Crossover and Point Mutation". Technical Report CSRP-97-13, University of Birmingham, School of Computer Science, Birmingham, B15 2TT, UK, 15 Apr. 1997.
- [7] Exploring Travelling Salesman Problem using Genetic Algorithm (ijert.org).
- [8] (46) An Introduction to Genetic Algorithms: Method and Implementation (Lecture 1) by Anirban Mukhopadyay – YouTube.

Thank You!