Web visualization of Genetic Algorithm for TSP

Ostap Maliuvanchuk, Institute of Computer Science, University of Tartu

Introduction

Genetic algorithm is an algorithm that is widely used to solve optimization and search problems. It belongs to the class of evolutionary algorithms and based on a process of natural selection. With a help of genetic algorithm problems such as Travel Salesman Problem can be solved much faster if compare to deterministic algorithms.

Objectives

Create a web tool for students that can visually show how the genetic algorithm works and how the different parameters affect the end result. The tool should be simple, easy to use and šelf explanatory. Parameters such as crossover rate should be changeable throught the web interface.

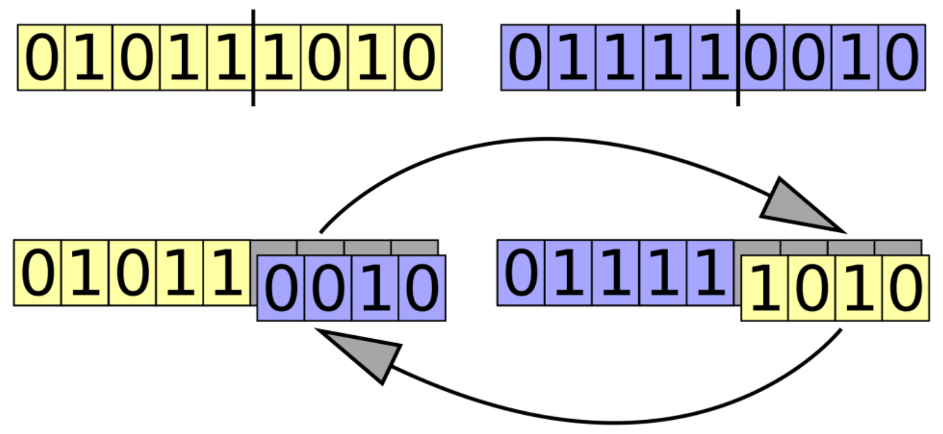
Algorithm

Here are the main terms which are needed to explain how the algorithms works.

- **Chromosome** – single possible solution to a problem. In case of TSP it is ordered list of point which represents a path.
- **Population** – a list of chromosome
- **Fitness** – a function which return numeric value for chromosome and is used to pick the best result. In case of TSP it is the length of the path.

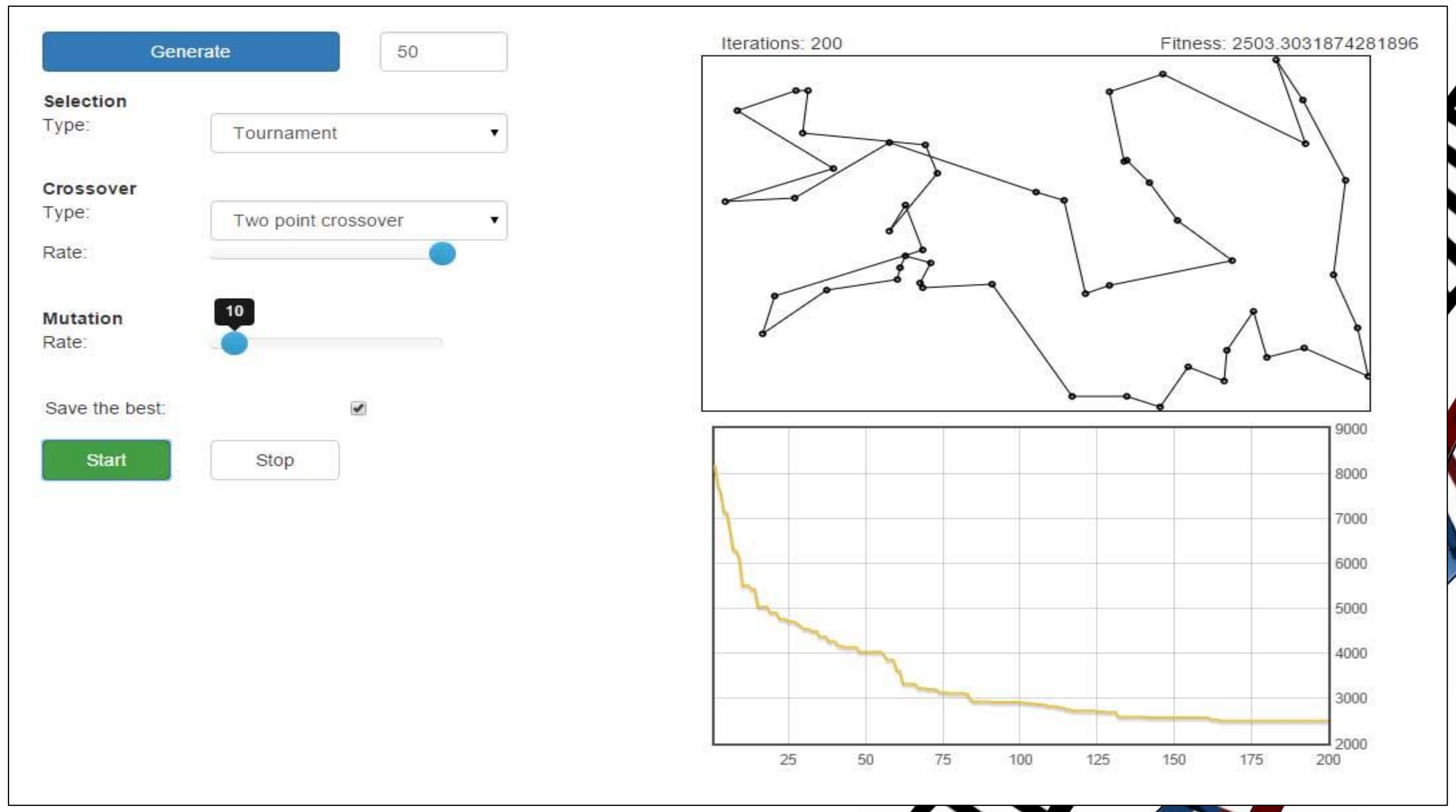
At the beginning of the algorithm we create a random population and then start an iterative process of natural selection with the following steps:

1. **Selection** – process of selection the individual chromosomes for further procession based on the fitness function. There are several types of selection, such as roulette wheel selection, random selection, tournament selection.
2. **Crossover** – process of exchanging information between two chromosomes. The frequency of crossover is controlled by a crossover rate.



3. **Mutation** – process of change, insight an individual chromosome. As well as crossover mutation is controlled by a mutation rate.

The algorithms stops after a certain number of such iterations.


Generate 50

Selection
Type: Tournament

Crossover
Type: Two point crossover
Rate:

Mutation
Rate: 10

Save the best: ☑

Start Stop

Iterations: 200

Fitness: 2503.3031874281896

Result

During the given project timeframes the above web tool was created using just Java Script and HTML. The tool is divided into three main sections:

Control bar – allows to generate random points and start/stop the algorithm execution. It also contains all the mutable values which are available for users for experiments, such as: selection type, crossover type, crossover rate, mutation rate, possibility to save the best result after each iteration.

Path view – shows the best result after each iteration. As well as the number of iterations and fitness of the best result.

Burndown chart – shows the fitness change history of the best result. It saves the results of the previous runs as well, which enables to compare how good or bad different parameters can be.

Future Work

The tool gives a better chance to understand how the algorithm works. Yet in can still be extended with other mutable parameters such other selection or crossover types..

Contributions are welcome at https://github.com/ostap0207/web-ga-tsp

Live demo: http://ostap0207.github.io/web-ga-tsp/