Técnicas de Inteligencia Artificial

Chinese Postman Problem

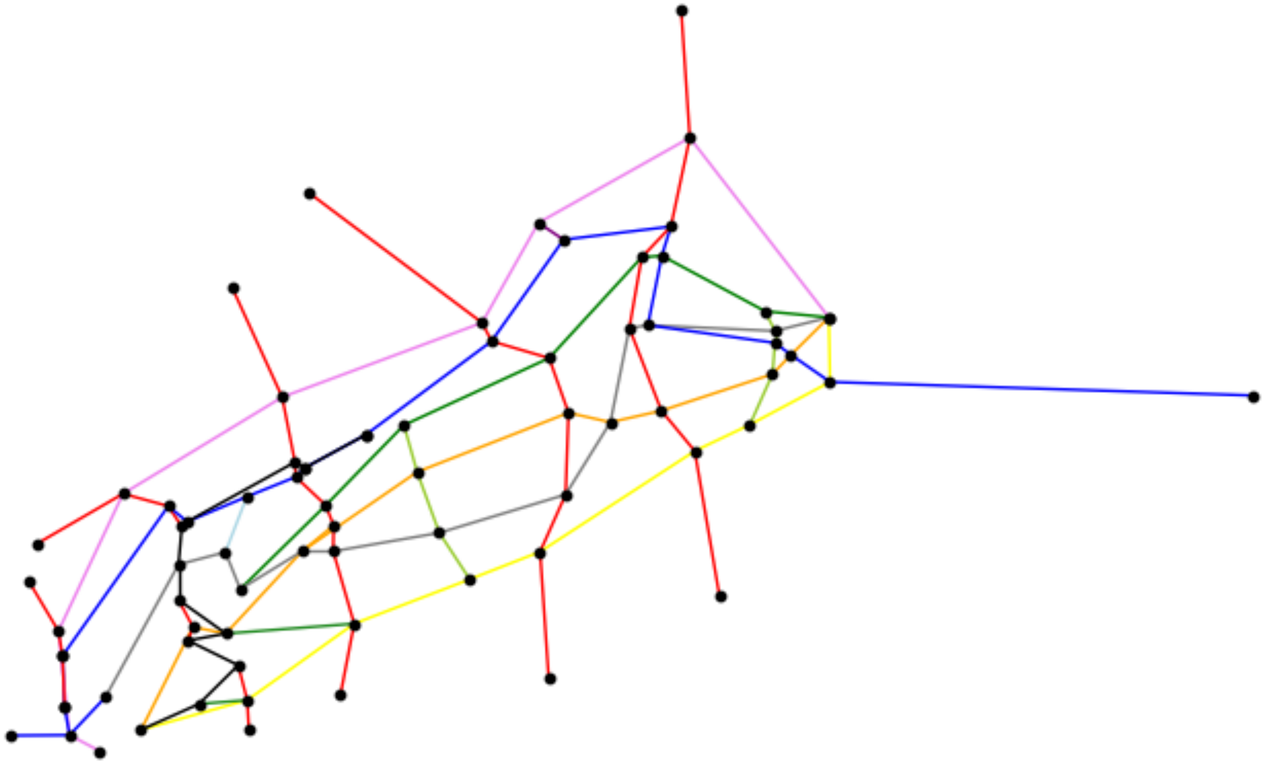
2021-2022

Adrián Vázquez Barrera

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Problem description

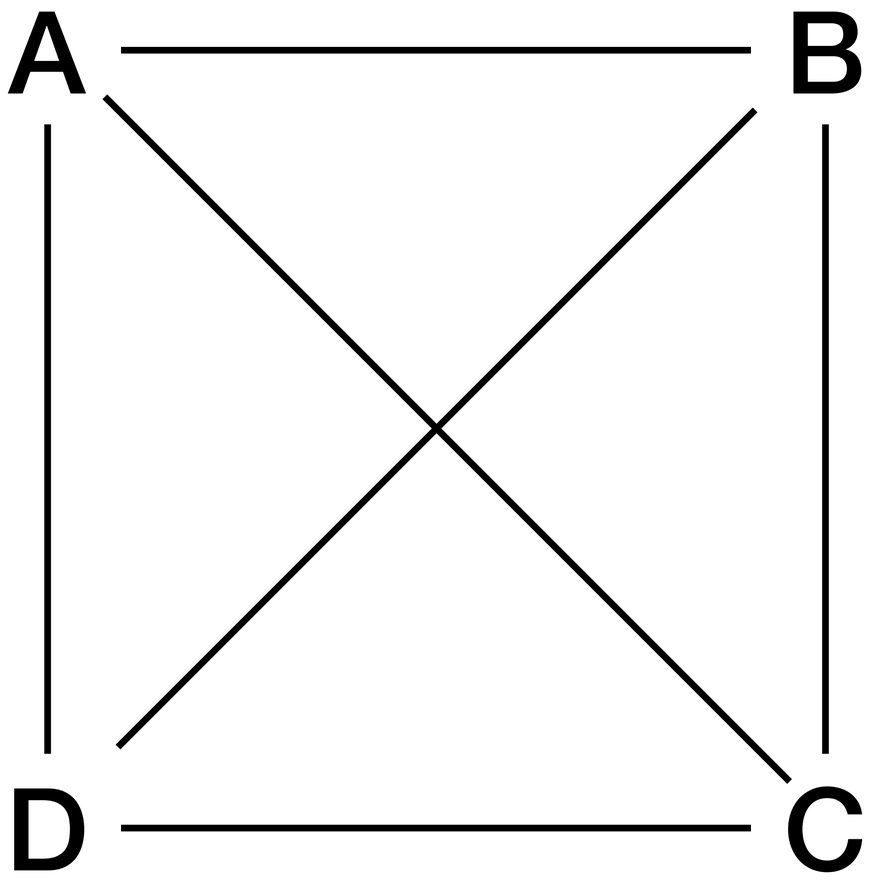
Chinese Postman Problem



http://brooksandrew.github.io/simpleblog/articles/intro-to-graph-optimization-solving-cpp/

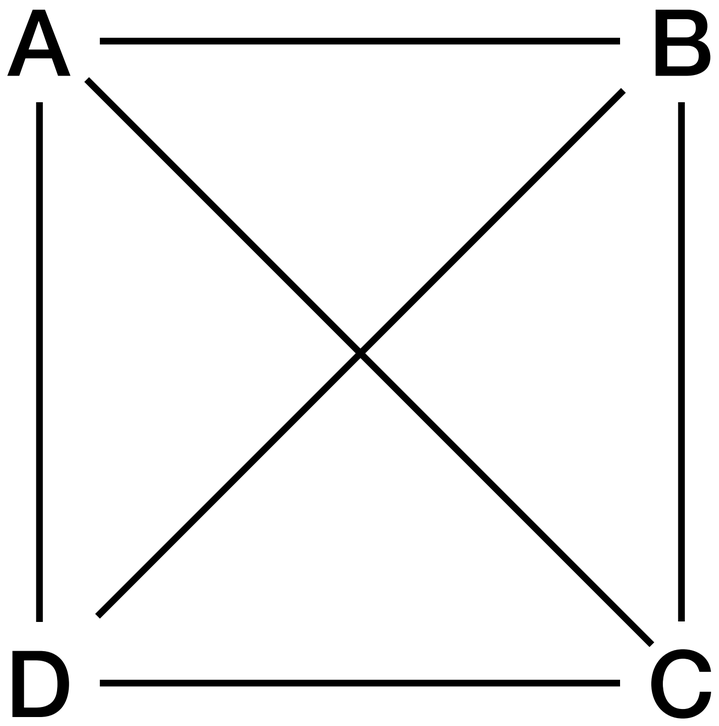
Problem description

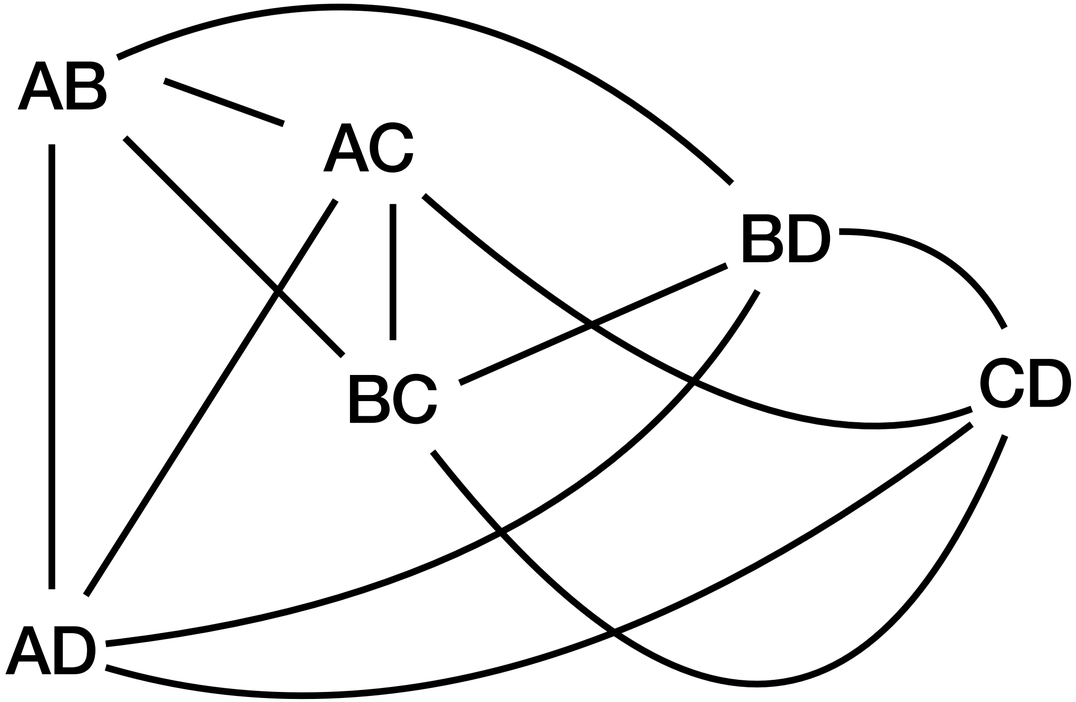
Chinese Postman Problem



Problem description

Chinese Postman Problem





Genetic Algorithm

Chinese Postman Problem

$$A = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 & \dots & d_{1n}^2 \\ d_{21}^2 & 0 & d_{23}^2 & \dots & d_{2n}^2 \\ d_{31}^2 & d_{32}^2 & 0 & \dots & d_{3n}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & d_{n3}^2 & \dots & 0 \end{bmatrix}$$

$$\mathrm{G} = [\mathrm{x}_0,\ \mathrm{x}_1,\ \mathrm{x}_2,\ \ldots,\ \mathrm{x}_{\mathrm{n}\text{-}1}]$$

A*

Genetic Algorithm

Chinese Postman Problem

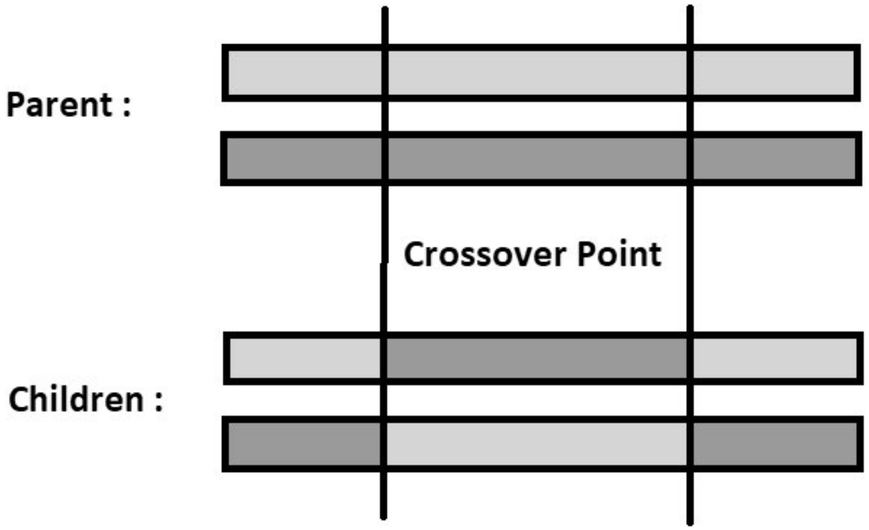
$$f(x) = dist(y_{n-1}, y_0) + \sum_{i=1}^{n} dist(x_i, x_{i-1})$$

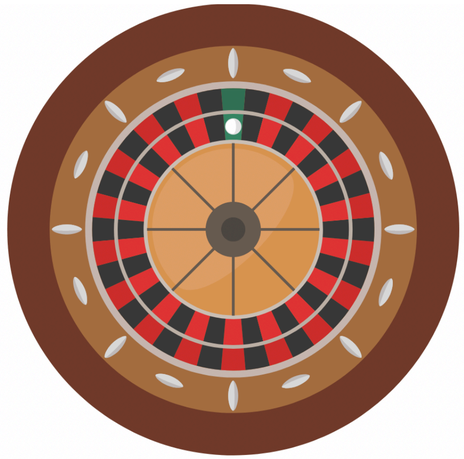
$$A = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 & \dots & d_{1n}^2 \\ d_{21}^2 & 0 & d_{23}^2 & \dots & d_{2n}^2 \\ d_{31}^2 & d_{32}^2 & 0 & \dots & d_{3n}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & d_{n3}^2 & \dots & 0 \end{bmatrix}$$

$$G = [x_0, x_1, x_2, \dots, x_{n-1}]$$

Genetic Algorithm

Chinese Postman Problem





Keep best 30%

Simulated Annealing

Chinese Postman Problem

$$G = [x_0,\ x_{n-1},\ x_2,\ \ldots,\ x_1]$$

Neighbourhood generated by swapping!

Memetic Algorithm

Chinese Postman Problem

```
public static final int MAX_ITER = 2000;
public static final int POP_SIZE = 10;
public static final int MAIN_CITY = 1;
public static final int MEME_POP_RATE = 50;
public static final double TEMPERATURE = 100;
public static final double COOLING_RATE = 0.9;
public static final double MUTATE_RATE = 0.30;
public static final double REPLACE_RATE = 0.30;
public static final int COMPACT_RATE = 5;
```

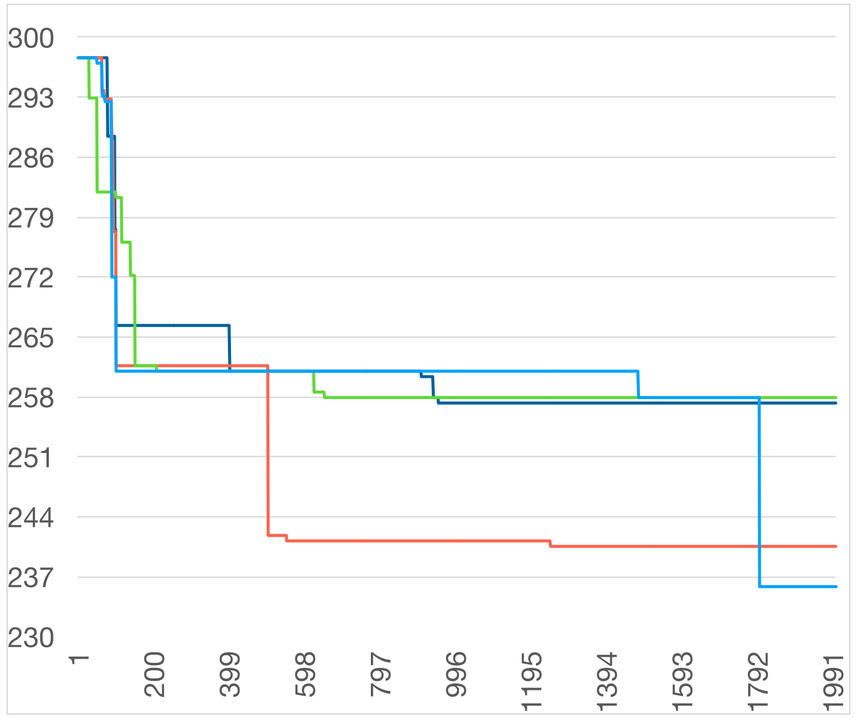
Comparison

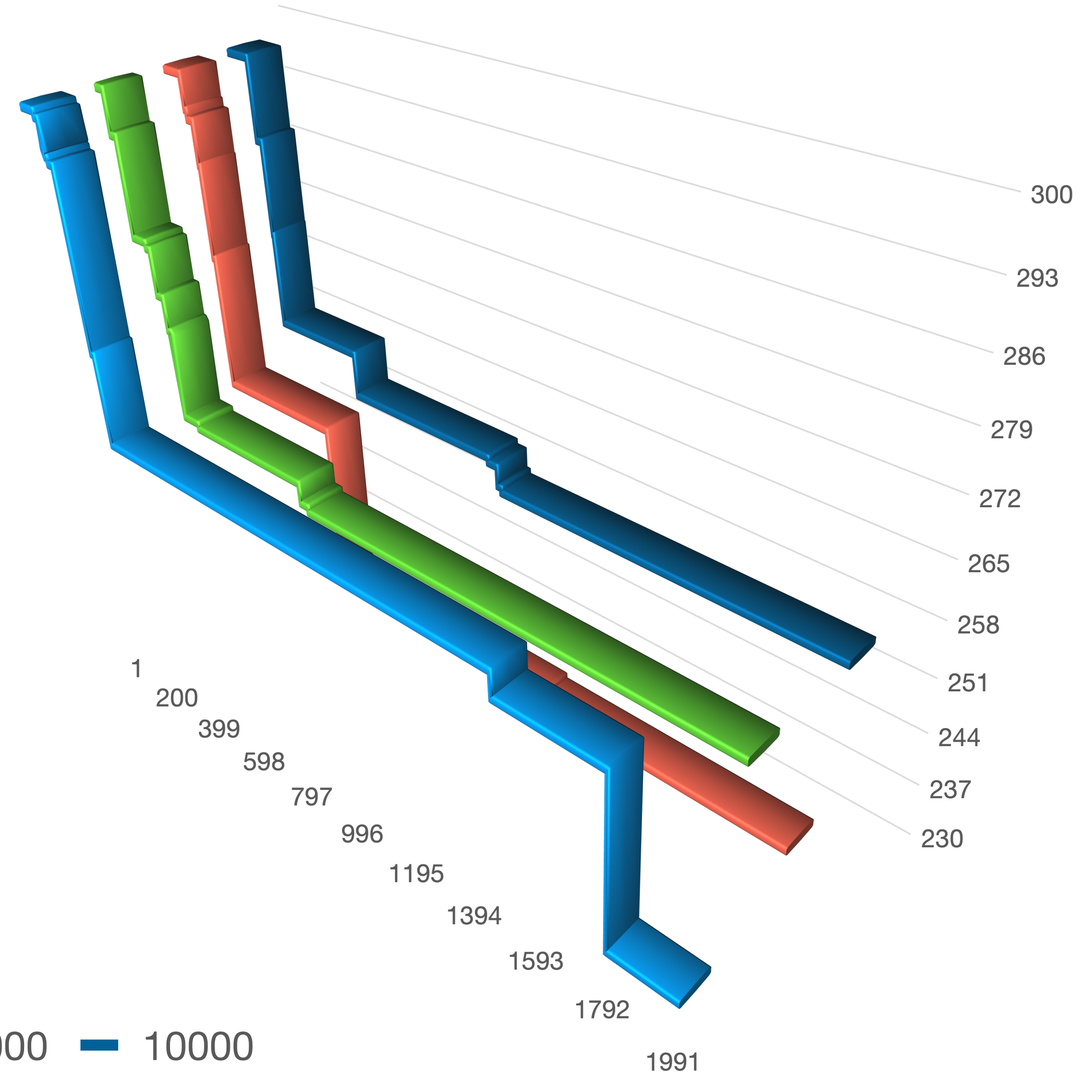
Chinese Postman Problem

2,000 iter. *77 edges*	Greedy	Simulated Annealing	Genetic Algorithm	Memetic Algorithm
Best fitness	350.47	287.60	236.48	235.20
Fitness Calls	1	250,800	9,127,701	29,224,069
Computation time	10 ms	2.3 s	4 min.	6 min.

Metrics

Chinese Postman Problem





Implementation

Chinese Postman Problem

Implementation

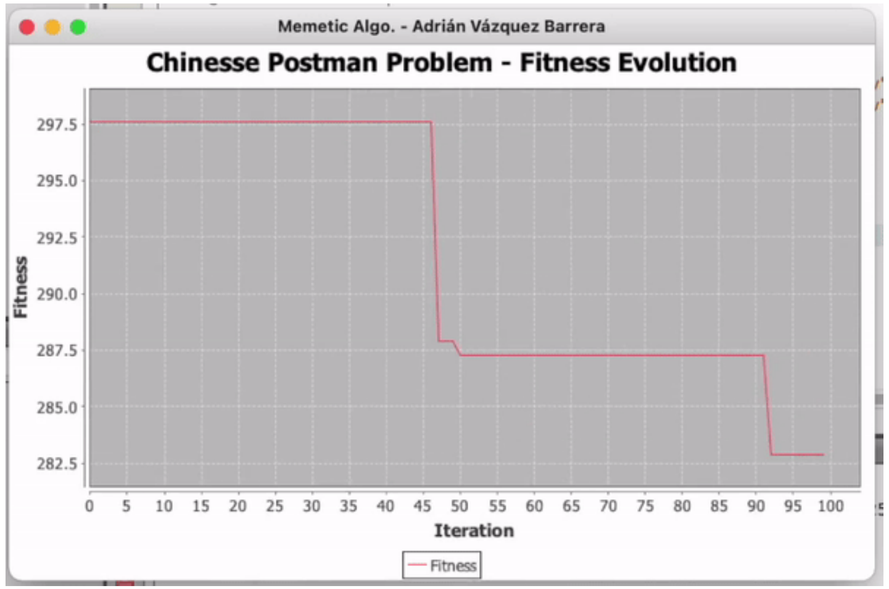
Chinese Postman Problem

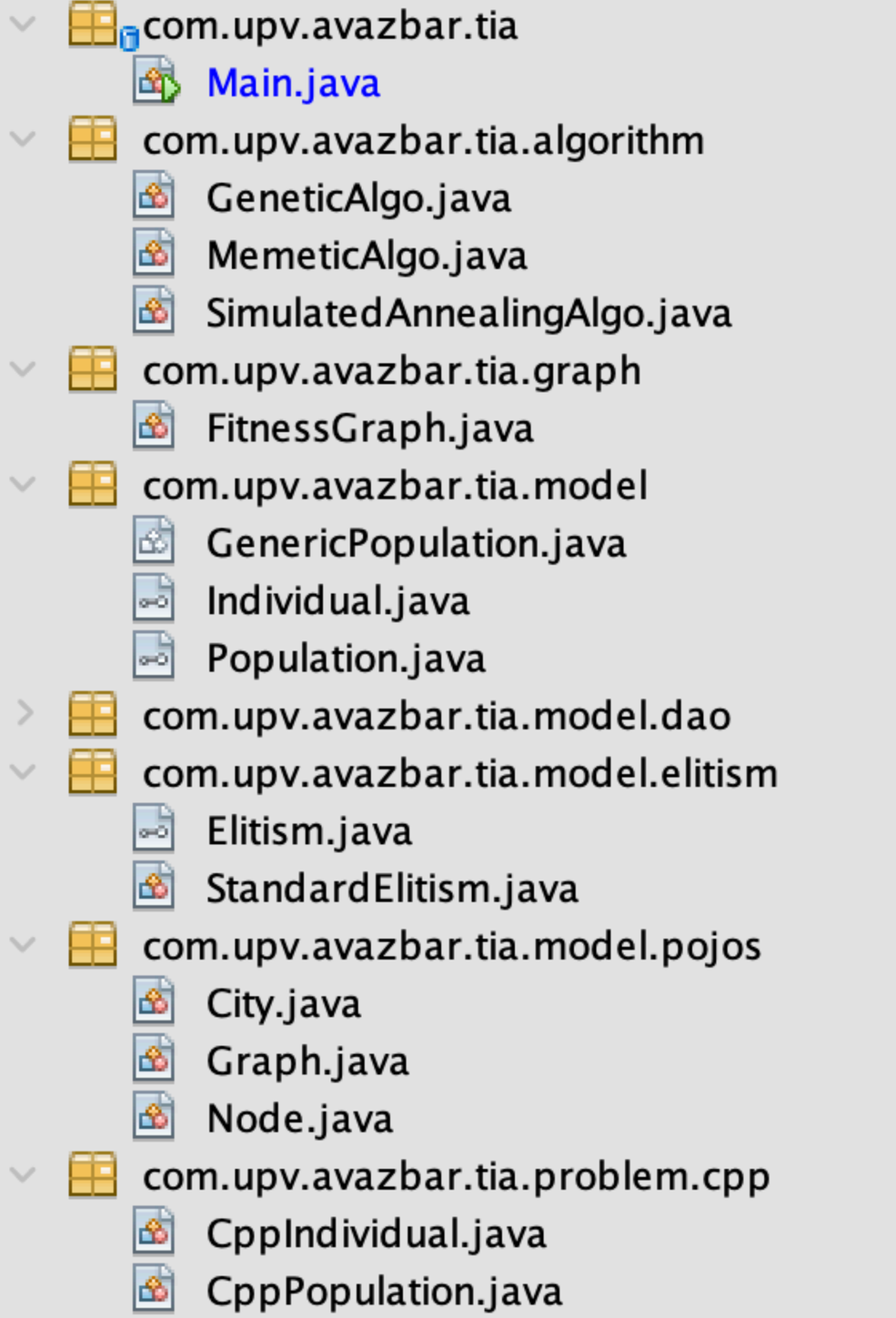
- com.upv.avazbar.tia
  - Main.java
- com.upv.avazbar.tia.algorithm
  - GeneticAlgo.java
  - MemeticAlgo.java
  - SimulatedAnnealingAlgo.java
- com.upv.avazbar.tia.graph
  - FitnessGraph.java
- com.upv.avazbar.tia.model
  - GenericPopulation.java
  - Individual.java
  - Population.java
- com.upv.avazbar.tia.model.dao
- com.upv.avazbar.tia.model.elitism
  - Elitism.java
  - StandardElitism.java
- com.upv.avazbar.tia.model.pojos
  - City.java
  - Graph.java
  - Node.java
- com.upv.avazbar.tia.problem.cpp
  - CppIndividual.java
  - CppPopulation.java

Conclusions

Chinese Postman Problem

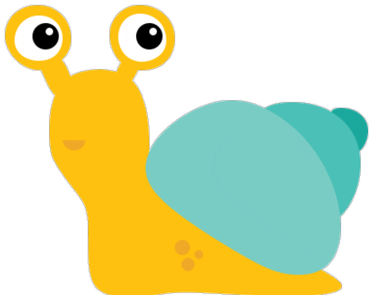
```
public static final int MAX_ITER = 2000;
public static final int POP_SIZE = 10;
public static final int MAIN_CITY = 1;
public static final int MEME_POP_RATE = 50;
public static final double TEMPERATURE = 100;
public static final double COOLING_RATE = 0.9;
public static final double MUTATE_RATE = 0.30;
public static final double REPLACE_RATE = 0.30;
public static final int COMPACT_RATE = 5;
```





Conclusions

Chinese Postman Problem



Técnicas de Inteligencia Artificial

Chinese Postman Problem

2021-2022

Adrián Vázquez Barrera

UNIVERSITAT POLITÈCNICA DE VALÈNCIA