

SIMONE QUADRELLI



GENETIC ALGORITHM MONTE CARLO

Academic year 2019 - 2020

Contents

1	Introduction	1
2	Simulated annealing	1
3	Metropolis algorithm	1
4	Genetic Algotirhm	1
5	dataset	1
6	results	1

List of Tables

List of Figures

Abstract

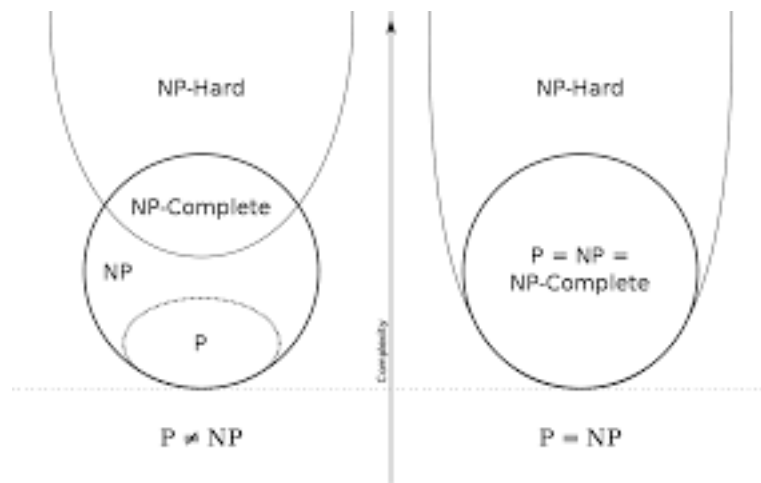
The aim of this project is to solve the travelling salesman problem (TSP) which consists in finding the shortest cycle among all the cities. The problem is a *NP-hard* problem and therefore there exist no feasible algorithm to solve it for any possible set of cities in input. The approximately optimal solution was computed by the simulated annealing, while the possible cycles are produced by a genetic algorithm.

1 Introduction

The travelling salesman problem consists in finding the shortest route a travelling salesman has to take to visit all the city he must reach only once and to return to the starting point. The route described is called in graph theory *hamiltonian cycle* (i.e a cycle that pass through each vertex of a graph just once), indeed the oldest and more technical formulation of the problem was proposed by Hamilton.

An algorithm that provides a solution to TSP can be exploited in a wide variety of applications: indeed it can be used in logistics to optimize the path of vehicles and therefore to reduce costs. It is possible to model the problem using an undirected, weighted and fully connected graph ¹ $G = (N, E)$, where N is the set of the cities to visit and $E = N \times N$ is the weight of the path. In this particular instance of the problem the weights correspond to the euclidean between each couple of cities. The hamiltonian cycle can be defined as a sequence (n_0, \dots, n_{k-1}) of length k such that $n_i \neq n_j \forall i, j \in \{1, \dots, |N|\}$ and such that $(i, j) \in E \forall i, j \in \{1, \dots, |N|\}$. The most naive solution is to compute all the possible paths but they are factorial in the number of cities n in input. The number of possible cycles is $(n - 1)!$ and grows faster than 2^n (i.e $2^n = o(n!)$). Therefore even if it was possible to find them in constant time the algorithm will have an exponential time complexity and would not be solved in finite time for large n . More technically, TSP is an NP-hard problem. A NP-hard problem is a problem at least as hard as the hardest NP-complete problem. NP-complete problem is a problem that can be solved in polynomial time by a non-deterministic algorithm. For such a class of problem there exists no polynomial algorithm that can solve the problems and may not exist, still no one was able to prove that such algorithms do not exist. Bearing in mind the complexity problem to face, the most reasonable solution is to solve the problem approximately. The technique I used to generate the possible hamiltonian cycles is a genetic algorithm, while the choice of the fittest hamiltonian cycle is done by the simulated annealing algorithm

¹For an hamiltonian cycle to exist the graph must be at least connected



- 2 Simulated annealing
- 3 Metropolis algorithm
- 4 Genetic Algorithm
- 5 dataset
- 6 results