# Automated algorithm selection

**Automated algorithm selection**
- I. Introduction
- II. Preliminaries
  - (Multiple) Travelling Salesman Problem
  - Automated Algorithm Selection
- III. State-of-the-art
- IV. Algorithms in the portfolio
  - Genetic Algorithm
- Bibliography

## I. Introduction

With the rapid development of fields such as machine learning, artificial intelligence and optimization, a multitude of algorithms and techniques are being developed, tested and improved. Therefore, even faced with simple problems, for which the scientific community has found satisfying solutions and approaches, better yet techniques can be found for reaching the end goal. Automated algorithm selection comes in handy for choosing the best approach for solving the given problem, tasking into consideration the instances, data and requirements of the solver.

For the scope of the current project, we will focus on the well-known *Travelling Salesman Problem (TSP)*, a classical optimization problem, with the end goal of finding the shortest possible route that visits a set of locations. By the end of this paper, we will have stated the *state-of-the-art* in automated algorithm selection and present two illustrative methods for achieving the end-goal of the study.

## II. Preliminaries

### (Multiple) Travelling Salesman Problem

The *travelling salesman problem (TSP)* 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?". From a graph theory standpoint, it can be modelled as an *undirected weighted graph*, where the vertices correspond to the cities, the paths are the edges and the path's distance is the edge's weight. A variation of TSP (asymmetric), considers that the paths between two cities may either not exist, or might differ, therefore, constructing a *directed weighted graph*. [12]

The Multiple Traveling Salesman Problem (MTSP) is a generalization of the well-known Traveling Salesman Problem (TSP), where multiple salesmen are involved to visit a given number of cities exactly once and return to the initial position with the minimum traveling cost.[11]

### Automated Algorithm Selection

Automated Algorithm Selection (AAS) aims at automatically selecting a solution algorithm from this portfolio. Formally, this can be expressed as an optimization problem as such:

> Given a set of instances $I$ of problem $P$ , a set $A$ = {$A1$,...,$An$} of algorithms that solve $P$ , and a metric $m : A \times I \rightarrow$ R that measures the performance of any algorithm $Aj \in A$ over the set of instances $I$, the goal is to build a selector $S$ that maps any instance $i \in I$ to an algorithm $S(i) \in A$ such that the average performance of $S$ over $I$ is optimal according to the metric $m$.[1] [10]

# III. State-of-the-art

For efficiently solving the Travelling Salesman Problem, multiple algorithms techniques can be used, taking into consideration the given instance or the requirements to be met.

1. **Exact algorithms**, such as branch-and-bound and linear programming:  suitable for relatively small instances, as these algorithms explore all the possible solutions, therefore guaranteeing the optimal solution.
2. **Evolutionary algorithms**: suitable for accommodating constraints and different optimization objectives; however, by their nature, evolutionary algorithms can not guarantee a global optimum.
3. **Reinforcement learning**: has shown promising results, especially when combined with techniques such as deep neural networks or Monte Carlo Tree Search; however, it is computationally expensive and requires a significant amount of training data, making it less practical for some instances of TSP. [2]

Taking into consideration some of the algorithms present in the literature, we can assembly the following (running time-wise) comparison:

| **Algorithm** | **Computational Complexity / Running time (s)** | **Performance** |
|---|---|---|
| Concorde TSP Solver (exact algorithm) [5] | *berlin52* instance - 0.29 \|\| *eil101* instance - 0.74 [3] | Provides optimal solutions, but is computationally expensive for (some) large problem instances (see [4]) |
| (Modified) Lin-Kernighan Heuristic [6] | *berlin52* instance - 0.1 \|\| *eil101* instance - 0.2 [6] | Provides high-quality solutions in reasonable time for small to medium-sized problem instances |
| Genetic Algorithm with edge assembly crossover [7] | instance *vangogh120K* (300 pop, 30 off-springs) - 10 s [7] | Provides good quality solutions for large problem instances |

| Algorithm | Computational Complexity / Running time (s) | Performance |
|---|---|---|
| Ant colony optimization based on parameters optimization (SOS–ACO) [8] | - | Provides good quality solutions for large problem instances. For example, on *eil51* instance it produces an error of 0.49% relative to the best known solution. |

Regarding the automated selection of the algorithms, in *Improving the state-of-the-art in the Traveling Salesman Problem: An Anytime Automatic Algorithm Selection* [1], the authors run tests on multiple datasets (such as *TSPLIB,TNM and NATIONAL*), using *Concorde*[][][3], *Lin Kernigham Helsgaun (LKH)*[6], *Edge Assembly Crossover (EAX)*[7] and *Multiagent Optimization System (MAOS)*[9].

The modus operandi for the automated algorithm selection consists of creating a *metasolver*, an *Anytime Automatic Algorithm Selection model*, having a portfolio consisting of the five above-mentioned algorithms. The input, consisting of the combinations of the TSP instances and the collection of solvers, is fed to a CNN model. The output corresponds to a ranking representation of the cost achieved by each solver over a fixed number of steps (here 500).

Results of this model are promising, as experimental tests show that the model obtains an accuracy of 79.8 % (better than running the five algorithms separately); moreover, the time required for predicting the best solver is drastically reduced in comparison to other traditional feature selection and machine learning methods.

Having overviewed the state-of-the-art in the field of automated algorithm selection, with applications in TSP, we will present two methods: genetic algorithms and constraint programming (using CPLEX).

# IV. Algorithms in the portfolio

## Genetic Algorithm

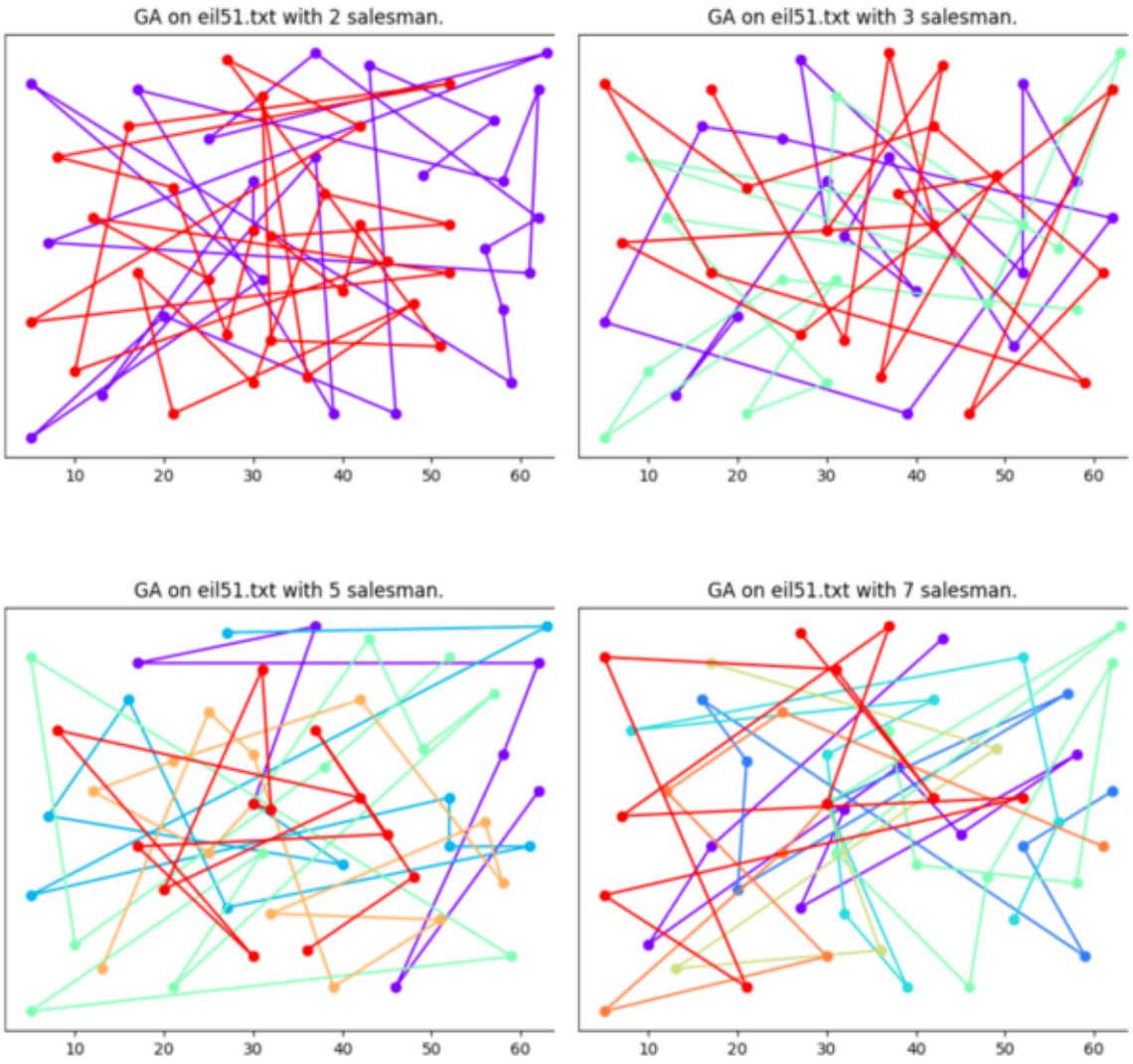
The genetic algorithm we implemented for multiple TSP and the corresponding test configuration have the following specifics:

- A population of 100 individuals
- 100 iterations
- 0.7 mutation probability
- 0.7 crossover probability
- the genome of an individual is composed of the (possible) tours for each (k) salesmen;
- the number of salesmen was 2, 3, 5 or 7.

The results of the tests are shown in the following table:

For **eil51** instance (TSPLIB), with 51 cities:

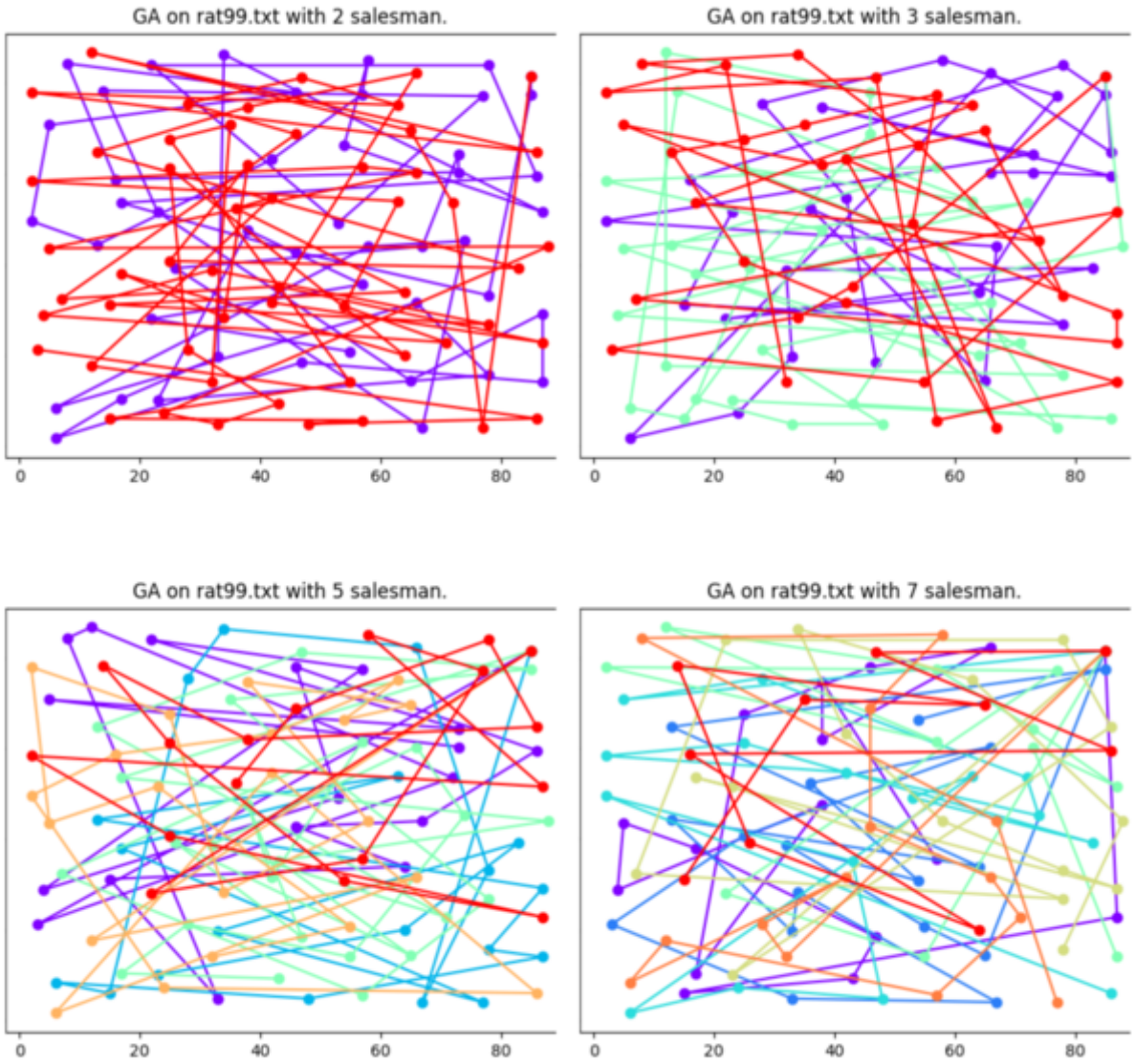
| | Cost | Best Known Solution (CPLEX)[13] | Running Time(s) |
|---|---|---|---|
| m = 2 (salesmen) | 705.50 | 442.32 | 4.15 |
| m = 3 | 636.80 | 464.11 | 3.86 |
| m = 5 | 864.90 | 529.70 | 3.77 |
| m = 7 | 911.82 | 605.21 | 3.94 |




For **rat99** instance (TSPLIB), with 99 cities:

| | Cost | Best Known Solution (CPLEX)[13] | Running Time(s) |
|---|---|---|---|
| m = 2 (salesmen) | 5467.78 | 1350.73 | 7.45 |

| | Cost | Best Known Solution (CPLEX)[13] | Running Time(s) |
|---|---|---|---|
| m = 3 | 5990.79 | 1519.49 | 6.53 |
| m = 5 | 6501.02 | 1855.83 | 6.35 |
| m = 7 | 6376.16 | 2291.82 | 6.56 |




## Bibliography

- [1] Isaías I. Huerta, Daniel A. Neira, Daniel A. Ortega, Vicente Varas, Julio Godoy, Roberto Asín-Achá,
  Improving the state-of-the-art in the Traveling Salesman Problem: An Anytime Automatic Algorithm Selection - ScienceDirect
- [2] Zhihao Xing, Shikui Tu, Lei Xu, Solve Traveling Salesman Problem by Monte Carlo Tree Search and Deep Neural Network
- [3] Concorde TSP Benchmarks (uwaterloo.ca)
- [4] Stefan Hougardy: Hard to Solve Instances of the Euclidean Traveling Salesman Problem (uni-bonn.de)
- [5] Concorde Home (uwaterloo.ca)
- [6] Keld Helsgaun, An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic
- [7] Kazuma Honda, Yuichi Nagata, Isao Ono, A parallel genetic algorithm with edge assembly crossover for 100,000-city scale TSPs (uwaterloo.ca)
- [8] Yong Wang, Zunpu Han, Ant colony optimization for traveling salesman problem based on parameters optimization
- [9] X. Xie, J. Liu, Multiagent Optimization System for Solving the Traveling Salesman Problem (TSP)
- [10] J. R. Rice et al., The Algorithm Selection Problem (purdue.edu)
- [11] O. Cheikhrouhou, I. Khoufi, A Comprehensive Survey on the Multiple Travelling Salesman Problem: Applications, Approaches and Taxonomy (arxiv.org)
- [12] Travelling salesman problem - Wikipedia
- [13] Comparison of the Best Know Solutions (BKS) and PCI-algorithm. | Download Scientific Diagram (researchgate.net)
- [14] Masoumeh Vali , Khodakaram Salimifard, A Constraint Programming Approach for Solving Multiple Traveling Salesman Problem