

# IFT6751: Homework 3

Gabriel C-Parent

April 1, 2015

## 1 Introduction

In this homework, an Elastic Net algorithm is used to solve the traveling salesperson problem (TSP). Using different approaches to solve the capacitated vehicle routing problem (CVRP) were designed. The first one is a genetic algorithm using specialized crossover and mutation operators whereas the second one uses the Tabu Search to control local search with the  $\lambda$ -interchange neighbourhood.

Two greedy local search methods are used within both methods to improve the results such as the 2-opt descent and the  $\lambda$ -interchange descent.

Another method based on the Clarke & Wright savings is also used to initialize solutions for both metaheuristics.

A description of all the optimization methods along with special implementation details is given. Some experimental results are then compared based on running time, implementation complexity and results quality.

Finally, a user guide is given in the supplementary section 6.1.

## 2 Local Search Methods

### 2.1 2-opt descent

First, a simple and fast optimization method for the TSP was needed to improve the path of each routes.

The local search method used to optimize individual routes is the steepest improvement method as described in [1]. Basically, it is an implementation of steepest descent implementation using the well known 2-opt operator.

At each iteration, the best possible 2-opt is chosen, according to the reduction in total distance, until there isn't any possible improvement. The complexity of the procedure  $O(n^2)$  on the number of edges in the route.

Although it might seem slow, usually the number of edges is quite small and the time spent optimizing routes is negligible.

### 3 Experimental Results

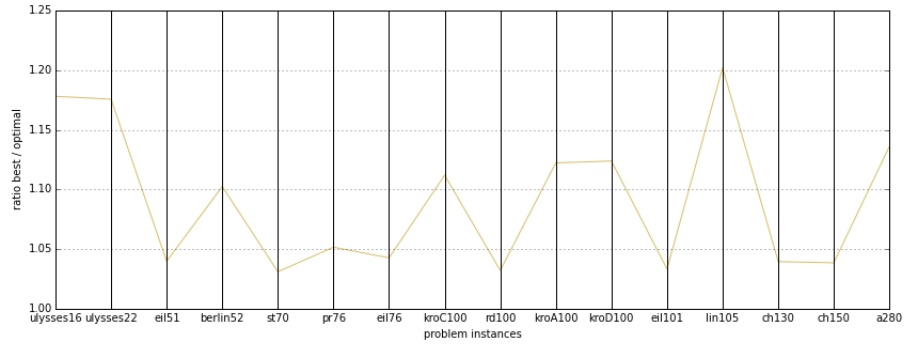


Figure 1: Ratio of obtained solution against optimal known solution. The problems are arranged by the number of clients to visit in increasing order from left to right. We can see that the number of clients doesn't seem to influence the relative performance.

## **4 Discussion**

## **5 Conclusion**

## **References**

- [1] Áslaug Sóley Bjarnadóttir. Solving the vehicle routing problem with genetic algorithms. Master's thesis, Technical University of Denmark, 2004.

## 6 Supplementary Materials

### 6.1 User Guide

The following section should help with verification of the results and repeatability.

The language used is a mix of python and cython, an optimising compiler that allows static typing and generates C code.

#### 6.1.1 Working Environment

All computational results obtained in this work should be repeatable given a suitable python environment. The particular dependencies of this work are the Cython, Numpy, IPython and Seaborn along with standard python environment.

The following python environment was used:

```
CPython 2.7.9
```

```
ipython 2.2.0
```

```
numpy 1.9.2
```

```
cython 0.21
```

```
ipython 2.2.0
```

```
seaborn 0.5.1
```

```
compiler : GCC 4.4.7 20120313 (Red Hat 4.4.7-1)
```

```
system : Linux
```

```
release : 3.13.0-46-generic
```

```
machine : x86_64
```

```
processor : x86_64
```

```
CPU cores : 4
```

```
interpreter: 64bit
```

As of now, my personal recommendation is to use the excellent Anaconda python distribution from Continuum Analytics.

#### 6.1.2 Running Computational Results

All computational results and figures are contained in the form of IPython Notebooks with the hope of allowing repeatability and reproducibility.

If IPython is available on the computer, an IPython Notebook service can be launched from command line using the following call.

This allows viewing and recomputation of the results.

Alternatively, the IPython notebooks can be viewed online if it is reachable from a url using the nbviewer tool. This allows viewing a static version of an IPython Notebook.

## 6.2 Experimental Solutions

For each of the figures following, the net, the corresponding path and the best path are arranged in lines.

