

# AI Cup 2020 report

Stefano Taillefert  
Artificial Intelligence

December 22, 2020

## 1 Introduction

To solve this problem, I chose to use the Nearest-Neighbour algorithm. More precisely, if no parameter is specified (see `README.md`), the best Nearest-Neighbour. The difference is that the latter will check all the nodes to start from and pick the best one (re: the one that results in the shortest path).

While it is pretty simple to implement, the main drawback is that it doesn't always produce the best possible path, since it is a greedy algorithm. To compensate this, it is a common practice to post-process the solution with an optimizer, like the `2opt` one we saw in class.

Unfortunately, I couldn't manage to make the optimizer work, hence why my error rate is so high.

## 2 Language used

I considered three options for this project: Python, C and Java. Since I am pretty rusty with both the first ones, I thought Java might be a good compromise between the blazing-fast C and the super-easy Python (which I don't personally like much, but that is beyond the point). I implemented a wrapper class (`Main`) to process the input file and call the solver (and theoretically also the optimizer), so that it should be easy to extend the program with other solvers or optimizers.

## 3 Results

Due to the lack of optimizer, my solutions are the raw output of the best-NN algorithm. As mentioned before, those are far from being optimal. I am aware of and apologize for the terrible quality of this submission, but I also got Covid literally the day after starting to work on it; I understand that it is not an excuse, but I am literally unable to work productively on anything as of now.