Our problem to solve was a modified version of the Travel Salesman Problem: additionally, we have costs associated with each node, and we had to select exactly 50% of the total nodes.

Pseudo codes:

Nearest neighbor

0. Calculate distance matrix
1. Add initial node to cycle
2. While length of current cycle is less than 100:

    2.1 Calculate difference between not yet visited node and last node
    2.2 Find NN
    2.3 Add this NN to cycle
    2.4 Add distance between last node and NN, and add also cost of this node to total cost
    2.5 Remove NN from not yet visited (set distance from this node to each other to infinity)

3. Add to total cost distance between first and last node in cycle

Greedy cycle

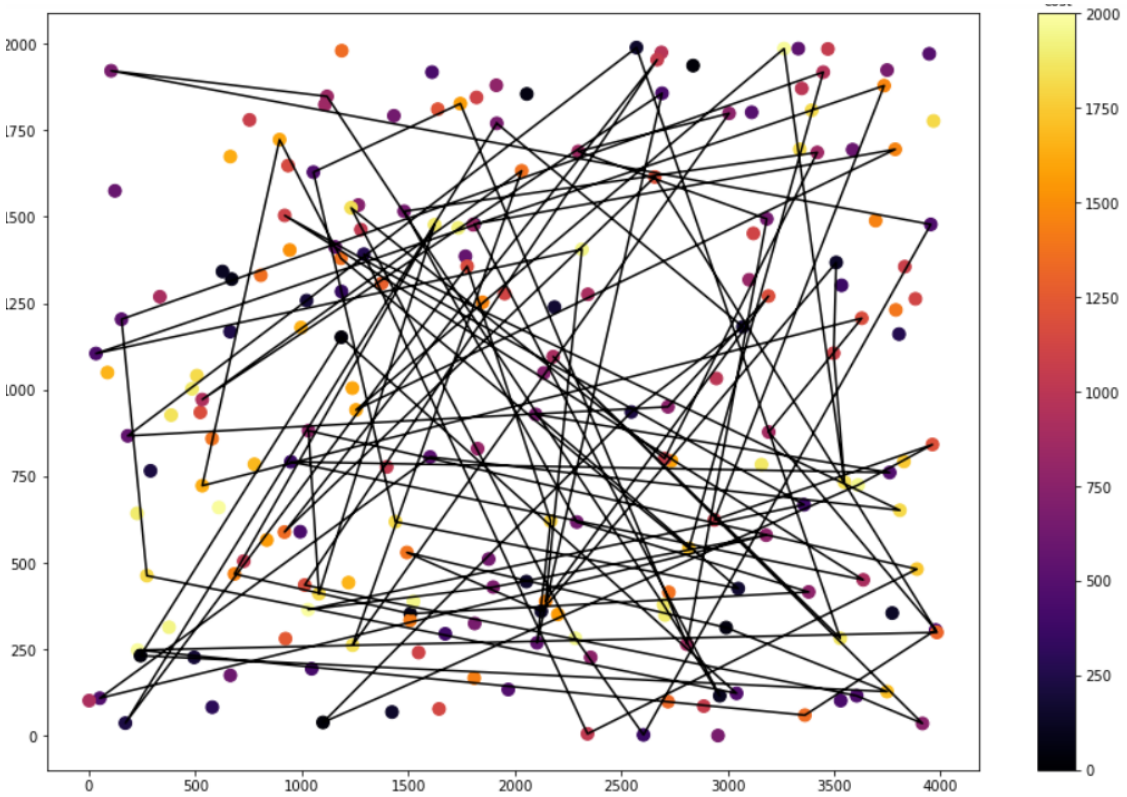
0. Calculate distance matrix between each pair of nodes, add to each distance avg cost of both nodes.
1. For the initial node find NN, and add edges between them(in both directions) to the output cycle.
2. While output cycle length is less then 100:
    2.1 For each edge(E_1, E_2) in output cycle:
        2.1.1 For each node(N) not yet in output cycle:
            2.1.1.1 Calculate change in distance(**edge**(E_1, N) + **edge**(N, E_2)
                - **edge**(E_1, E_2))
    2.2 Select this new node, for which above change is minimal
    2.3 Remove "useless" edge from cycle, and add 2 newones
    2.4 Remove newly find node from not yet in output cycle

Code: https://github.com/Imrauviel/Evolutionary-computation

Conclusion:
Nearest neighbor is very fast(almost instant after calculating distance matrix. Greedy cycle approche, despite being much slower, gives us much better solutions.

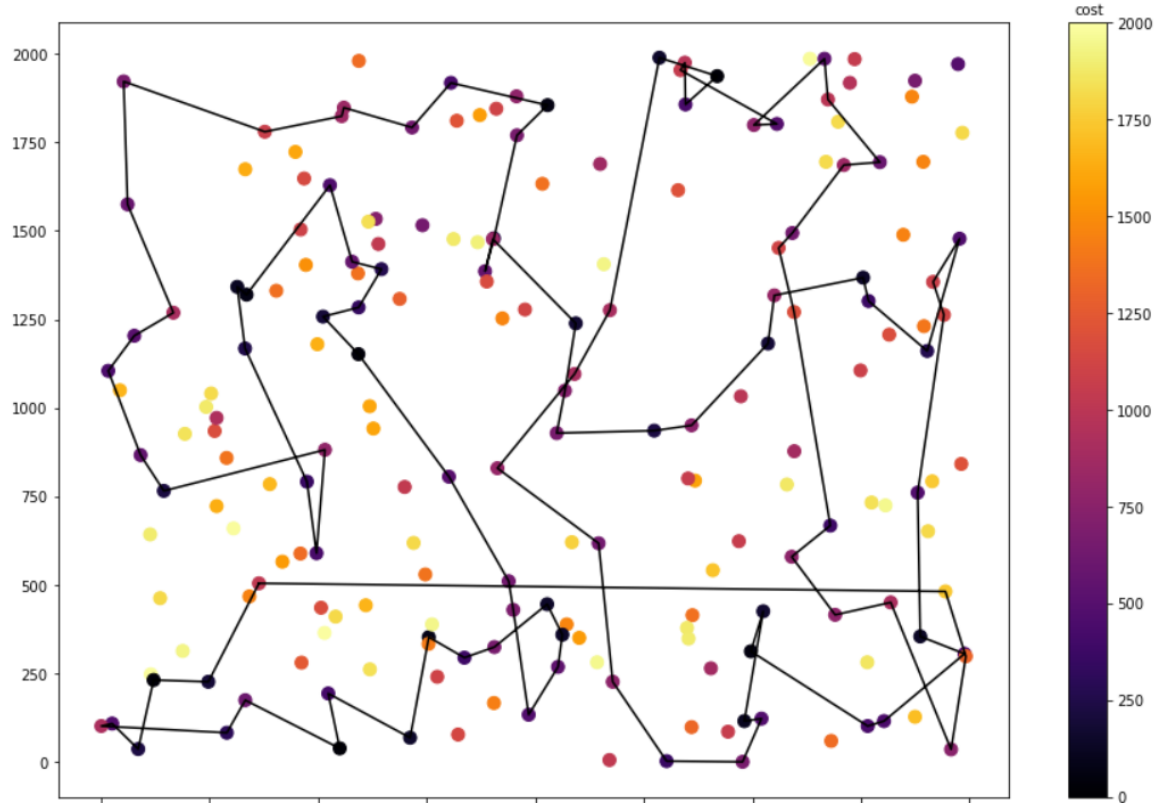
TSPA	Random	NN	Greedy Cycle
Min	244344	84149	75566
Max	291673	97583	79799
Mean	264423.555	89420.22	76961.72

TSPB	Random	NN	Greedy Cycle
Min	242446	74413	68655
Max	294115	93154	76230
Mean	267442.125	82446.675	70656.38



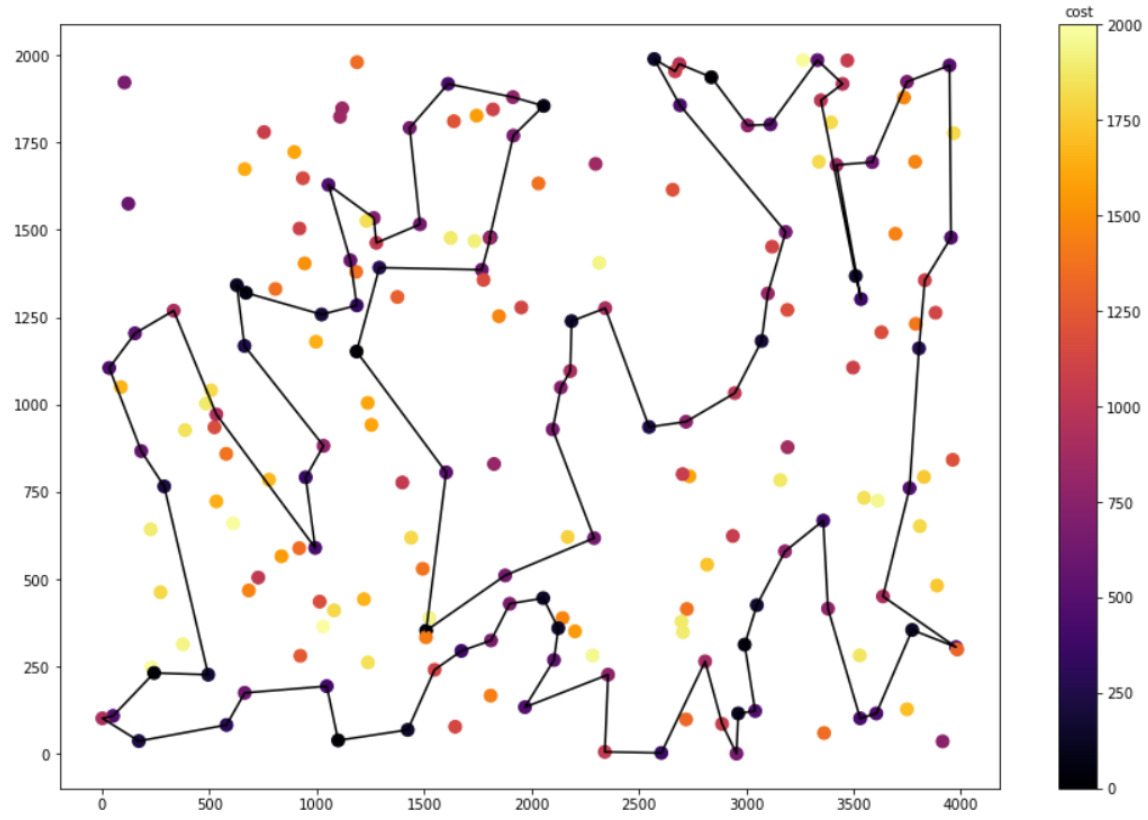
NN:

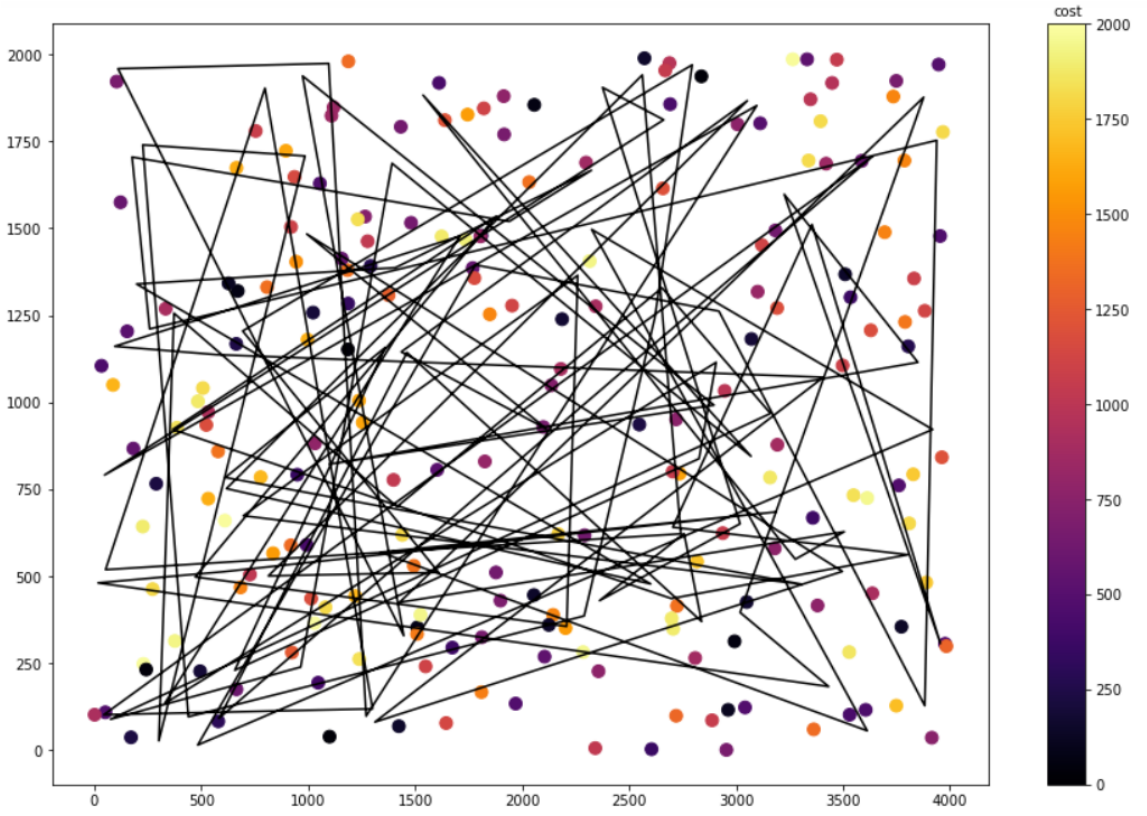
```
37
```



Greedy:

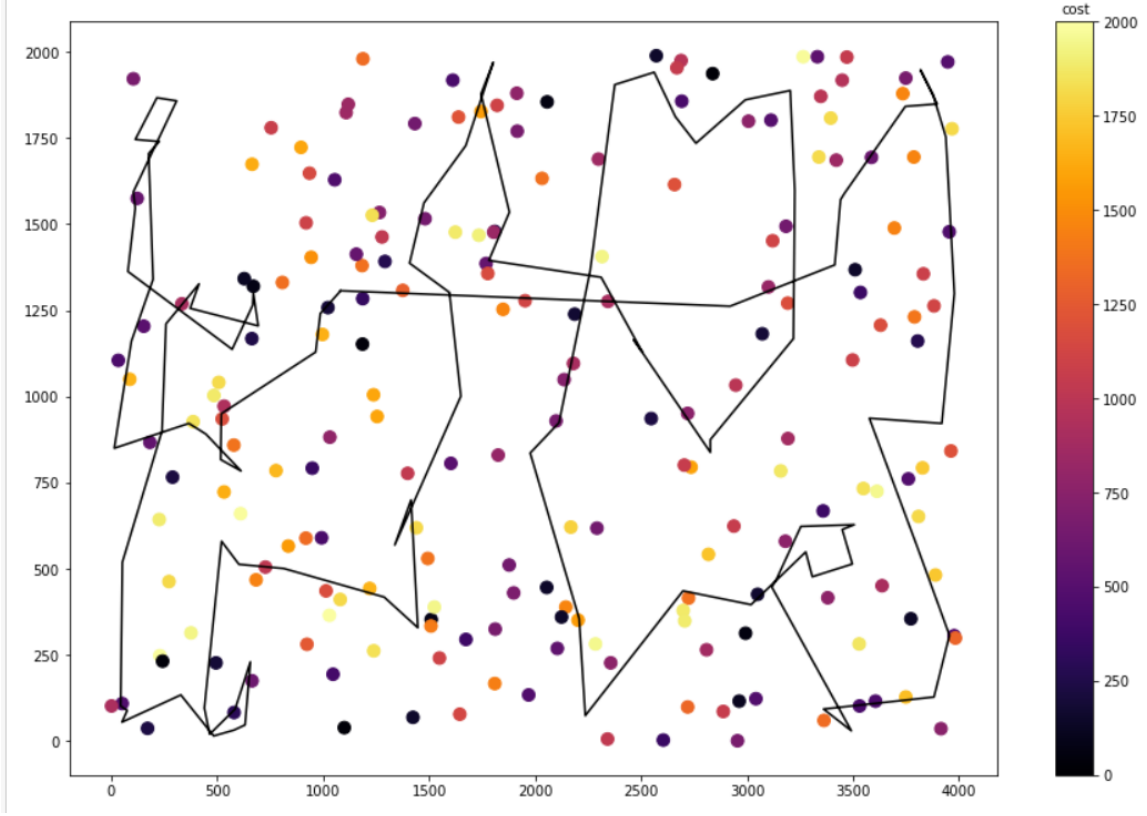
```
150
```




cost
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000
2500
3000
3500
4000

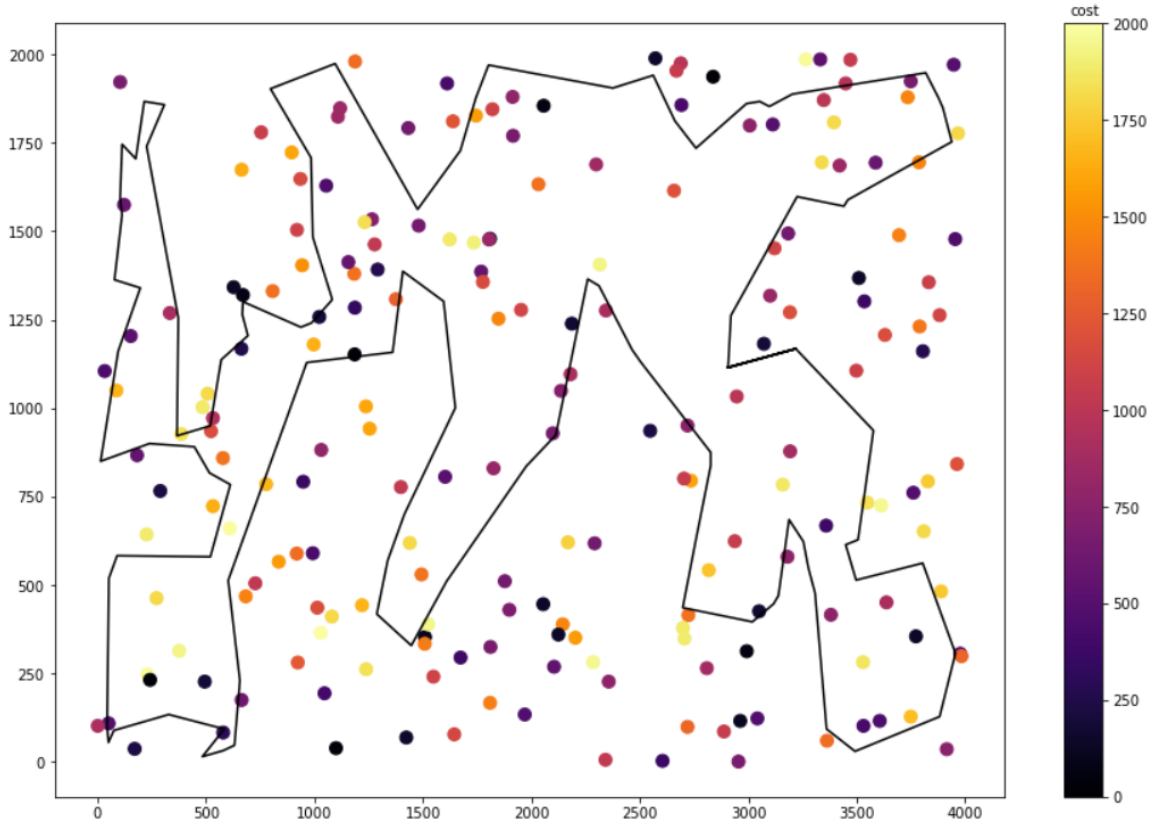
NN:

12


cost
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000
2500
3000
3500
4000

Greedy:

```
50
```


cost
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000
2500
3000
3500
4000