Local Search for TSP: Datasets, tests and results

Datasets

To test our Local search algorithms for TSP we have used datasets of different sizes. In particular, we have chosen to use both constellations of nodes, randomly generated by us, and real datasets such as the case of the cities of Italy, imported from TSPLIB.
We report below the results of the analysis made using datasets of three different sizes, made up of 100, 5000 and 16863 nodes.

Figure 1: 100 random nodes

Figure 2: Italy from TSPLIB

Before examining the results of the Local Search algorithms, let's see the quality of the initial solutions, previously discussed, obtained with each of the three datasets.

The table below shows the Nearest Neighbour solution and random initial solution, in terms of both total distance and computational time.

MAPS	# NODES	NN LENGHT	NN TIME	RANDOM LENGHT	RANDOM TIME
ITALY	16863	701404	100.96 s	70305161	0.09 s
RANDOM 1	100	1443	0.01 s	8436	< 0.01 s
RANDOM 2	5000	477613	11.57 s	19711844	0.03 s

Figure 3: Initial solutions of the three datasets

As we can see and as expected, the total length of the path of the Nearest Neighbour algorithm is highly much better than that one generated by the random algorithm.

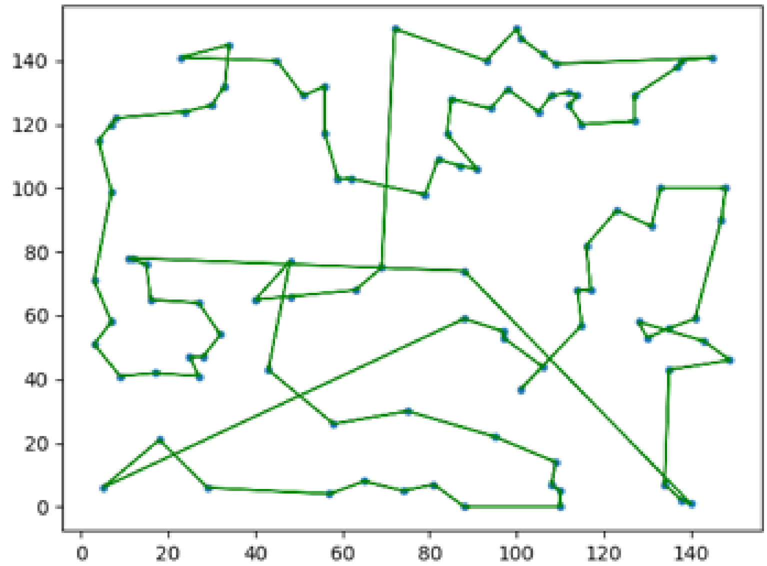

Figure 4: Initial solution of NN algorithm for 100 nodes constellation

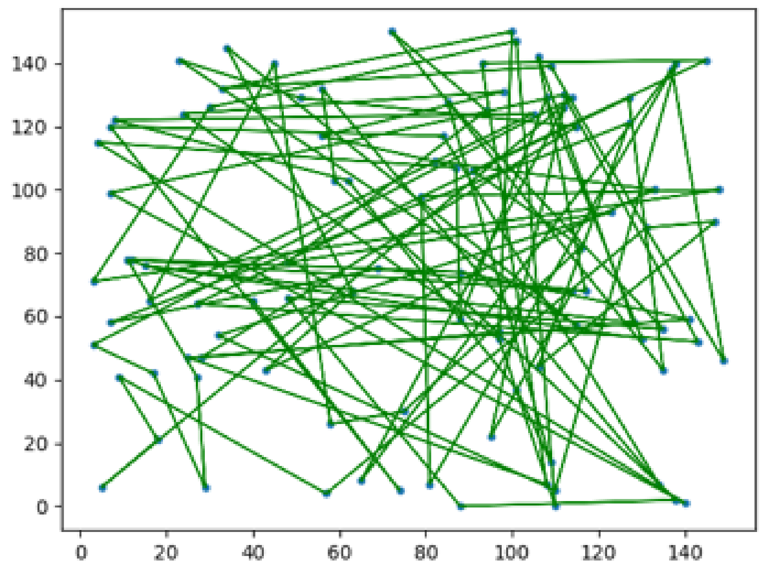

Figure 5: Initial solution of random algorithm for 100 nodes constellation

Test and results

We report here a general table of the results, for each of the three tested datasets, produced by the various Local Search algorithms implemented for TSP.

ALGO/NODE	100 NN	100 RND	5000 NN	5000 RND	16863 NN	16863 RND
Initial	1443/0.01s	8436/0.00s	477613/11.26s	19711844/0.02	701404/100s	70305161/0.09s
2Opt	1257/0.06s	1364/0.15s	413132/382s	434884/539s	604029/3943s	630746/5856s
2OptDLB	1257/0.06s	1358/0.14s	413132/378s	435010/512s	604336/2560s	630746/4912s
2OptPnearest 0.965	1288/0.06s	1534/0.13s	433524/343s	1124541/647s	623384/6406s	3061520/8537s
2OptRadius 1.2	1293/0.08s	1430/0.10s	419382/302s	433871/582s	606189/4051s	636849/5341s
3opt	1232/377s	1224/286s	-	-	-	-

Figure 6: Test results table of Local Search algorithms

100 nodes

As for the initial solution of Nearest Neighbour algorithm, we can note that 2-opt algorithm and 2-opt with DLB find the same solution in terms of total path length. Given the small size of the dataset, the 2-opt algorithm doesn't take much computational time, which is why we don't appreciate an improvement in terms of speed-up due to the DLB technique.
As far as the pruning techniques, we can note that both require less computational time than the 2-opt algorithm, but, in both cases, the quality of the solution in terms of tour length is slightly worse than that obtained with the 2-opt algorithm. In any case, the fixed radius method seems to obtain a better solution than the p-nearest one, in a time that is even slightly less.
The best solution is obtained with the 3-opt algorithm, but this requires a really high computational time.

Starting, instead, from the initial random solution, we note that the classic 2-opt and 2-opt algorithms with DLB allow us to improve a lot the length

of the path get from the initial solution, but they obtain a lower quality than that one, previously seen, having Nearest Neighbour initial solution. This shows us that the initial condition influences the solution of the Local Search algorithms.
The DLB technique manages to slightly improve the computational time of the classic 2-opt algorithm and it obtains a total tour length that is even slightly less.
The pruning methods allows to speed-up the 2-opt algorithm, but the length of the path is longer than that one of the classic 2-opt algorithm, especially in the case of the p-nearest method.
Again, the best solution, in terms of total distance, still remains the path produced by the 3-opt algorithm, despite it requires again a very long computational time; we can note that the quality is even better than one obtained starting from Nearest Neighbour initial solution.

5000 nodes

Algo/node	5000 NN	5000 RND
Initial	477613/11.26s	19711844/0.02
2OptPnearest = 0.90	445157/306s	2378265/506s
2optRadius = 1	430336/294s	462395/550s

Figure 7: Test results table of 2-opt algorithm with pruning

As we can see from the results table, we find a far better solution starting from the initial condition Nearest Neighbour rather than the random one.
Starting from the initial solution Nearest Neighbour, the classic 2-opt agorithm and the DLB technique, applied to this, produce the same solution.
The fixed radius prunig method, taking the percentage value of 120% of *GetMinMaximum* parameter, manages to find a good solution in a good computational time, therefore representing a speed-up for the 2-opt agorithm. The same is not true for the value of 96.5% for p-nearest method, which requires a very long time without significantly improving the initial solution.
However, given the size of the dataset, the percentage value of 96.5% is very high. Trying to take a much smaller size of the neighbours, such as 90% of

the total nodes, we note that the computational time decreases a lot, but, obviously, the total length of the tour gets worse.

Instead, starting from the random condition, the best solution found in terms of distance is that one of the 2-opt algorithm.
On the other hand, of course, we obtain a worse quality both in terms of tour lenght and computational time. Furthermore, even in the case of fixed radius method, it is necessary to resort to a smaller size because the computational time at 120% is very high. We also note that the p-nearest method let us obtain a poor quality solution, both in time and especially in length. Obviously, in order that the pruning methods could represent a speed-up for the 2-opt, it is necessary to take lower percentages that decrease the computational time, as the results in figure 7 show, taking as new values 90% of the total number of nodes and 100% of *GetMinMaximum* parameter. As a consequence, the total length of the path gets worse further.
Unfortunately, given the size of the dataset at issue, this time we are unable to test the 3-opt algorithm.

Italy

Algo/node	16863 NN
Initial	701404/100s
2OptPnearest = 0.85	642772/5453s
2optRadius = 0.90	633144/2869s

Algo/node	16863 RND
Initial	70305161/0.09s
2OptPnearest = 0.70	21722918/6265s
2optRadius = 0.80	7413287/4355s

Algo/node	16863 NN
Initial	701404/100s
2OptPnearest = 0.50	680463/2530s
2optRadius = 0.50	668768/1192s

Algo/node	16863 RND
Initial	70305161/0.09s
2OptPnearest = 0.50	35814195/3645s
2optRadius = 0.50	22027570/2904s

Figure 8: Test results table of 2-opt algorithm with pruning

For this particularly large dataset, we can see that the best solution is found with the 2-opt having Nearest Neighbour initial condition.
It should be noted that the time needed to run the only 2-opt with random initial condition is even greater than that one required by the 2-opt with Nearest Neighbour initial solution added to the time of the Nearest neighbour algorithm iteself.
The DLB technique works well, starting from both initial solutions. Indeed, it allows to obtain a tour having a total length equal or close to that one produced by the 2-opt algorithm, in a rather shorter computational time.
As regards the results obtained starting from the Nearest Neighbour initial solution, the pruning method having the fixed radius set at value of 120% of the parameter at issue, obtains a path that has a length really close to that one produced by the classic 2-opt algorithm, in a computational time which is higher. This is not a good result in terms of speed-up of the 2-opt algorithm.
The same is not true for the p-nearest method, taking for it a percentage of 96.5% of nodes, for which the length of the tour is worse than all the others 2-opt algorithms, in a really high time.
Given the size of the dataset, to put right the problem we have chosen to consider a much smaller size of the neighborhoods, taking as a parameter only 85% of the total number of nodes and 90% of *GetMinMaximum* but this reduction is still insufficient. Finally, we have attempted to reduce the size of both parameters, choosing for both them a percentage of 50%. We note that to decrease the computational time the quality of tour length also decrease.

Given the bad results obtained starting from the initial random solution with both pruning methods, initially having parameters set at 96.5% for the p-nearest and 120% for the fixed radius, we have tried to consider much lower percentages. In particular, we have decided to take only 70% of the number of all cities and 80% for the fixed radius. As we can see from the results shown in figure 8, in the case of p-nearest method the computational time is still too high compared to that of the 2-opt algorithm.
Finally, we have tried to take, also for this initial solution, a percentage of

50% for both parameters. In both cases it is possible to obtain an improvement in terms of time, but the length of the both final solutions is very far from that of the tours produced by classic 2-opt algorithm and by 2-opt with DLB.
Obviously, also in this case, given the huge size of the dataset, it hasn't been possible to test the 3-opt algorithm.

Conclusions

What we can deduce, at the conclusion of the tests made regarding Local Search algorithms studied, is that we notice a strong sensitivity of the final solution to the initial condition.
Starting from an initial solution already having a good quality, in most cases it allows to obtain better solutions in terms of tour length, in a shorter computational time.
Usually, the DLB speed-up technique, applied to the 2-opt algorithm, leads to a good result in terms of quality-time ratio.
As long as you are dealing with small datasets, the two pruning methods seen allow us to get a tour with a good quality in terms of length, effectively speeding up the 2-opt algorithm.
Increasing the size of the datasets, the pruning methods don't always allow us to obtain really good results. Since their goal is to speed-up the algorithm to which they are applied, in order that the computational time can remain less than that of the 2-opt algorithm, we can't choose very large neighborhood size; this means that the quality in terms of tour length we get is not always so good.
Moreover, the pruning technique seems to be particularly sensitive to the initial condition and it considerably worsens with the initial random solution.
For what has been shown with the 100 nodes dataset, we have seen that the 3-opt algorithm is the one that allows us to find the tour with the shortest length compared to all the other implemented algorithms.
Unfortunately, we are unable to say the same for bigger datasets due to the computational time that the 3-opt algorithm requires.