IMPROVING LOCAL SEARCH PROBLEMS WITH TOMATO CLUSTERING

MARIANA HENRIQUES, JOÃO NOGUEIRA, AND ANTÓNIO SALGUEIRO

Abstract. Among the most well-known and widely used heuristic methods for the Traveling Salesman Problem (TSP) is the local search with k-exchange neighbors, k-opt, and in particular the 2-opt. This paper explores further how Topological Data Analysis (TDA) can improve the performance of the 2-opt heuristics in solving the TSP.

1. Introduction

Local search algorithms, such as the 2-opt algorithm, have proven to be effective heuristics for addressing the TSP. These algorithms iteratively improve the solution by making small, local changes. However, local search methods often face challenges, such as getting trapped in local optima, which prevent them from finding the global optimal solution. To overcome these limitations, various enhancement techniques have been explored, including the integration of insights from TDA [1].

Topological Data Analysis is an emerging field that provides tools for analyzing the shape of data. TDA uses concepts from algebraic topology and computational geometry to uncover the intrinsic geometric and topological structure of data sets. Persistent homology, one of the main tools in TDA, allows for the identification of features across multiple scales, which makes it particularly useful for understanding complex data structures from a global perspective.

In this paper, we study a method for incorporating TDA into the 2-opt algorithm complementing existing work in the literature as in [1]. However, instead of warm-starting the TSP algorithm with a traditional persistence homology approach, we explore further with the use of the Topological Mode Analysis Tool (ToMATo) method.

The findings suggest that TDA, particularly through ToMATo clustering, can significantly influence the performance of local search algorithms compared to simple TSP or a traditional persistence approach. In addition, a single-cluster study approach yielded promising results comparable to those of the ToMATo approach.

In Section 2 we introduce the main aspects of the TSP, the 2-opt heuristics, and TDA. In Section 3 we present a description of the methodology used to integrate TDA with local search algorithms. In Section 4 we present the main experimental results of the method presented in the paper.

This work was partially supported by the Centre for Mathematics of the University of Coimbra - UIDB/00324/2020, funded by the Portuguese Government through FCT/MCTES.

2. Preliminaires

2.1. **Local search problems.** Local search algorithms are iterative methods designed to identify local minima within a set of feasible solutions. With each iteration, these algorithms improve the value of the objective function, typically by decreasing it, until a local optimal solution is reached.

Over the past decade, there has been a resurgence of interest in local search algorithms. This renewed enthusiasm can be attributed to several factors. Firstly, the relative ease of implementation of some local search algorithms makes them applicable across various disciplines. Secondly, many local search algorithms have robust mathematical foundations, offering reliable theoretical results and insights. Finally, advances in computational resources and data structures have increased the capability of local search algorithms to be used in large-scale problems.

2.1.1. *Travelling salesman problem and the 2-opt heuristics.* The Traveling Salesman Problem is a classical optimization problem. The basic concept is the following: a traveling salesman wishes to visit a list of n cities only once. To go from city i to city j there is a distance (cost) associated. What is the least costly route the salesman can take, always returning to the starting point? In other words, what is the shortest Hamiltonian cycle on the complete graph determined by the cities, where cities are represented as nodes and edge weights represent the costs of traveling between them?

Since the 1900's many attempts to solve the TSP have emerged but it was first formulated, as we know it, by Karl Menger in the 1930's. There are some algorithms and heuristics capable of providing an approximate solution to the TSP [5]. Yet, since it is a NP-hard problem, an efficient optimal method is unlikely to exist. One classical approach to the TSP is the 2-opt heuristics. It is called 2-opt because it makes changes in 2 edges if the cost function is improved. One of the many advantages this heuristic offers is its speed on each iteration. In addition, the solutions obtained are very close to optimal or, in some cases, the actual optimal solution.

Starting with an initial Hamiltonian cycle on the complete undirected graph determined by the cities, the 2-opt heuristic iteratively improves the tour by removing two edges and reconnecting the two resulting paths in the opposite way, effectively reversing the order of the nodes between these two edges. This swap continues until no further improvements can be made, reducing the total travel cost and converging towards a more optimal solution. This approach leverages the properties of Hamiltonian cycles and binary decision variables to systematically reduce the search space and find a near-optimal solution in a computationally efficient manner. In Chapters 8 and 9 of the book [3] by Gutin and Punnen we can find an overview of these TSP heuristics.

2.2. **Topological Data Analysis.** Topological data analysis is a rapidly growing area of applied mathematics. It involves the application of mathematical concepts and techniques from algebraic topology and computational geometry to analyze and interpret data.

With TDA we can extract topological features from point cloud data that are invariant under different metrics. These features can be calculated from the data using mathematical objects called simplicial complexes. By constructing the simplicial complexes from the data, TDA captures the local and global connectivity and higher-dimensional relationships present in the data set. Key Instruments used in topological data analysis include persistent homology and Betti numbers. Persistent homology measures the persistence of topological features across different scales or levels of threshold, such as connected components and global connectivity, holes, voids, and higher-dimensional voids. The Betti numbers describe the number of connected components, holes, and voids in a topological space. They can be computed from the persistent homology and provide quantitative measures of the data's topological structure. In [4], Carlsson and Johansson provide a recent introduction to TDA.

2.2.1. *ToMATo algorithm.* The ToMATo algorithm (Topological Mode Analysis Tool) is a clustering algorithm based on the local density of the point cloud [2]. It starts by estimating the density of each point, using the distance-to-measure function. Then, considering a nearest-neighbors graph, each point is associated with the densest neighbor, thus creating an initial clustering, whose clusters are dominated by the vertices (the modes) whose neighbors are less dense. Then we determine the proeminence of each of these clusters, which is given by the connections with other clusters: traversing the points in descending order of density, if a point P has neighbors in two different clusters, the proeminence of the least dense cluster C is the difference of the densities of P and the mode of C; these two clusters are then merged. Once we finish the process, thus arriving at a single cluster, we restart the process. Since we do not want to merge all clusters, we set a threshold that determines that the merging occurs only for proeminences lower than the threshold. This threshold can be set automatically, by looking at the largest interval between proeminences, and choosing it inside this interval, or can be chosen appropriately to reach a given number of components.

3. Methodology

In this paper, data sets will be treated as point clouds, the points being local minimizers of the TSP problem. TDA will be used to find similar features that are usually present in good local minimizers. The goal is to identify those features and then encourage the presence of those segments in subsequent runs. This approach has some similarities to some common heuristics known in the optimization field, such as Simulated Annealing, Guided Local Search or Tabu Search.

Let X be a point cloud with N vertices. A collection E of edges connecting points of X will be called *simple* if it consists of disjoint simple open paths. A simple collection E can be extended to a closed simple tour with N edges (therefore passing by all vertices of X), which we call a *completion* of E.

The general method used to find short tours on X, in all algorithms, is the following:

(1) Start with a simple collection E of edges;
(2) Pick T random completions of E;
(3) Apply the 2-opt algorithm to each of these completions.

We first use the general method with $E = \emptyset$, obtaining a set S of T tours. Then, we use again the general method with a specific E, which depends on the algorithm. For algorithm 1 (2-opt+2-opt), we set again $E = \emptyset$. For algorithm 2 (2-opt+TDA), we consider the Hamming distance on S and use TDA to group the tours of S in C classes. For algorithm 3 (2-opt+ToMATo), we use instead the ToMATo algorithm to cluster S in the automatic number of classes determined by the algorithm. For algorithms 2 and 3, we then consider the set E of edges that are present in all classes. We prune this set in the following way: we order the edges by decreasing order of frequency and eliminate an edge e_n if $\{e_1, \ldots, e_n\}$ is not simple. If we end up with a collection E with more that pN edges (where $p < 1$ is a parameter), we consider only the pN most frequent edges.

After applying each algorithm we obtain $2T$ tours. In each algorithm, we choose the shortest of the $2T$ tours obtained.

4. Experimental Results

In order to test the efficacy of the association of the ToMATo algorithm with the 2-opt strategy, we performed a series of tests on specific point clouds.

We used the parameters $T = 500$, $C = 3$ and $p = 0.9$. Moreover, for each point cloud, we repeated the process 100 times and averaged the lengths of the shortest tours obtained.

The 46 point clouds used were taken from TSPLIB [6], with sizes varying from 48 to 493.

In smaller sets, using 2-opt alone is sufficient to obtain optimal solutions, so there is no advantage in using other methods.

Figure 1 shows the box plots of the results obtained for one of the point clouds, which shows that using ToMATo provides consistently better results. The complete set of results is given in Table 1.

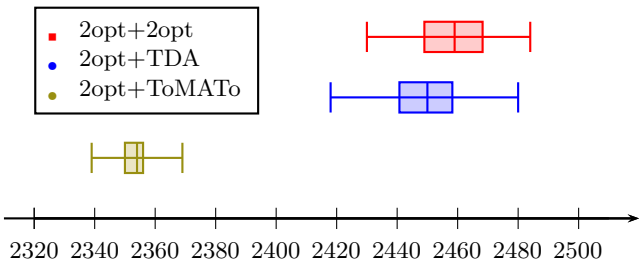

Figure 1. Box plots of results obtained by the three algorithms on the point cloud `rat195`.

In Figure 2, we have represented a graph with the results achieved by the three methods. The x-axis corresponds to the excess, relative to the optimal solution, obtained by the best of T runs of the 2-opt algorithm. The y-axis corresponds to the excess, relative to the optimal solution, obtained by the three algorithms. For example, for the point cloud `rat195`, the optimal solution has length 2323, the average shortest tour obtained by performing 2-opt T times had length 2466 (excess of 6.2%) and the average shortest tour obtained by algorithm 3 had length 2353 (excess of 1.3%), thus corresponding to the triangular point with coordinates

$(0.062, 0.013)$ in the graph below. We also trace the three regression lines and indicate their slopes.

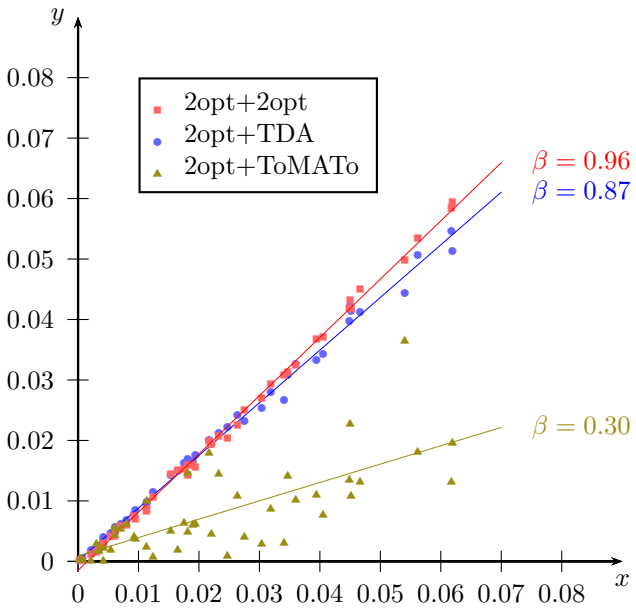

FIGURE 2. Representation of the results achieved by the three methods applied to the 46 point clouds taken from TSPLIB. The x-axis corresponds to the excess, relative to the optimal solution, obtained by the best of T runs of the 2-opt algorithm. The y-axis corresponds to the excess, relative to the optimal solution, obtained by the three algorithms. We also trace the three regression lines and indicate their slopes.

We concluded that running the 2-opt algorithm T more times improved the solution obtained by 4%. Applying the TDA method is slightly better, improving the solution by 13%, confirmimg the results obtained in [1]. Applying the ToMATo method has proven to be much better, improving the solution by 70%.

By testing further the case of a single clusters (that is, setting $C = 1$), we obtained results very similar to Algorithm 3. In fact, the clustering provided by ToMATo had generally a small cluster of just one tour and a large cluster consisting of all other tours.

Point cloud	N	**Optimal**	**2-opt+ 2-opt**	**2-opt+ TDA**	**2-opt+ ToMATo**
a280	280	2579	2707	2693	2673
att48	48	10628	10630	10634	10630
bier127	127	118282	119547	119640	118376
ch130	130	6110	6203	6209	6149
ch150	150	6528	6691	6679	6554
d198	198	15780	16017	16014	15809
d493	493	35002	36579	36441	35459
eil101	101	629	649	649	635
eil51	51	426	428	428	428
eil76	76	538	550	551	543
fl417	417	11861	12032	12030	11921
gil262	262	2378	2477	2472	2410
gr137	137	69853	71295	71339	70863
gr202	202	40160	41894	41858	41069
gr229	229	134602	138552	138360	135769
gr96	96	55209	56311	56317	56201
kroA100	100	21282	21341	21360	21329
kroA150	150	26524	27038	27036	26645
kroA200	200	29368	30159	30114	29454
kroB100	100	22141	22296	22327	22223
kroB150	150	26130	26546	26563	26287
kroB200	200	29437	30359	30345	29853
kroC100	100	20749	20836	20848	20862
kroD100	100	21294	21477	21502	21505
kroE100	100	22068	22231	22239	22158
lin105	105	14379	14410	14412	14402
lin318	318	42029	42682	42768	42286
pcb442	442	50788	53808	53390	51785
pr107	107	44303	44487	44507	44387
pr124	124	14379	14410	14412	14402
pr136	136	96772	98312	98417	98182
pr144	144	58537	58545	58549	58552
pr152	152	73682	73904	73978	73682
pr226	226	80369	80843	80867	80802
pr264	264	49135	50645	50442	49280
pr299	299	48191	49981	49843	48559
pr439	439	107217	111153	110781	108396
pr76	76	108159	108286	108359	108166
rat195	195	2323	2458	2449	2353
rat99	99	1211	1235	1237	1211
rd100	100	7910	7976	7985	7928
rd400	400	15281	16096	16053	15558
st70	70	675	676	676	676
ts225	225	126643	127300	127368	127186
tsp225	225	3916	4079	4077	3958
u159	159	42080	42682	42768	42286

TABLE 1. Comparison of the three optimization methods results on the 46 point clouds taken from TSPLIB. Each result is the averaged length of the shortest tours obtained on repeating the method 100 times. For each point cloud, the best method is highlighted.

References

[1] E. Carlsson, J. G. Carlsson, S. Schweitzer, Applying Topological Data Analysis to Local Search Problems, Foundations of Data Science 4 No. 4 (2022), pp. 563-579.
[2] F. Chazal, L. Guibas, S. Oudot and P. Skraba, Persistence-Based Clustering in Riemannian Manifolds, Journal of the Association for Computing and Machinery 60 No. 6 (2013), article 41, 38 pp.
[3] G. Gutina and A. Punnen, The Traveling Salesman Problem and Its Variations, Springer, 2007.
[4] G. Carlsson and M. Johanson, Topological Data Analysis with Applications, Cambridge University Press, 2022.
[5] S. Lin and B. W. Kernighan, An Effective Heuristic Algorithm for the Traveling-Salesman Proble, Operations Research 21 No. 2 (1973), pp. 498-516.
[6] G. Reinelt, TSPLIB - A Traveling Salesman Problem Library, ORSA Journal on Computing 3 No. 4 (1991), pp 376-384.

CMUC, Department of Mathematics,
University of Coimbra
E-mail: jrhmariana@gmail.com and nogueira@mat.uc.pt and ams@mat.uc.pt