



HAL
open science

Connecting VRP solutions to homology group generators and community modularity of morphologically deformed point cloud

Guillaume Bouleux, Lorraine Trilling, Kirill Sidorets

► To cite this version:

Guillaume Bouleux, Lorraine Trilling, Kirill Sidorets. Connecting VRP solutions to homology group generators and community modularity of morphologically deformed point cloud. CoDIT 2024, Jul 2024, La Valetta, Malta. hal-04645851

HAL Id: hal-04645851

<https://hal.science/hal-04645851v1>

Submitted on 12 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Connecting VRP solutions to homology group generators and community modularity of morphologically deformed point cloud

1st Guillaume Bouleux

INSA Lyon, Université Lumière Lyon 2,
Université Claude Bernard Lyon 1,
Université Jean Monnet Saint-Etienne,
DISP UR4570
Villeurbanne, 69621, France
guillaume.bouleux@insa-lyon.fr

2nd Lorraine Trilling

INSA Lyon, Université Lumière Lyon 2,
Université Claude Bernard Lyon 1,
Université Jean Monnet Saint-Etienne,
DISP UR4570,
Villeurbanne, 69621, France
lorraine.trilling@insa-lyon.fr

3rd Kirill Sidorets

INSA Lyon, Université Lumière Lyon 2,
Université Claude Bernard Lyon 1,
Université Jean Monnet Saint-Etienne,
DISP UR4570,
Villeurbanne, 69621, France
sidorets.kirill97@gmail.com

Abstract—Improving the resolution of a VRP instance often requires knowledge extracted from the instance itself. However, very little work focuses on analyzing or exploiting the topology of the point cloud associated with this instance. In this work, we aim to fill this gap by proposing the characterization of generators of the homological group of the VRP instance that represent natural cycles (tours). To improve the connection between these generators representing tours and optimal or exact tours, we morphologically deform the point cloud of the instance beforehand. The results obtained are clearly encouraging, showing a strong connection between generators obtained by polynomially bounded complexity and optimal or exact tours.

Index Terms—Community Detection, Modularity, Topological Analysis, Homology Group Generator, VRP, Complexity, Graph Morphing, Circular Layout, Smith Normal Form.

I. INTRODUCTION

A vehicle routing problem [8] (VRP) is a combinatorial optimization problem that addresses the challenge of determining the optimal routes to visit a set of locations that satisfy specific constraints. The objective of VRP is to minimize the total distance or other cost metrics, such as the number of vehicles, time, or price. VRP is widely used in various industries to optimize logistics, resulting in reduced overall costs and improved service quality.

In industrial applications, the minimal total cost is usually not the only requirement for routes [5]. They also need to satisfy certain constraints, such as delivery time windows, limited vehicle capacity, or other limitations. There are different variations of the vehicle routing problem that are specified in ways to satisfy this limitation. Each variation addresses a different constraint (or combination of constraints) that the routes must satisfy. Vehicle routing problems are known to be NP-hard problems, therefore they cannot be solved optimally in polynomial time. However, heuristic and meta-heuristic algorithms could be used to provide near-optimal solutions in polynomial time. This can be very useful for big problem instances. Since the definition of the capacitated vehicle routing problem (CVRP) in 1959 by Dantzig, many heuristic and meta-heuristic algorithms have been proposed. The study of this problem continues to this day.

The limitations of exact or heuristic approaches lie partly in the fact that these methods search for Hamiltonian cycles in a complete graph (i.e. closed paths or circuits that pass through each vertex once and only once). It is assumed that all arcs can participate in these circuits, which is a major difficulty. However, we know that not all arcs, especially those between two very distant points, will be useful for creating tours. This information, linked to the instance's topology, is rarely exploited in existing work.

In this research we wish to open a door towards the exploitation of such information by looking for the generators of the homology group [4] associated with the point cloud of the VRP instance and show some first elements on the relation between the optimal tours and the generators obtained.

Let us consider the case of a problem where a delivery vehicle must visit a set of locations. We propose to focus on the distribution of these locations and search for natural circles in the point cloud, without considering the complete graph associated. This corresponds to searching for information on the topology of the point cloud, specifically homological information. Indeed, the use of algebraic topology through the computation and search of homology groups of the point cloud, i.e. searching for intrinsic circles (and by extension spheres, hyperspheres), allows describing the cycle relationship (closed circuit) formed by the set of locations and the vehicle in our example. The idea is that, if such information exists, we can already propose a tour linking a vehicle to a group of locations and directly solve the problem. Otherwise, the search for homology groups can be used as a heuristic to guide the resolution. This idea has already been explored in image processing contexts [10] but, to our knowledge, not for solving variations of vehicle routing problems.

This paper is organized as follows. Section II provides definitions, theoretical elements and implementation guides regarding homology in its broadest sense. Section III is dedicated to illustrate the approach on benchmark instances. Finally section IV gives some concluding remarks and research perspectives.

II. USEFUL GRAPH TOPOLOGICAL INFORMATION

Topology is a branch of mathematics that studies the property of the shapes that are invariant under continuous deformation such as stretching, twisting, or bending, but not tearing or gluing [1]. For example, a donut (Solid torus) could be transformed continuously into a coffee mug, therefore they are topologically equivalent, since they both have a hole [11]. Extracting topological invariants and counting them or associating them with algebraic structures (such as vector spaces) is studied in the algebraic topology area. One of major key invariant for topological analysis is the homology. The very basis of homology is to characterize n-dimensional shapes (let's say discrete shapes) by their k-dimensional holes. Homology uses the notion of cycles, i.e. a closed path without the consideration of its origin. To determine the number of dimensional holes of a shape, the points are joint together to form what is called a cell in the algebraic topology context. When the cells are glued together, they form a complex, a sort of skeleton of the shape. There are several possibilities for defining a cell, and thus a complex. In order to meet our needs, we chose the simplicial complex that meets the simple interpretation and calculation objectives. This approach is called simplicial homology.

A. Simplicial Homology

A simplicial complex is a topological object that consists of a union of points, edges, triangles, tetrahedra, and higher-dimensional polytopes [2] as well as information on how they glued together. A simplicial complex could be divided into parts that are called simplices.

A simplex is the simplest possible polytope that can exist in a given dimension and is formally defined as follows: A k-simplex is a k-dimensional polytope which is the convex hull of its k + 1 independent points (vertices). We denote simplex with the σ symbol. For example, a 0-simplex is just a point, a 1-simplex is a segment between two points, a 2-simplex is a filled triangle and a 3-simplex is a face of tetrahedra (Fig. 1).

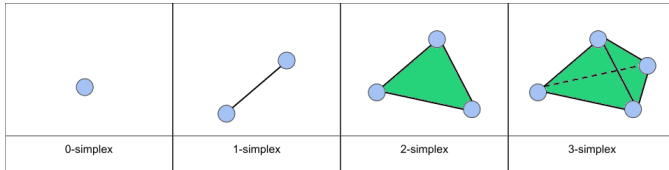


Fig. 1: Visual example of 0,1,2 and 3 dimensional simplices

Simplicial homology is the way to find the holes in a simplicial complex, but in order to give a complete definition of simplicial homology we need to define the chains, and two special types of chains, namely cycles, and boundaries.

Let X be a simplicial complex. Then q-chain is a linear combination of q-simplices with coefficients such as

$$C_q(X) = \sum a_i \sigma_i \quad (1)$$

where each σ_i is a q-simplex of X and the coefficients a_i come from a field \mathbb{F} . The set of all q-chains with a

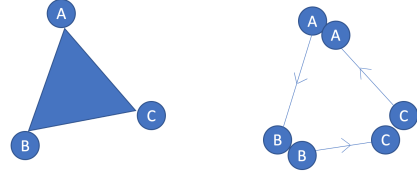


Fig. 2: Boundary operation on a 2-simplex

commutative internal composition law (usually understood as addition) forms an abelian group, denoted $C_q(X, \mathbb{F})$. This (free) abelian group has a basis, and a q-chain is a vector in this basis:

- If the field $\mathbb{F} = \mathbb{R}$, then $C_q(X, \mathbb{R})$ is a vector space.
- a basis of $C_q(X, \mathbb{F})$ is the ordered list of k-simplices.

Since we have a sequence of spaces $\dots, C_{q-1}, C_q, C_{q+1}, \dots$ we can construct an application ∂_q , called the boundary operator.

$$\partial_q : C_q(X) \rightarrow C_{q-1}(X) \quad (2)$$

For a simplex $\sigma = [v_0, \dots, v_k]$ the boundary operator writes

$$\partial_k \sigma = \sum_i^k (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_k] \quad (3)$$

where \hat{v}_i means we have moved this vertex, and $\sum_i^k (-1)^i$ allows to give the oriented edges of the simplex 2.

A cycle is by definition a sum of simplices that is zero, a cycle is therefore the kernel of ∂_q , i.e. $= Ker \partial$. The sum of the edges must be zero. From the above, a "hole" cannot be a closed structure that is the edge (a face) of a higher-order simplex. We use the boundary operator to decompose a higher-order simplex and look at its faces, i.e. $Im; \partial_{k+1}$. The k-th homology group is defined by:

$$\mathcal{H}_q(X) = \frac{Ker \partial_q}{Im \partial_{q+1}} \quad (4)$$

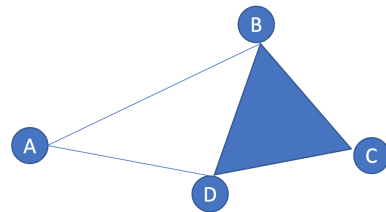


Fig. 3: Representation of simplicial complex X , composed by 0-simplices A, B, C, D , 1-simplices $[AB], [AC], [AD], [BC], [BD], [CD]$, and 2-simplex $[BCD]$

Fig. 3 shows an example of a simplicial complex X . The two matrices below are the matrix representation of the boundary operators ∂_2 and ∂_1 . Matrix ∂_2 is generated according to Eq.(2) and Eq.(3), which lead to decompose the 2-simplex $[BCD]$ (triangle) into 1-simplices (segments). Matrix ∂_1 is generated in the same way, and decomposes all

the 1-simplex (segment) into 0-simplices (points).

$$\partial_2 = \left(\begin{array}{c|c} & [BCD] \\ \hline [AB] & 0 \\ [AC] & 0 \\ [BD] & -1 \\ [BC] & 1 \\ [CD] & 1 \end{array} \right) \quad \partial_1 = \left(\begin{array}{c|c|c|c|c} [A] & [AB] & [AC] & [AD] & [BC] & [CD] \\ \hline [A] & -1 & -1 & 0 & 0 & 0 \\ [B] & 1 & 0 & -1 & -1 & 0 \\ [C] & 0 & 1 & 1 & 0 & -1 \\ [D] & 0 & 0 & 0 & 1 & 1 \end{array} \right)$$

As seen previously, a cycle is determined as the quotient of the kernel of the boundary operator and its higher-order image. However, to determine the kernel and image of these matrices, several choices are possible depending on the group the coefficients belong to. If $\mathbb{F} = \mathbb{R}$, a singular value decomposition will be used, and if $\mathbb{F} = \mathbb{Z}_2$, a "Smith Normal Form" decomposition will be preferred [4]. An example of a "Smith Normal Form" decomposition (analogous to the Gauss pivot) is given by the following matrix:

$$\begin{pmatrix} b_0 & & & 0 & 0 \\ & b_1 & & 0 & \vdots \\ & & \ddots & \vdots & \vdots \\ & & & 0 & b_n \\ 0 & \dots & \dots & \dots & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}$$

where each b_i divides b_{i+1} and gives an integer representation of the spaces. We stress here that the search for the bases, i.e. tours, is polynomial in time with this method.

B. Persistent homology

During the construction of the simplices, some homological features may appear and then disappear. The persistence of these features can be considered as the features of the input. There exist multiple variations of simplicial complex filtration. One of the most popular is Vietoris-Rips Filtration [2] whose principle is illustrated by Fig. 4.

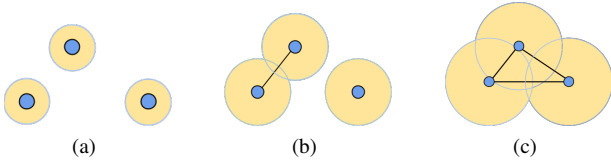
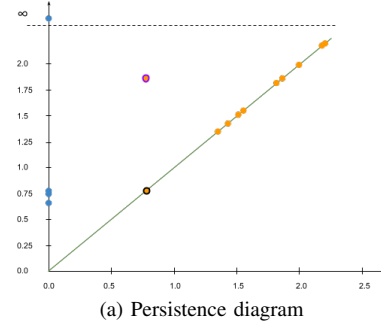


Fig. 4: Visualization of Vietoris-Rips filtration intermediate steps

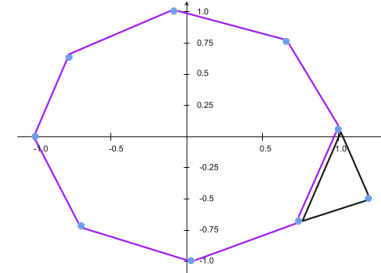
Around each vertex, we could imagine circle with some radius r (Fig. 4(a)). A set of $n+1$ vertices form an n -simplex when each pair of this circle have common point. We also say that n -simplex is born at the radius r of the circle if $r = \max_{v_i, v_j \in V} (\text{distance}(v_i, v_j))$. On Fig. 4(b) 1-simplex was born. On Fig. 4(c) 2-simplex was born whereas 1-simplex is dead (while becoming a face of a higher-order simplex).

During the construction of the Vietoris-Rips simplicial complex multiple 1-simplex holes appeared but all of them eventually disappear due to the creation of higher dimension simplices. We want to study the persistence of chains to find the moment when they disappear. This will give us

information about the length of the life of chains. In order to visualize the length of the life of the feature we use a persistence diagram, in which features are represented by points, features of different dimensions have different colors, abscissa corresponds to the distance on which the feature was born, and ordinate corresponds to the distance on which feature disappears. Chains with long life lengths represent persistent holes and chains with relatively short life lengths represent noise.



(a) Persistence diagram



(b) Simplicial complex

Fig. 5: Persistence diagram showing birth and death of simplices

For example in Fig. 5-(b) we could see that a more persistent hole is highlighted with purple color, which characterizes space better than a hole that is highlighted with black color. And we could see that the purple feature has the best persistence (point with purple border on Fig. 5-(a)) when other features disappear immediately after they are born. In order to compute the life length of the features we use the algorithm given by [13].

C. Computing homology groups generators for routing

The general algorithm used to compute homology groups generators is inspired from [6]. This method is bounded by polynomial time.

Computing a homology generator from point cloud consists of several steps (Fig. 6). The first step (1) is to apply filtration in order to obtain simplicial complex (Vietoris-Rips filtration in our case). This simplicial complex approximately defines the topological space of the input point cloud. In step (2) we detect the most persistent features of this space. In step (3), knowing when features in which we are interested are persisting we are able to compute boundary operation matrix for these features and then compute (4) image and kernel of these matrices. In step (5), the homology generators of the

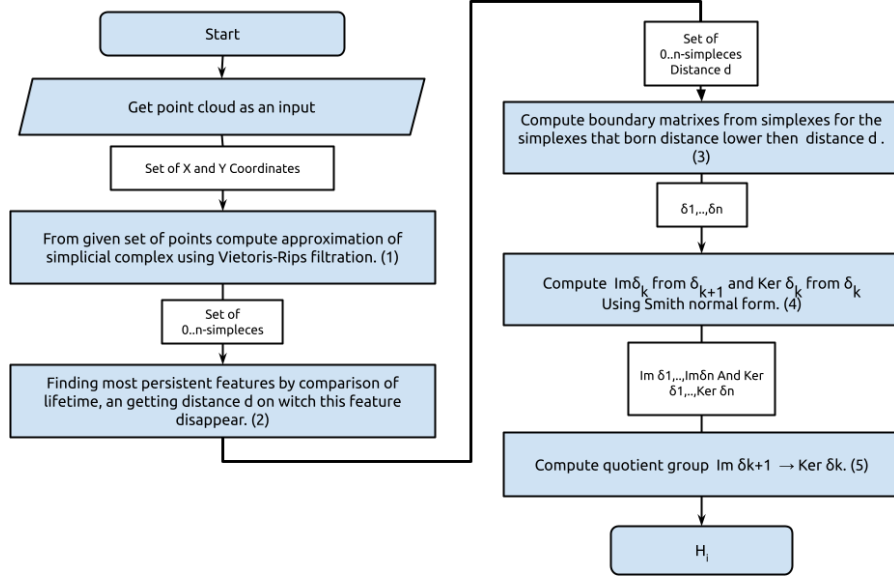


Fig. 6: Sequence of actions required to obtain persistent homology generators

most persistent features are computed, which are compared next to routes in VRP solution.

D. Morphological deformations

So far, we've seen how it's possible to extract cycles - in our context, tours - from a homological analysis of the point cloud formed by the cities/warehouses/customers or whatever of a VRP instance. For lack of space, we can't go into too much detail about the difficulties involved in such an analysis directly and without modifying the initial point cloud and the results derived from it. We would point out to the reader that when the elements to be visited are close to the depot, the homology group generators are often concentrated around the depot. Finally, in the very essence of the methodology presented, it is preferable for the elements or group of elements to be visited to be geometrically closer to a circle. For all these reasons, we propose to morphologically modify the initial point cloud before searching for generators.

To ensure that the cycles are not influenced by the depot, we decided to dilate the point cloud around it. The point cloud is therefore stretched with the depot as its central point. This corresponds to the first deformation, and we point out to the reader that the depot must be added to the generators thus obtained. Next, and with a view to circularly distributing the groups of elements for better characterization of the generators, it is necessary to know these groups of elements. Here, we use a community search in the graph associated with the point cloud. We have opted for a community search by modularity [3], as it seems to be relevant for classifying the difficulty of solving instances [7]. Modularity gives an idea of the coherence structure of element groups compared with a purely random structure in terms of node degree. This is the second step of the deformation which leads to the third one: the circular graph layout [12] of the estimated communities.

Even with these morphings, the whole method remains limited by polynomial time since community detection is linear [3].

III. EXPERIMENT

In this section, we test the method proposed in this work on two VRP instances: one considered as a toy instance, and one extracted from [9]. The topology of these instances is shown in Fig. 7. Fig. 7-(a) shows the exact solution of the VRP problem under capacity constraints for four vehicles found by Integer Linear Programming. Vehicle capacity is not directly included in any constraints, but is taken into account when determining the number of vehicles needed to serve all customers. We therefore first tested the search for generators of the homology group without morphological modification of the point cloud associated with the instance. The persistence diagram detected two homological features in this point cloud, and the generators obtained are shown in Fig. 8. We can see that naturally a cycle (a tour) is detected with the set of points (0,7,11,12,13,15) which, with the exception of point 7, corresponds to an exact route of the problem. The second cycle detected and estimated corresponds to the points (0,5,2,6,10,16,9) which do not correspond to an exact solution to the problem, but have similarities: points 8 and 14 are not visited, whereas the other points are included in one single cycle instead of two. This tour could be a tour for 3 vehicles only. After morphological deformation of the point cloud by (i) dilation around the depot, (ii) characterization of the communities by modularity and then (iii) by circular deformation of these communities, we were able to identify the homology generators much more clearly. These are shown in Fig. 9. Clearly, for three vehicles only, all three exact tours have been found. For four vehicles, all the generators coincide with the exact routes of the problem.

While this first problem was a toy example, we propose to analyze the results of the method proposed here on a much

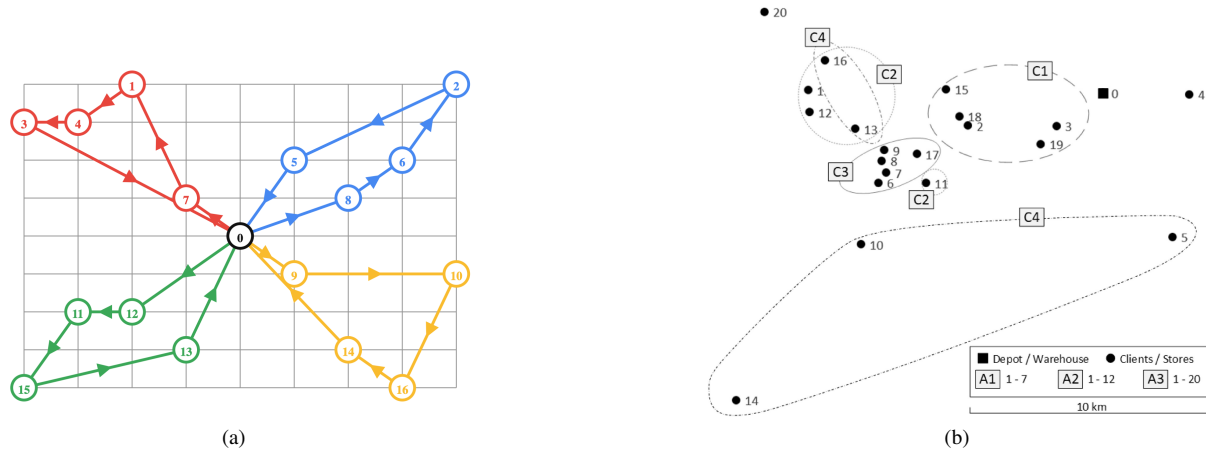


Fig. 7: (a) Exact solutions to capacity constrained VRP toy example, (b) Topology representation for the VRP treated in [9] and associated to a real problem.

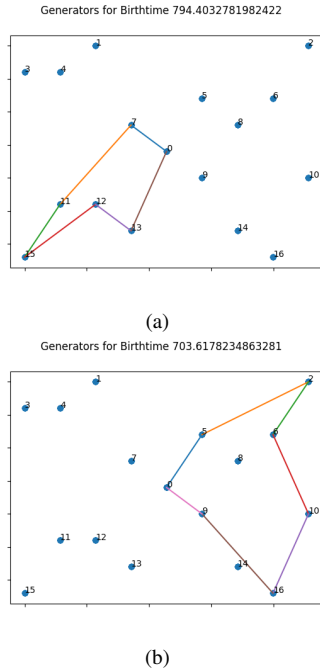


Fig. 8: Homology generators obtained from the VRP of Fig. 7-(a) without morphological modification

more complicated VRP problem treated in [9]. Here again, the point cloud topology associated with the VRP problem is shown in Fig. 7-(b). The topology of this problem is already more complex, in [9] the instances are also closer to reality, including different time related constraints, and the authors showed just how difficult it is to solve this problem that includes 20 customers to visit. The search for generators of the homology group without morphological modification of the point cloud yielded only one generator shown Fig. 10-(a). Although this is the only generator found by the method, which we recall uses only the structure of the point cloud, it is interesting to note the relevance of the cycle (tour) obtained.

Indeed, referring to the solutions given in [9], we see that this generator is extremely close (not in terms of solution's cost, but more in terms of precedent-successor) to a tour obtained for the instance L2 where just one point differs. Perhaps even more remarkable are the generators obtained after morphological deformation of the point cloud. These generators, which correspond to Fig. 10-(b) and 10-(c), are within a single point of the optimal tours for the instance L4 of [9].

IV. CONCLUSION AND PERSPECTIVES

The search for solutions to VRP problems is a very difficult task, due to its high complexity. Many heuristic-based methods are regularly proposed, but very few exploit knowledge linked to the problem's topology. When this topology is used, it enables us to classify problem instances according to their difficulty of resolution cite. In this work, we have tried to make the most of the information provided by the topology of the point cloud associated with a VRP instance. The idea was to show that the structure of the point cloud alone, i.e. the layout of cities/warehouses/customers or whatever, was packed with sufficiently powerful information that it could, with polynomial bounded complexity, extract solutions to very complex instances. It's a conclusive attempt when three morphological operations are used. Indeed, it seems necessary first to expand the point cloud around the depot, then to search for communities by modularity, and finally to circularly distribute these communities around their barycenter, so that the search for and characterization of homology generators is as relevant as possible. Here again, we stress that the entire methodology proposed in this work has a polynomially bounded complexity.

Oviously, this is a preliminary work, but it opens the way to a more precise search for similarities between the homological information and the tours generally obtained after long hours of computation. It is therefore planned to test this method on many more instances, and then to propose an optimization algorithm that could at least give a set of solutions close

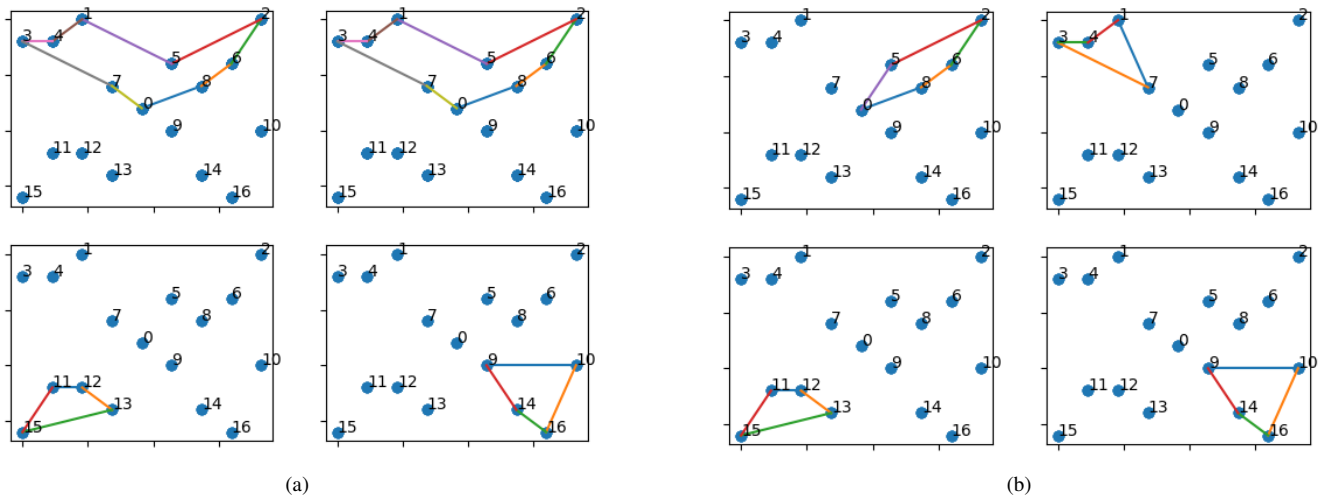


Fig. 9: Homology generators obtained from the VRP of Fig.7-(a), (a) and (b) with morphological deformations explained in II-D. Two sets of generators are obtained. In (a), three generators which coincide with the VRP with three vehicles whereas for (b) the four generators obtained are similar to the exact solution of Fig. 7-(a)

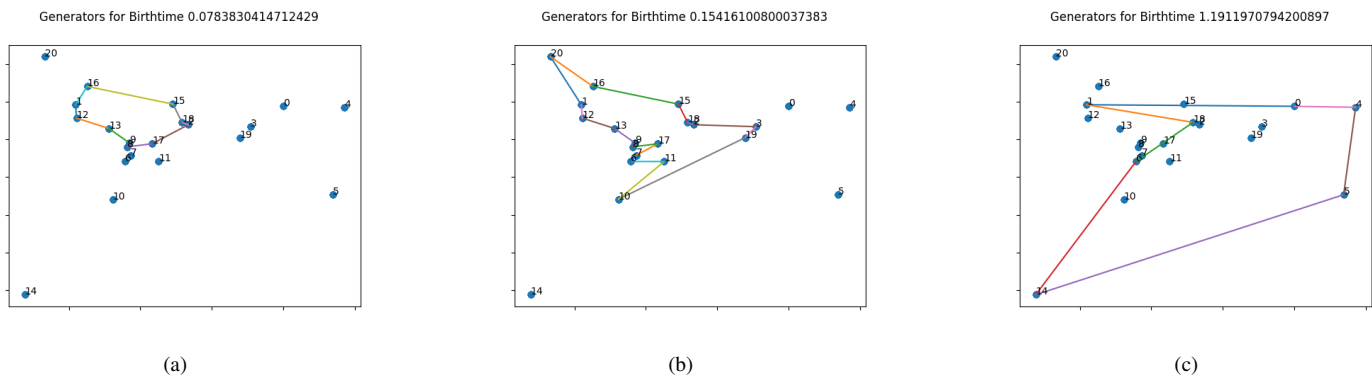


Fig. 10: Homology generators obtained from the VRP of Fig.7-(b), (a) without morphological deformation, (b)-(c) with morphological deformations cited in II-D. For (b) and (c) two generators are found and are quite the same as the solutions obtained for instance L4 of [9].

enough to optimality to greatly accelerate the search for solutions by more conventional algorithms.

REFERENCES

- [1] Mehmet E. Aktas, Esra Akbas, and Ahmed El Fatmaoui. Persistence homology of networks: methods and applications. *Applied Network Science*, 4, 12 2019.
- [2] Guillaume Bouleux, Maël Dugast, and Eric Marcon. Information topological characterization of periodically correlated processes by dilation operators. *IEEE Transactions on Information Theory*, 65(10):6484–6495, 2019.
- [3] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004.
- [4] Jean-Guillaume Dumas, Frank Heckenbach, David Saunders, and Volkmr Welker. Computing simplicial homology based on efficient smith normal form algorithms. pages 177–206. Springer Berlin Heidelberg, 2003.
- [5] Geir Hasle and Oddvar Kloster. Industrial vehicle routing. pages 397–435. Springer Berlin Heidelberg.
- [6] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. Computing homology groups. In *Computational homology*, volume 157, pages 93–141. 2004.
- [7] Pavel Krömer and Jan Platoš. Evaluation of Traveling Salesman Problem Instance Hardness by Clustering. *Advances in Intelligent Systems and Computing*, 682:379–387, 2018.
- [8] Suresh Nanda Kumar and Ramasamy Panneerselvam. A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 04:66–74, 2012.
- [9] Isidoros Marampoutis, Marina Vinot, and Lorraine Trilling. Multi-objective vehicle routing problem with flexible scheduling for the collection of refillable glass bottles: A case study. *EURO Journal on Decision Processes*, 10:100011, 2022.
- [10] Ippei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2:508–534, 1 2018.
- [11] Animikh Roy and Andor Kesselman. A novel approach to topological graph theory with r-k diagrams and gravitational wave analysis, 2021.
- [12] Ashley Suh, Mustafa Hajij, Bei Wang, Carlos Scheidegger, and Paul Rosen. Persistent homology guided force-directed graph layouts. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):697–707, 2019.
- [13] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33:249–274, 2 2005.