physical sciences forum

MDPI

Proceeding Paper

Manifold-Based Geometric Exploration of Optimization Solutions †

Guillaume Lebonvallet *, Faicel Hnaien and Hichem Snoussi

Laboratory of Computer Science and Digital Society (LIST3N), Université de Technologie de Troyes, 12 rue Marie Curie, 10300 Troyes, France; faicel.hnaien@utt.fr (F.H.); hichem.snoussi@utt.fr (H.S.)

* Correspondence: guillaume.lebonvallet@utt.fr; Tel.: +33-6-04-49-20-58

† Presented at the 42nd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Garching, Germany, 3–7 July 2023.

Abstract: This work introduces a new method for the exploration of solutions space in complex problems. This method consists of the build of a latent space which gives a new encoding of the solution space. We map the objective function on the latent space using a manifold, i.e., a mathematical object defined by an equations system. The latent space is built with some knowledge of the objective function to make the mapping of the manifold easier. In this work, we introduce a new encoding for the Travelling Salesman Problem (TSP) and we give a new method for finding the optimal round.

Keywords: manifold; optimization; Traveling Salesman Problem (TSP)

1. Introduction

Manifolds are mathematical objects which can represent high-dimension points intrinsically linked to a low-dimension space [1]. Manifolds are used in dimension reduction for the classification of complex objects like facial expressions or different views from the same scene. We want to achieve the resolution of complex problems like the Travelling Salesman Problem (TSP) [2,3] by using manifolds to map the objective function to the solutions space. We can then use the manifold to explore the solutions space and find the optimum solution. However, the solutions space is often too complex and various dimensionsto allow this. Our solution is to describe the solutions space with a latent space of lower dimensions, which is simpler to explore. The manifold can then be defined with simple functions which make the findings of the optimum solution easier. We will apply this protocol on the TSP and give an example.

2. Materials and Methods

The goal of this paper is to find a latent space which is linked to the objective function. This latent space will be mapped by a tree formed by the distance matrix. In fact, we normally define the objective function of TSP problems by the Equation (1) [4–6].

$$Z = \sum_{i \neq j} d_{ij} x_{ij} \tag{1}$$

The link between the solution represented by x and the objective value Z is the distance matrix d. We want to include a part of the knowledge from this distance matrix in the encoding of the solution.

To achieve this, we will use a tree to sort the different solutions based on the links used in the travel. We assume that travels with short links have lower lengths than travels which do not use them. We can then repeatedly split the set of solutions in respect of the use of some links until all solutions have been allocated to a leaf node of the tree. The new encoding is then obtained by numbering the solutions by the use of the best links.

The Algorithms 1 and 2 shows the recursive building of the encoding tree.

Citation: Lebonvallet, G.; Hnaien, F.; Snoussi, H. Manifold-Based Geometric Exploration of Optimization Solutions. *Phys. Sci. Forum* **2023**, *9*, 25. https://doi.org/10.3390/psf2023009025

Academic Editors: Udo von Toussaint and Roland Preuss

Published: 16 May 2024

Algorithm 1 Calculate the encoding tree

```
S ← set of all solutions
L ← list of links sorted accordingly to the distance matrix
tree ← buildNode(S, L)
```

Algorithm 2 Function buildNode(Solution Set S, Links List L)

```
if S has more than one solution then
  l ← first link of L (the best link)
  while S has no solution with l or S has no solution without l do
    Remove l from L
    l ← first link of L (the new best link)
  end while
  S1 ← the set of solutions of S with the link l
  S2 ← the set of solutions of S without the link l
  node ← new intermediate node with l
  Remove l from L
  node.leftChild ← buildNode(S1, L)
  node.rightChild ← buildNode(S2, L)
  return node
else
  return new leaf node with the unique solution of S
end if
```

3. Results

For example, the Equation (2) gives the distance matrix of a TSP with 4 nodes.

$$d = \begin{pmatrix} 0 & 8 & 39 & 37 & 50 \\ 8 & 0 & 45 & 47 & 49 \\ 39 & 45 & 0 & 9 & 21 \\ 37 & 47 & 9 & 0 & 15 \\ 50 & 49 & 21 & 15 & 0 \end{pmatrix} \tag{2}$$

The list of the ten links of this problem is sorted in the Table 1.

Table 1. List of the links for the TSP example sorted by the distance.

Distance	Link
8	(0, 1)
9	(2, 3)
15	(3, 4)
21	(2, 4)
37	(0, 3)
39	(0, 2)
45	(1, 2)
47	(1, 3)
49	(1, 4)
50	(0, 4)

With this list, we can build the encoding tree (Figure 1). On each node, we choose the first link in the list allowing the split of the solutions space. The left child of this node takes the solutions using that link and the right child the others which do not use the link. By continually splitting the solutions space, each solution is assigned to a leaf node of the tree. We can now number the solutions from left to right and we obtain the new encoding in the Table 2.

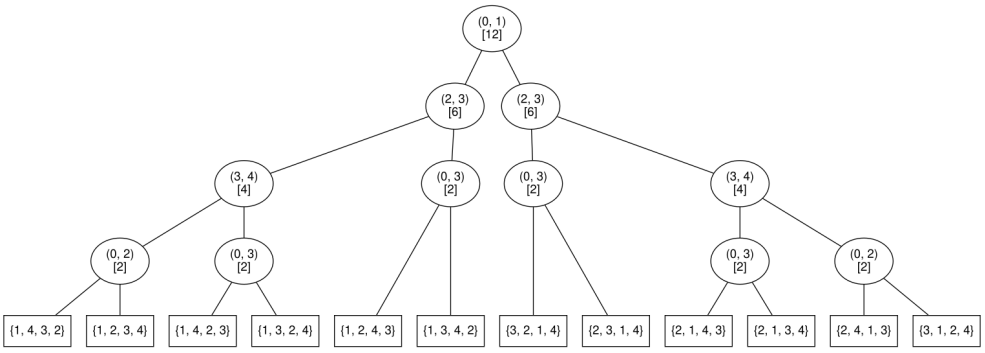


Figure 1. Encoding tree: each intermediate node (round shape) represents a splitting of the solution space based on the use of a link (the left branch keeps the solutions using the link, the right branch the solutions without the link), the number in square brackets indicates the number of solutions; the leaf nodes (rectangle shape) represents the solutions.

Table 2. The new encoding of the solutions of the TSP example.

Encoding	Solution
0	{1, 4, 3, 2}
1	{1, 2, 3, 4}
2	{1, 4, 2, 3}
3	{1, 3, 2, 4}
4	{1, 2, 4, 3}
5	{1, 3, 4, 2}
6	{3, 2, 1, 4}
7	{2, 3, 1, 4}
8	{2, 1, 4, 3}
9	{2, 1, 3, 4}
10	{2, 4, 1, 3}
11	{3, 1, 2, 4}

Figure 2 compares this new encoding with the real rank of the solution, i.e., a sort of the solutions from the best (numbered 0) to the worst. Despite being imperfect, the new encoding shows a tendency between the new encoding and the objective value represented by the rank. We can also see the different branches of the encoding tree.

To demonstrate the drawbacks of the method, the new encoding has been calculated on several simulated TSPs with seven nodes. The results are displayed in Figure 3. The plots show the distribution of the objective value according to the encoding. Three cases are represented with two examples in each:

- The plots (a) and (b) represent a good resolution of the TSP. The optimum is the left most point, encoded with the value 0. The plots clearly show some branches of the encoding tree;
- The plots (c) and (d) display two examples where the optimum is not encoded with 0. It indicates that the tree incorrectly splits the solutions space at some point in the tree. Furthermore, the plot (d) shows that the optimum is not seen clearly in the first branch, which indicates that the wrong split is situated in the tree;
- The plots (e) and (f) show some examples where the encoding fails. The reason could be the low range of distances used for these TSPs. On the plot (e), the branches are difficult to see.

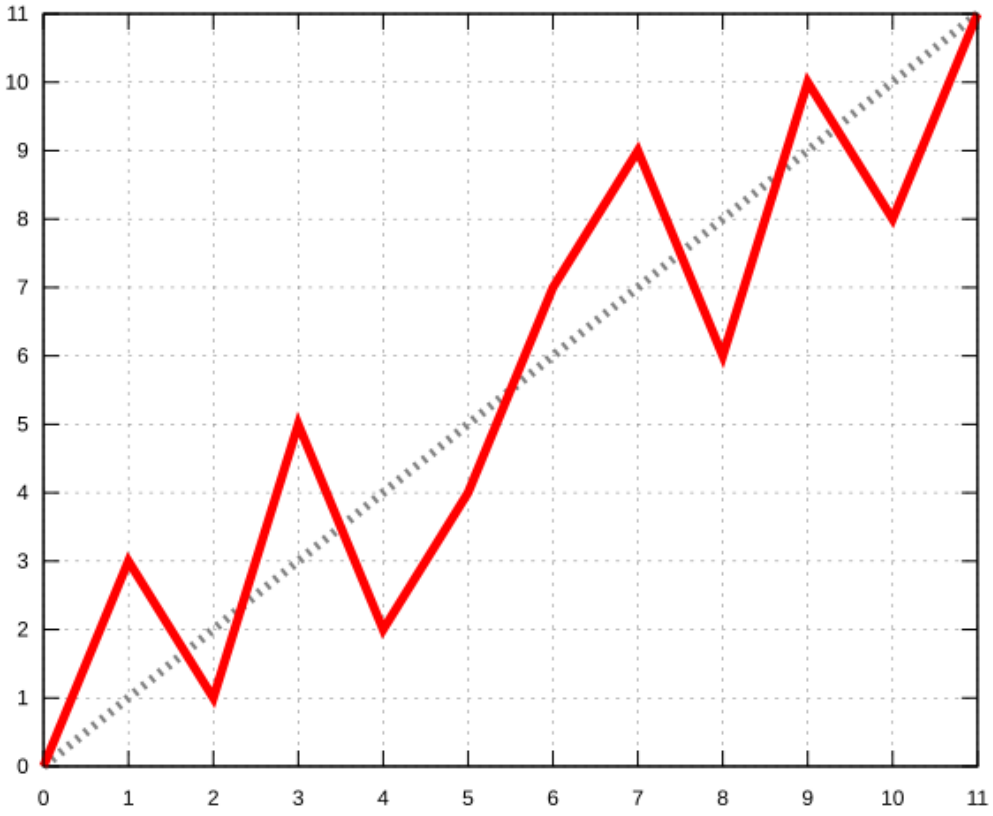
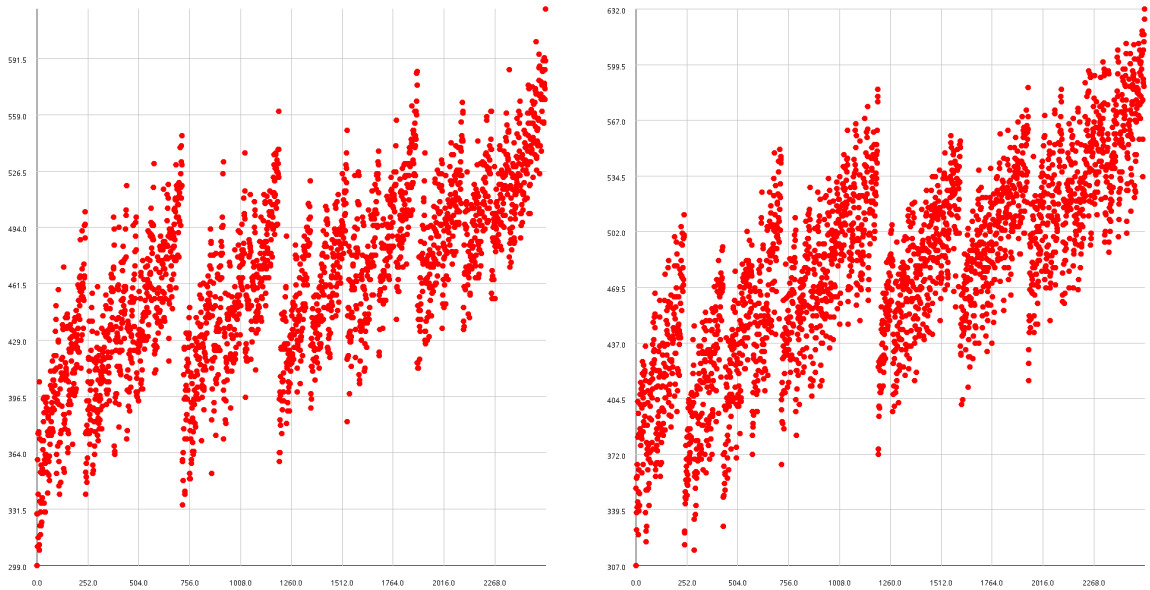


Figure 2. Graph of the new encoding: the horizontal axis represents the rank of a solution (0 is the optimum) and the vertical axis the encoding of that solution.



(**a**) (**b**)

Figure 3. *Cont.*

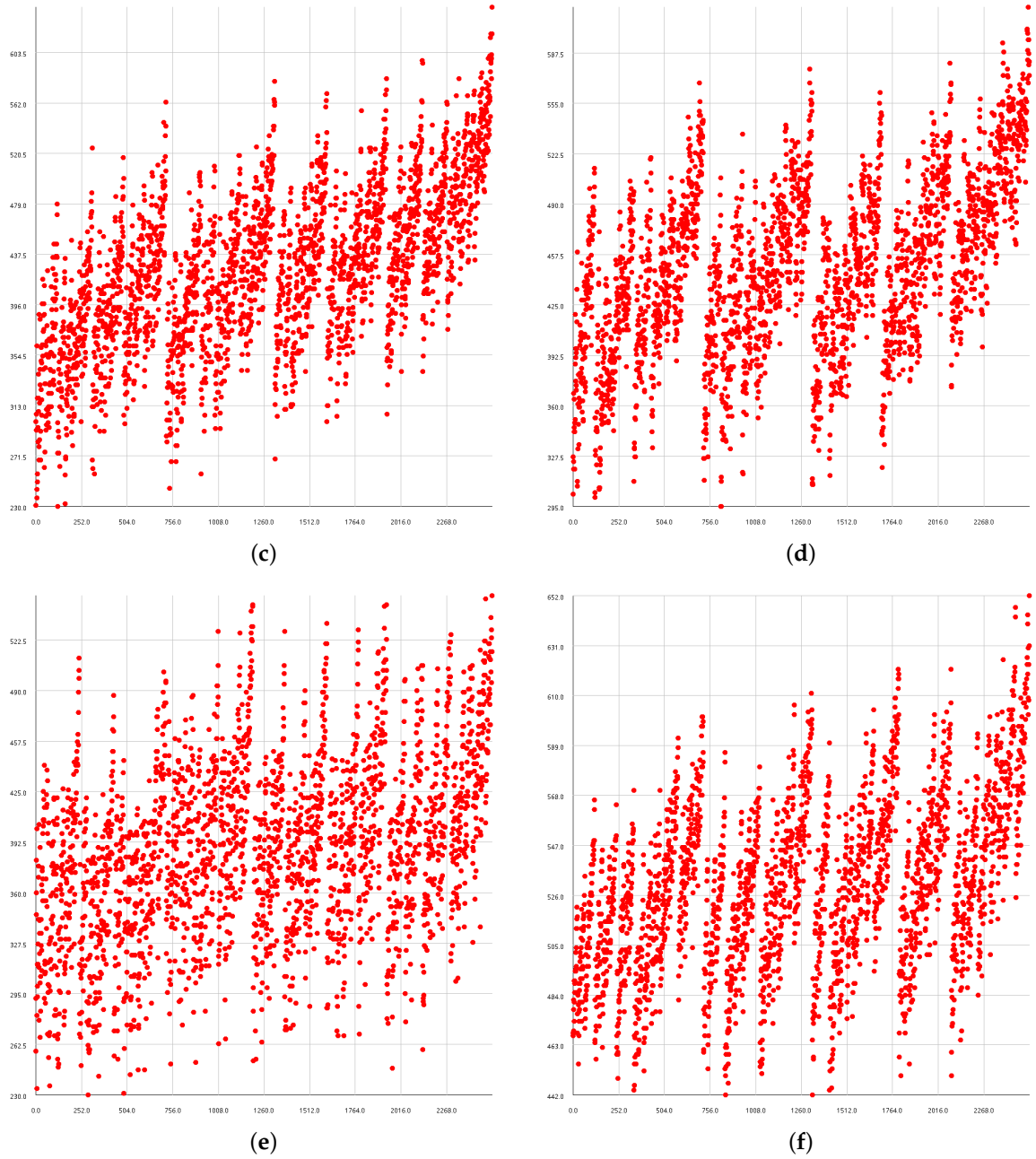


Figure 3. Simulations of the new encoding on several TSP problems with 7 nodes.

4. Discussion

We have an encoding which links the solution to the distance matrix; we need to link this encoding to the objective value. As the encoding is partially aware of the objective value, the neighbourhood of each solution is defined on the objective function. A simple function can be used to link this encoding to the objective value, e.g., a polynomial function. The fitting of this polynomial function to the different points representing the solutions can be made with a simple least-square optimization. Finding the minimum of the polynomial in the encoding space is also simple but can lead to multiple stationary points. As the fitting is not perfect, we need to explore a neighbourhood of each minimum to expect finding the optimum. As the encoding is scalar, it is fairly easy. This neighbourhood will be defined on the latent space of low dimension, which is easier to explore. In fact, the parametrization creates a new encoding of a solution which allows a guided exploration of the solutions space according to the objective, which can simplify the search of the optimal value.

Our new method for finding the optimum is describe as follows:

1. Build a portion of the encoding tree: the stopping criteria can be based on an upper bound of the optimum value [7];
2. For each leaf node which has multiple solutions, choose the solution which uses the better links: the encoding is found by counting the solutions on the branches placed on the left side;
3. Map a manifold of the objective value on the solutions space with the subset of solutions chosen:
4. Find the minimum/minima of this manifold: as the function used is very simple, it should not be too difficult to differentiate and search for the stationary points;
5. The optimum should be found on a neighbourhood of the minimum or one of the neighbourhoods of the minimum/minima if there are multiple local minimums on the solutions space.

Author Contributions: Conceptualization, methodology, software, and writing, G.L.; validation and review, F.H. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no extra funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article and/or generated as described.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, J.; Liu, X.; Wen, Z.W.; Yuan, Y.X. A Brief Introduction to Manifold Optimization. *J. Oper. Res. Soc. China* **2020**, *8*, 199–248. [CrossRef]
2. Pop, P.C.; Cosma, O.; Sabo, C.; Sitar, C.P. A comprehensive survey on the generalized traveling salesman problem. *Eur. J. Oper. Res.* 2023, *in press*. [CrossRef]
3. Bektas, T. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega* **2006**, *34*, 209–219. [CrossRef]
4. Miller, C.E.; Tucker, A.W.; Zemlin, R.A. Integer Programming Formulation of Traveling Salesman Problems. *J. ACM* **1960**, *7*, 326–329. [CrossRef]
5. Dantzig, G.; Fulkerson, R.; Johnson, S. Solution of a Large-Scale Traveling-Salesman Problem. *J. Oper. Res. Soc. Am.* **1954**, *2*, 393–410. [CrossRef]
6. Papadimitriou, C.H. The Euclidean travelling salesman problem is NP-complete. *Theor. Comput. Sci.* **1977**, *4*, 237–244. [CrossRef]
7. Fischetti, M.; Salazar González, J.J.; Toth, P. A Branch-and-Cut Algorithm for the Symmetric Generalized Traveling Salesman Problem. *Oper. Res.* **1997**, *45*, 378–394. [CrossRef]