Usefulness of Solution Algorithms of the Traveling Salesman Problem in the Typing of Biological Sequences in a Clinical Laboratory Setting

Javier Garcés Eisele[1], Carolina Yolanda Castañeda Roldán[2], Mauricio Osorio Galindo[2], Ma. del Pilar Gómez Gil[2]

[1] Universidad de las Américas, Puebla, Depto. de Química y Biología. CIQB.
Ap. Postal 100. Santa Catarina Mártir 72820 México
jgarces@mail.udlap.mx
[2] Universidad de las Américas, Depto. de Ing. en Sistemas Computacionales. CENTIA.
Ap. Postal 100. Santa Catarina Mártir 72820 México
{ccastane, josorio, pgomez}@mail.udlap.mx

Abstract. Our concern is to solve the problem of the typing of deoxyribonucleic acid (DNA) sequences in a laboratory setting. Here we try to find solution algorithms for the classification of restriction patterns, which forms part of the above-mentioned problem, in order to evaluate the amount of information generated by a given restriction enzyme. A distance matrix is generated by comparison of each restriction pattern and used to classify the patterns according to their similarity. This problem can be mapped to the Traveling Salesman Problem (TSP). Several known and new solution algorithms have been tested. Interestingly, a very simple and modified nearest neighbor analysis performed best for this kind of problem. However, when the distance matrix is replaced by a "distinction matrix" (expressing directly with the help of a threshold function the similarity (0) or dissimilarity (1) between restriction patterns) the complexity of the problem was reduced dramatically and it could now be solved easily after transitive closure.

1 Introduction

For the TSP, we are given a complete, weighted graph and we want to find a tour (a cycle through all the vertices) of minimum weight [1]. One formal definition of the TSP can be found in [2]. Interestingly, several problems arising from the analysis of DNA sequences can be formulated analogous to the TSP, one of which will be presented and analyzed herein.

DNA is the deoxyribonucleic acid, i.e. the genetic material that encodes the characteristics of living things DNA consists of strings of molecules called nucleotides. There are four nucleotides in DNA distinguished by its base, each denoted by the first letter of the base: adenine (A), cytosine (C), guanine (G) and thymine (T) [3]. A DNA sequence can, therefore, be treated as a character string using an alphabet of 4 letters. The sequence of these letters defines the characteristics of any living being, thus the

I. Rudomín, J. Vázquez-Salceda, J. Díaz de León. (Eds.). e-Health: Application of Computing Science in medicine and Health Care.

knowledge of the sequence or at least part of it allows the identification of the organism to which the sequence belongs. Thus different types of sequence analysis can be employed in a clinical laboratory setting in order to identify an infectious agent present in a sample taken from a given patient. The instance that will be treated is an example of the so-called sequence-typing problem (STP) applied to the case of the Human Papilloma Viruses (HPV), which is associated with the development of cervical cancer [4]. The required sequence analysis may be performed by a technique called RFLP-PCR (Restriction Fragment Length Polymorphism coupled to Polymerase Chain Reaction). Briefly a segment of the viral genome is analyzed with the help of so-called restriction enzymes, which cut the segment where a small substring is located, i.e. the enzyme EcoRI recognizes the substring GAATTC [5]. The pattern (sizes) of the generated fragments is then determined as it is obviously a function of the sequence itself. The HPV types may then be identified, as long as the corresponding patterns generated by an enzyme are different for each virus. Otherwise, combinations of enzymes have to be used. Until now 48 reference sequences have been published and more than 180 restriction enzymes are available to perform the typing, each recognizing a different subsequence or substring.

In order to select an optimal combination of enzymes to carry out the typing, it is important to evaluate each enzyme, i.e. how much information is yielded on average by the enzyme. This requires in a simple approach to group the restriction patterns according to their similarity, which means that we have to determine the distance between each pair of them and order them linearly according to their similarity. This in turn yields a distance matrix from which we have to select a Hamiltonian path or circuit of minimal weight. Thus, we are confronted with a problem similar to the TSP. The instances are symmetric but not always geometric. However, due to the evolutionary relationship of the viruses, the instances may no behave like random symmetric, non-geometric instances. Furthermore, while a TSP requires the construction of a Hamiltonian cycle, the STP requires finding an optimal Hamiltonian path (restriction patterns ordered linearly according to their similarity). Therefore, although there have been several algorithms published in order to find exact or approximate solutions of the TSP [6], due to the evolutionary relationship between the members it is important to test the behavior of the published and novel hybrid algorithms on these instances. We have not yet analyzed or characterized further the postulated special characteristics of the phylogenetic structure of the TSP.

We started by building a software tool for solving the TSP [7]. This tool has eight solution algorithms; in which five of them are approximate and the other three exact methods. A restricted version of the implementation is accessible online [8]. As almost all algorithms previously implemented in our tool are searching for Hamiltonian cycles, we adapted them to the STP by removing from the solution (a cycle) the edge of maximum weight. As indicated, we evaluated also the usefulness of hybridizing algorithms. We, therefore, constructed and tested two hybrid methods combining initial path with local search algorithms. One of them has already been presented in a previous article [9]. All algorithms are presented briefly in the following section.

2 Methods

Approximate algorithms are a class of algorithms, which do not guarantee optimal solutions but warrant a bound worst-case performance (near to the optimal solution) and run faster than any algorithm that achieves optimality. Subsequently, we describe the approximate algorithms studied in this article.

2.1 2Opt

This solution technique is also called "Two Opt" the short name "Two Optimal", also "Double Option". This technique is one of the most successful heuristic to obtain the approximate solution of the TSP. The Two Optimal Technique is fully described in [10], [11].

2.2 Adaptation-Prim-2Opt-Hybrid method

We modified Prim's algorithm for the minimum spanning tree problem in order to generate an initial path, which was used by the local search 2Opt algorithm in order to optimize the path [9].

A spanning tree is a tree that comprises all the nodes of a given graph and not any more [8]. Greedy algorithms for optimization problems consist of making choices in sequence such that each individual choice is best according to some limited "short term" criterion, which is relatively easy to evaluate. Once a choice is made, it cannot be undone; even if it becomes evident later that it was a poor choice. Although in general greedy strategies don't always lead to optimal solutions or aren't always efficient, Prim's greedy strategy for the minimum spanning tree problem always produces optimum solutions efficiently [1].

Prim's algorithm begins at an arbitrary start vertex and grows a tree from there. During each of the iterations of the main loop an edge is chosen from a tree vertex to a fringe vertex; it "greedily" chooses such an edge with minimum weight [1]. Prim's algorithm produces a minimum spanning tree T; it means an undirected graph with weighted edges. The method is fully described in [9], [10].

In the Adaptation-Prim-2Opt-Hybrid method, Prim's algorithm has been modified such that the result is not a tree but a path. It is another greedy strategy, which has been described previously. Once the path is found, the first and the last node are linked in order to obtain a Hamiltonian cycle (this step is called Adaptation-Prim APRIM), which was fed into the 2Opt method that tries to improve this initial cycle (second step). Both steps joined are called Adaptation-Prim-2Opt-Hybrid method (P2OH).

2.3 MST-2Opt Hybrid method

The problem of determining an optimal tour in the symmetric n city TSP (with a symmetric cost matrix) can be viewed as the problem of finding a minimum cost Hamiltonian cycle in a weighted, undirected graph. With this point in mind it is easy to see that the problem of determining the existence of a Hamiltonian cycle in a complete undirected graph G = (V, E), which is transformable to the symmetric TSP [12].

The following algorithm computes a route close to the optimum of an undirected graph G, using the Minimum-Spanning-Tree algorithm of Prim. Observe that the cost function has to satisfy the inequality of the triangle. In this case the given route found by this algorithm is in the worst case twice as big as the optimal route [12], [13], [14].

The solution algorithm MST-2Opt Hybrid (M2OH) is described below. We have a set (G, c), where G is a complete graph, with a non-negative cost "c".

1. Select any vertex "r" of V [G], which will be the "root" vertex.
2. Compute a minimum spanning tree T for G from a root "r" using the minimum spanning algorithm of Prim: T = (G, c, r).
3. Evaluate L as a list of visited vertices in a walk of a general tree in preorder of T.
4. Link the first and the last node in order to close the path, which will create a Hamiltonian cycle H. This closed path visits all the vertices in the L order.
5. The route obtained from all the vertices of the L list, is transformed in the initial route for the 2Opt algorithm. These five steps form the M2OH.

The execution time of the M2OH is of the order $\Theta(E) = \Theta(V2)$ since the input is a complete graph [13].

2.4 Nearest-Neighbor Adaptation

The Nearest-Neighbor Adaptation (NNA) is a modification of the simple nearest neighbor algorithm [15]. The modification consists in selecting as a specific starting node the one, which has on average the largest distance from the rest of the vertices. From this node we take the nearest neighbor, which hasn't yet been visited, yielding an algorithm with voracious characteristics [15].

2.5 Lin-Kernighan

The heuristics of Lin and Kernighan (LKH) [16] is considered one of the best approximation algorithms of the TSP problem, yielding surprisingly often, optimal solutions for small instances. Briefly, the algorithm uses a flexible n-Opt strategy, where the number n of edge exchanges is determined dynamically at each iteration considering some sort of limiting criterion. We used in our study the implementation of Heldsgaun [17].

2.6 A Better Branch and Bound

This method is based on a search tree where in each step all possible solutions are partitioned in two subsets, one representing all nodes of the remaining search space containing a selected node and the other containing all nodes of the remaining search space without that particular node. Once the ramification has been carried out the bounds for each subset are calculated and the one of the least bound is chosen to continue the search. The particular node picked for the next ramification according to a heuristic that is intended to prune the tree as much as possible. This process is repeated recursively until the Hamiltonian circuit is found [7], [18].

3 Experiments and Results

TSPLIB is a library of sample instances for the TSP that can be used in order to assess the efficiency of algorithms [19]. However, as our goal is not to test the behavior of the algorithms but to analyze their usefulness within the sequence typing problem, we selected a sample of 48 HPV sequence types corresponding to a genomic segment of the L1 gene, which were studied by restriction analysis with 182 restriction enzymes. Only 138 restriction enzymes cut at least one HPV sequence.

In addition, in order to increase the sample size, we performed a restriction analysis with "synthetic" enzymes recognizing all 4-tuples missing from the above mentioned natural set of recognition sequences. We obtained an additional set of 120 instances.

The theoretical restriction patterns were compared with each other. As a measurement of the similarity we used the sum of differences between the migration positions. The matrices showed various degrees of redundancy due to the existence of identical restriction patterns, thus we eliminated the linearly dependent vectors, and some matrices were reduced down to 2x2 matrices. We analyzed only matrices of 3x3 and larger. The distribution of instances is shown in Fig. 1. Later on, we grouped the instances according to the following ranges: 3-9 (10), 10-19 (20), 20-29 (30), 30-39 (40), and 40-48 (50).

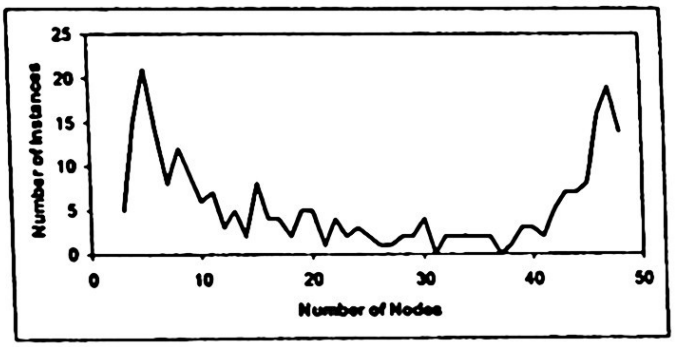
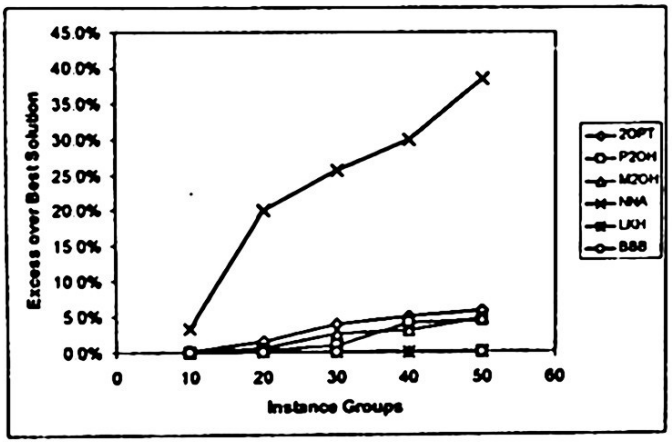


Fig. 1. Size distribution of the instances. We show the total distribution comprising natural as well as "synthetic" enzymes

The matrices were symmetric but frequently not geometric (only 30 out of 259 comply with the triangle's inequality). As the execution times were too short, (less than a millisecond, with the obvious exception of the BBB), the methods were compared by the weight of the Hamiltonian path and cycle. Each instance was analyzed by the mentioned methods, and the result was expressed as the percentage above the shortest path or cycle. We then calculated the average of the above-mentioned ranges. These results are shown in Fig. 2.

A



B

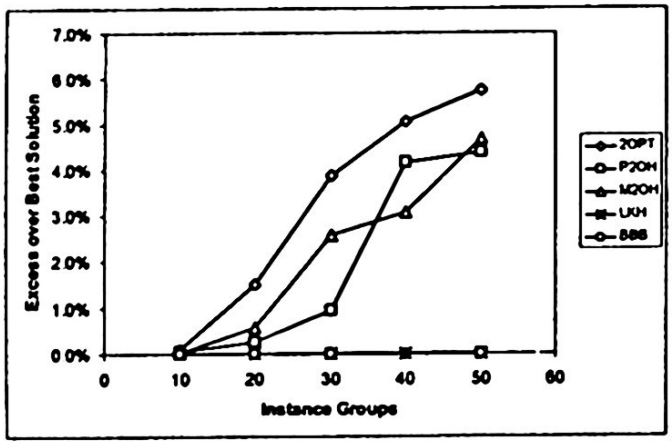


Fig. 2. Hamiltonian cycles. (A) Comparison of the 2OPT, P2OH, M2OH, NNA, LKH, and BBB algorithms. (B) For a better resolution of the results we eliminated the NNA method

As we can see, in Fig. 2 the various methods behave as expected, i.e. the initial path algorithm yields only good solutions for very short instances. All local search algorithms produce better approximate solutions. However, as the instances grow, they deviate stronger from the optimal solution with the exception of the Lin-Kernighan algorithm, which found almost always, optimal solutions. The hybrid methods work better than the local search method by itself.

In Fig. 3 we show the results for Hamiltonian paths. As almost all algorithms previously implemented in our tool are searching for Hamiltonian cycles, we adapted them to Hamiltonian paths by removing from the cycle the edge of maximum weight.

NNA shows a very good behavior with tiny instances; however, its efficiency decreases rapidly with instances above 20 nodes. With the analyzed instances, all other algorithms show a similar behavior. There is only a small but noticeable success of the BBB and LKH for instances larger than 30.

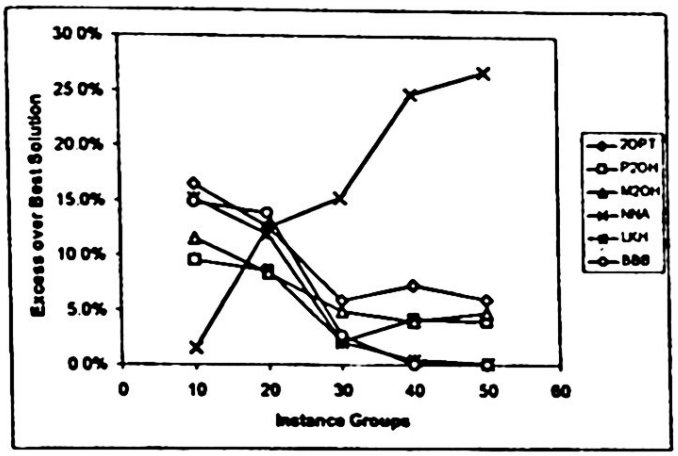


Fig. 3. Hamiltonian paths. Comparison of the 2OPT, P2OH, M2OH, NNA, LKH, and BBB algorithms

Once the restriction patterns have been ordered according to their similarity (either as a Hamiltonian circuit or path), we can now proceed to group them by means of a threshold function in classes of restriction patterns. The number and size distribution of the classes created by an enzyme is related to the information provided. The more groups are formed, the more information may be provided. However, at a given threshold, the less groups are formed the more striking differences between the classes of restriction patterns are emphasized. We analyzed, therefore, the formations of pattern classes and calculated the number of classes formed in excess over the best solution (see Fig. 4).

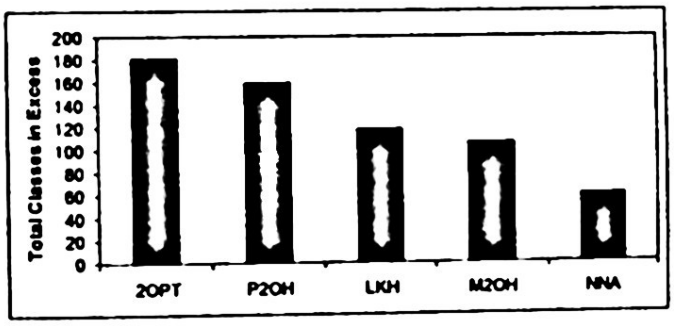


Fig. 4. Number of excess classes. Comparison of the 2OPT, P2OH, M2OH and NNA algorithms

First of all it should be noted that there was no difference in the use of either circuits or paths for the generation of classes, most likely due to the fact that the heaviest

edge, which has been removed from the circuit to generate a path, is always a border between two pattern classes. It has been hypothesized that by optimizing a Hamiltonian circuit based on a distance matrix, the transitions from one pattern to the next are smoothed out so that the number of pattern classes is reduced simultaneously. However, from the Fig. 4 it is apparent that these two events of optimization are not related: the algorithm that yields the worst result in generating circuits forms the smallest number of excess classes. The failure to optimize the number of classes resides in the fact that the target to be optimized is not directly represented by the matrix underlying the TSP. Therefore, we generated with the help of our threshold function a so called distinction matrix, where we report whether two restriction patterns are either similar (0) or belong to different classes (1). This procedure reduced drastically the complexity of the problem and it could now be solved easily by reducing the matrix after its transitive closure (i.e. if $d_{ij} = 0$ and $d_{jk} = 0$, then it must also be $d_{ik} = 0$ for the distances d between any combination i, j, k of three patterns).

4 Discussion and Conclusions

We have analyzed the behavior of several methods for approximate solutions of the TSP and a classification problem applied to our viral instances. As most of the methods have been developed to construct a Hamiltonian circuit, in our first series of experiments we analyzed how good they behave in finding them. As was to be anticipated, initial path algorithms do not perform well in comparison to the implemented local search or combined initial path and local search algorithms. According to our results the most efficient algorithm was the LKH, especially for larger instances. However, by solving the related TSP we could not optimize the classification problem. Nevertheless, the representation of the classification problem as a distinction matrix allowed us to reduce significantly the complexity of the problem and solve it after transitive closure by a simple reduction of the distinction matrices. However, it is worthwhile to note that the restriction patterns ordered by LKH serve as a better template to identify by visual comparison a given pattern obtained from a patient, thus the solution of the underlying TSP may still prove useful.

We have still to test whether there is a significant difference in the complexity of matrices derived from sequences of phylogenetically related organisms or generated at random.

In conclusion, we have demonstrated that in our case even simple traveling salesman heuristics are highly useful to address part of the sequence-typing problem.

References

1. Baase, S., Van Gelder, A.: Computer Algorithms. Introduction to Design & Analysis. 3rd Edition. Addison Wesley. 2002.
2. Garey, M.R., Johnson, D.S.: Computers and Intractability A Guide to the theory of NP-Completeness. W.H. Freeman and Co. 1979.
3. Panduro, A.: Biología Molecular en la Clínica. McGraw-Hill Interamericana. 2000.

4. Nacional Cervical Cancer Coalition NCCC. http://www.nccc-online.org/
5. Waterman, M.S.: Introduction to Computational Biology. Maps, sequences and genomes. Chapman & Hall/CRC. 2000.
6. Garcés Eisele, J., Castañeda Roldán, C.Y., Osorio Galindo, M., Gómez Gil, M.P.: El pequeño universo del Problema del Agente Viajero dentro de la Tipificación de Secuencias (DNA). CONIELECOMP03. University of the Americas Puebla. 2003.
7. Castañeda Roldán, C.Y.: Estudio Comparativo de diversos Métodos de Solución del Problema del Agente Viajero (PAV). Thesis. University of the Americas Puebla. 2000.
8. http://mailweb.udlap.mx/~ccastane/Tesiswww/Tesis2/AppletPOO-30Abr-1/example1.html
9. Castañeda Roldán, C.Y., Osorio Galindo, M., Gómez Gil, M.P., Garcés Eisele, J.: Híbrido MST-2Opt para la Solución Aproximada del Problema del Agente Viajero y la Tipificación de Secuencias de DNA en Modelos Virales. SAAEIE03. Universidad de Vigo España. 2003.
10. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison Wesley. 1974.
11. Ross, K.A., Wright, C.R.B.: Matemáticas Discretas. pHH Prentice Hall. 1990.
12. Mehlhorn, K.: Data Structures and Algoritms 2: Graph Algorithms and NP-Completeness. Springer-Verlag. 1984.
13. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. 6th Edition. McGraw-Hill Company. 1992.
14. Meza, O., Ortega, M.: Algoritmos y Estructuras. Universidad Simón Bolivar. Departamento de Computación y Tecnologia de la Información. Caracas, Venezuela. 2003.
15. Fernández Pellón Zambrano, R.: Desarrollo de Algoritmos para la Clasificación de Secuencias. Thesis. University of the Americas Puebla. 2002.
16. Lin, S., Kernighan, B.W.: An Effective Heuristic Algorithm for the Traveling-Salesman Problem. Oper Res 21:498-516 (1973).
17. Helsgaun, K.: An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. Eur J Oper Res 126:106-130. 2000.
18. Reingold, E.M., Nievergelt, J., Deo, N.: Combinatorial Algorithms Theory and Practice; Prentice-Hall, Inc. 1977.
19. Baldonado, M., Chang, C.C.K., Gravano, L., Paepcke, A.: Benchmarks for TSP. The Stanford Digital Library Metadata Architecture. Int. J. Digit. Lib. 1 (1997) 108–121