A New Parallel Approach to the Solution of the Traveling Salesman Problem

M. Kamali, F. Laleh, and A. R. Mirzai

SECOMP Research Lab., Dept. of Computer Eng.
Iran University of Science and Technology
Narmak, Tehran-16844, Iran

Abstract

In this paper, a parallel approach to the solution of one of the most popular NP-C problems, the traveling salesman problem, will be introduced. This method is based on a physical interpretation of the problem. The computational time in this approach is not dependent on the number of the cities in the TSP. The paper also describes the implementation of the proposed method using the conventional computers (Turing machines). The efficiency of the method when implemented by the Turing machines will also be investigated. Finally, we discuss about the time complexity of the proposed algorithm and future developments.

Keywords: NP-C problems, TSP, Parallel computation

I. Introduction

The traveling salesman problem (TSP) is a classical problem which is believed to be an NP-C problem [1], i.e., as the number of cities increases linearly the time required for the computation of the optimal path will increase factorially (exponentially). Therefore, traditional search approaches will fail to give a solution in a realistic period of time. Other methods have been introduced where the problem is defined as an optimization problem and it is solved using methods such as genetic algorithms [2] and Hopfield neural network [3, 4]. In recent years, new computational approaches, such as DNA computing [5, 6] have been proposed which are theoretical solutions but currently their implementations are difficult.

We believe that the main reason for the TSP to be an NP-C problem is the way we consider its computation. Therefore, in this paper a parallel approach for the solution of the TSP is introduced which is based on a physical interpretation of the problem. Section II gives the basic idea of the proposed method. In section III, the simulation of the method using conventional computers will be given and finally in section IV, the performance of the method will be investigated.

II. Physical Approach

Suppose there are n cities and it is required to find the shortest path from city “A.” A light beam is transmitted from city “A” to all the cities simultaneously. As the beam from city “A” is received by the other cities, it is modulated in order to indicate the path that the beam has traveled. The modulated beam is transmitted again to other cities. The first beam that has gone through all the cities and finally reaches its original source, i.e., city “A,” will indicate the shortest path.

It must be mentioned, that the method can be used to find the shortest paths for the traveling salesman starting from all the n cities simultaneously if initially all the cities transmit light beams. Also the solving time is not dependent on the number of the cities, but on the physical distances between the cities that the beam has to travel.

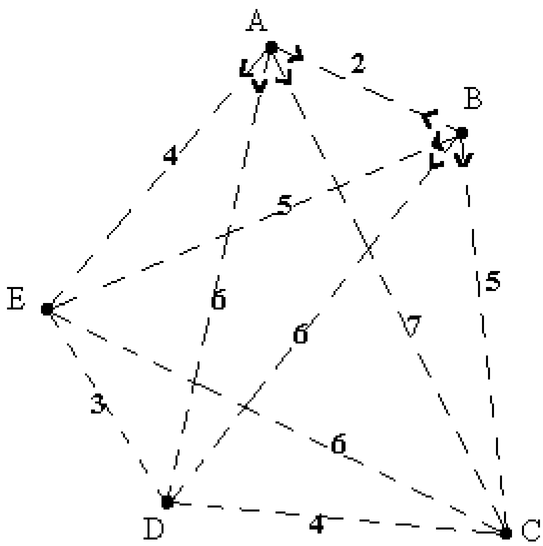

Fig. 1- A 5-City TSP with 5 Corresponding PPEs

Further author(s) information: <Farnad Laleh> lalehf@ariatg.com; http://www.ariatg.com

The best approach to implement the proposed algorithm will be to use n photonic processing elements (PPE) corresponding to the n cities in the problem. Each PPE modulates the beam, which it receives and sends them to the other PPEs. The main advantage of a PPE is its ability to modulate many beams simultaneously. Figure 1 shows five PPEs and city 'A' as the source.

III. Implementation Using Conventional Computers

As mentioned in the previous section, when we use n PPEs, the solving time is not dependent on the number of the cities, but it is dependent on the distances between them. An important question arises here is whether the proposed approach can be simulated using conventional computers? If yes, does the efficiency of the method change when compared to the photonic approach?

In response to the above question, it must be told that there are several methods for simulating the proposed approach. An approach would be to use a full connection computer network with n computers each corresponds to one of the cities. At a glance, one can find that the number of beams or signals generated will grow exponentially. So analyzing and processing of signals will not be practical. On the other hand, analyzing a full connection computer network is not easy and requires to solve many other problems. So using a sequential computer for simulation is easy and more practical but involves some new considerations.

From now, we investigate how the method can be simulated by means of a sequential computer. We investigate two major methods. One of them is referred to as *conventional* method because of its traditional concept and the other is referred to as *space tape* method. The conventional method is as follows.

Consider city 'A' as the starting city of the desired path. The paths from 'A' to any other cities will be registered as the *subpaths* of length (number of edges in a subpath) one. Then from each of the registered subpaths, we determine the distance from the last city in the subpath to the other cities, which have not been traveled and then register the new subpath of length two. This work is repeated in order to generate all paths of length n. The shortest registered subpath of length n will be the minimum tour, i.e., the answer of the problem.

In the method mentioned above, we follow a *breadth first search* strategy. In addition, the *depth first search* strategy can also be used. It is quite evident that the required time and space for these strategies increase in an exponential rate. Therefore, this simulation (conventional) is not a practical method for simulating the proposed approach.

For simulating the proposed approach, we must find a method in which the concept of the approach can be seen, i.e. as implementation with PPEs, the shortest path can be achievable sooner than the other paths, automatically. For this purpose, we use a concept, which is called *space tape*. A space tape is similar to several mosaics that lied beside each other in a straight line. We refer to each of the mosaics as a location in the tape.
Each location of the tape is used to register some events. Figure 2, shows the general form of the space tape. As can be seen, it is similar to natural numbers axis. According to the concept of space tape, an efficient method for simulation can be founded.

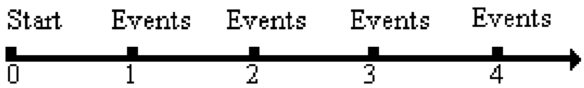

Fig. 2- General form of the Space Tape

Because of the discrete nature of the space tape, all distances between the cities in the TSP are considered as natural numbers. If they are real numbers, they must be normalized into natural numbers. The space tape algorithm is as follows.

An array in memory is considered as a space tape. Consider a pointer, which can point only to one location in the tape at a time. At the starting time of algorithm, this pointer points to location zero of the tape.

Consider city 'A' as the starting city of the shortest path. The distances between city 'A' to each of the other cities are determined. Then each of the subpaths will be registered at one of the locations of the tape corresponding to the distance which is traveled in the subpath, i.e., if distance from 'A' to 'C' is 12, then "AC" is registered at the 12th location of the tape. After registration, the pointer points to the next location, i.e., location one. If there are several subpaths at this location then from each of the registered subpaths, we determine distances from the last city in the subpath to the cities which have not been traveled and then register the new subpaths at their corresponding locations in the tape. After that, the pointer points to the next location. This work is repeated until the pointer points to the first subpath of length n. In this case, the pointer points to the shortest path.

The maximum number of subpaths, which are saved at a location in the tape, is called *width* of the tape and the length in which the first completed path is generated is called *length* of the tape. The length of the tape is equal to the length of the shortest path.

The proposed algorithm was implemented for the example in figure 1. In figure 3, all steps of the algorithm are shown schematically. As can be seen, at the 18th location of the tape, the first completed path is generated. So at this location, the algorithm comes to an end and the generated path is identified as the shortest path. In this example, width and length of the tape are equal to 7 and 18, respectively.

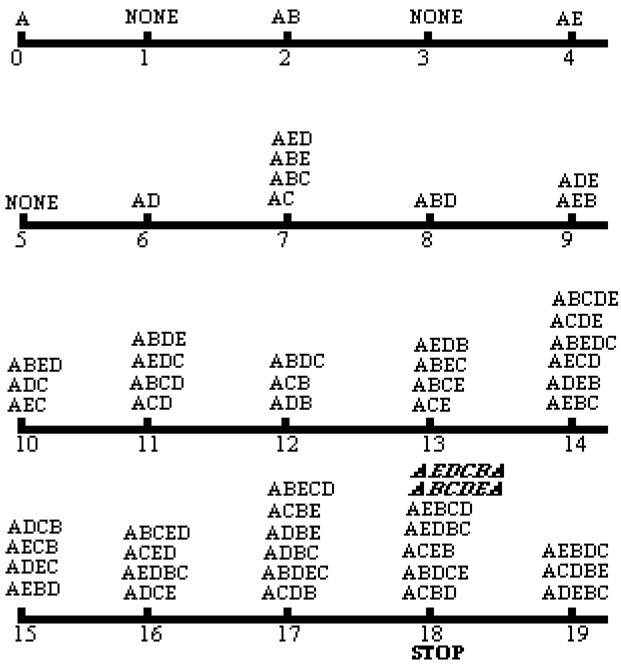

Fig. 3- Space Tape for Fig. 1

IV. Discussion

It is quite evident that the proposed algorithm is an exact solution to the TSP, i.e., it does not generate an approximate solution for the problem. The approximate solutions of the TSP are polynomial time algorithms [2, 3, 4]. But up to now, all of the exact algorithms have been proposed are in NP. The best exact solution had been gained by dynamic programming [1] and its time complexity is $O(n^2 \times 2^n)$.

In the proposed algorithm, the propagation of subpaths takes a time of $O(n^2 \times \text{width} \times \text{length})$, where width and length are the width and length of the tape, respectively. It is proved by using linear algebra rules that the number of subpaths at each location for *major applications* cannot exceed of $n \times$ number of edges or

$$n \times \frac{n \times (n-1)}{2},$$

which is equal to $O(n^3)$. Therefore, the required width and length for finding an exact solution is $O(n^3)$ and the time complexity of the algorithm is $O(n^5)$.

The same algorithm can be used for the Hamiltonian path problem (HPP). Its time complexity is also $O(n^5)$. The interesting result appears here is to achieve polynomial time algorithms for the TSP and HPP for major applications. It means most of the NP problems can be reduced to the P problems!

In the future works, the major area of research will be to find new computational structures, in which many AI problems can be solved automatically. The photonic structure for the TSP is one of them. However, the final goal is to establish a more general computational structure.

References

[1] C. H. Papadimitriou, "*Computational Complexity*," First Ed. Addison Wesley, MA, 1994.

[2] A. Homaifar, S. Guan and E. Liepins, "A New Approach on the Traveling Salesman Problem by Genetic Algorithm," in *Proc. of the Fifth International Conf. on Genethic Algorithms*, pp. 460-466, 1993.

[3] J. A. Freeman, "*Simulating Neural Network with Mathematica*," First Ed. Addison Wesley, MA, 1994.

[4] Page, Ed, G. A. Tagliarini, and F. Christ, "Optimization Using Neural Networks," *IEEE Trans. on Computers*, Vol. 40, No. 12, pp. 1374-58, Dec. 1991.

[5] L. M. Adleman, "Molecular Computation of Solution to Combinatorial Problems," *Science* 266, 1021-1024, 1994.

[6] R. J. Lipton, "DNA Solution of Hard Computational Problems," *Scienc*e 268, 542-545, 1995