State Space Reduction for the Symmetric Traveling Salesman Problem through Halves Tour Complement

Kamal R. Al-Rawi
Dept. of Computer Science, Faculty of Information Technology,
Petra University, JORDAN
E-mail:kamalr@uop.edu.jo

Abstract

The Traveling Salesman Problem (TSP) is an NP-hard. The state space increases exponentially with the number of nodes N. The number of paths is (N-1)!.

There are two main points in the TSP we took advantage of: The number of states N in a path is known in advance, which is the number of nodes in the tour; and touring each node only once. For symmetric TSP an optimal complete tour can be constructed by concatenating an optimal half path with one of its optimal half complement. This limits a partial path to extend no more than its half way to the complete tour.

This limitation reduces sharply the state space and the searched state space as well. This leads to reduction in both memory requirement and execution time which are the major challenges for computer scientist to tackle the TSP with exact algorithms.

Keywords: *Traveling Salesman Problem, TSP, state space reduction, state space search, optimal path.*

1. Introduction

The TSP represents N nodes that we have to find the optimal tour that starts at a node, visits every other node exactly once, and returns to the starting node. The TSP can be considered as a graph G with vertex set V and edge set E:

G(V,E); V={1,…,N};E={cost c_{ij}; i=1,…,N-1;j=1,…N}

For asymmetric graph $c_{ij} \neq c_{ji}$, however for symmetric graph, which is our concern here, $c_{ij}=c_{ji}$.

The TSP is an NP-hard. As the number of nodes increase the number of paths increases exponentially. For N nodes, the total number of paths are (N-1)!. Optimal solutions to small instances can be found in reasonable time; however, it will be very time consuming to solve large instances with optimal algorithms.

The TSP has many applications in the real words: areas of vehicle routing, workshop scheduling and computer wiring [1], logistics, genetics, manufacturing and telecommunications [2].

There are many exact algorithms for the TSP in the literature. However, since it is an NP-hard problem many approximation algorithms have been developed. Fully polynomial approximation can be solved by pseudo polynomial algorithms. Such algorithms require the upper and lower bounds for the optimal solution [3]. [4] Designed an algorithm to conduct a very special case of the TSP with distance one and two. Their approximated path is up to 8/7. The best known approximation algorithm for the general problem of TSP is (1+1/α), where α>1 [5]. As α increases better approximation will have, however, it will be more time consuming. [6] Claimed that they constructed an algorithm for the asymmetric TSP of O(logn). However, [7] proved that their algorithm is not accurate. For more details about such approaches see [8]. Many local search algorithms have been published. Such algorithms are needed when an acceptable (we mean non optimal nor even near optimal) path is required due to lack of time, since finding the optimal path is time consuming task. The well known Greedy algorithm construct the tour by adding the shortest edge available until building the complete tour. Its complexity is $O(n2log2(n))$. Insertion heuristic as its name indicates construct a tour by building a sub tour first then we expand the tour according to some heuristic measure. Its complexity is $O(n2)$. [9] Algorithm, worst-case ratio 3/2, *with complexity* $O(n3)$.

2. Objective

This work has been conducted to improve the performance of the optimal algorithms to find the optimal tour for the symmetric TSP. Performance improvement comes from reducing the state space, and reducing the searched state space. This leads to reduction in memory requirement and the execution time.

3. Algorithm development

Branch-and-Bound (B-n-B) is an exact algorithm that is typically used to find optimal solutions of optimization problems. However, employing it for the TSP for large N is very tedious task. We will try in this work to reduce the state space and searched state space to obtain the optimal path for the TSP using B-n-B algorithm.

3.1 Ordinary B-n-B optimal search for TSP

It is well known that number of paths for the TSP with N nodes is (N-1)!. The number of states in the whole state space is:

1+(N-1)+(N-1)*(N-2)+ (N-1)*(N-2)*(N-3)+…+ (N-1)*…* 4*3+3[(N-1)*(N-2)*…*2]…………………….....(1)

We can rewrite the above formula as:

$$S_N = t_0 + [\sum_{i=1}^{N-1} t_{i-1}(N-i)] + t_{N-2};\ t_o = 1; \quad \text{…………(2)}$$

Where each of the last three terms=(N-1)!

Table1 (column2) shows the number of states for different number of cities N using ordinary approach. Figure1 shows the state space, and the searched state space for ordinary approach, using the data in table2.

3.2 Complement approach (this work)

There are two main points in the TSP we have to take advantage of: The number of states N in the optimal path is known in advance, which is the number of cities in the tour; and touring each node only once. This let us branch a path no more than its half way to the complete tour. A complete tour can be constructed by concatenating a partial path with one of its complements halves partial paths. Table1 (column3) shows the number of states for different number of cities N using half tour complement approach. Figure2 shows the state space, and the searched state space for complement approach, using the data in table2.

In order to explain the complement approach we have to give an example. Let us say we have five cities to tour ABCDE. The complement halves partial paths for the half partial path ABC are ADE and ADE. These halves partial paths are complement for the half partial path ACB too. So, when we have the partial paths ABC or ACB and the partial paths of their complements ADE or AED then a complete tour can be constructed. For seven cities ABCDEFG, the complement halves partial paths for the half partial path ABCD are AEFG, AEGF, AFEG, AFGE, AGEF, and AGFE. These halves partial paths are complement for halves partial paths ABDC, ACBD, ACDB, ADBC, and ADCB. So, a complete tour can be constructed by concatenating a partial path from (ABCD, ABDC, ACBD, ACDB, ADBC, ADCB) with a partial path from their complements (AEFG, AEGF, AFEG, AFGE, AGEF, AGFE).

Limiting the partial path branching just to the half way of the complete tour reduces the state space of the TSP and decreases sharply the searched state space. This leads to reduction in both memory requirement and execution time which are the major challenges for computer scientist to tackle the TSP for optimal path.

The number of states in the state space is:

1+(N-1)+(N-1)*(N-2)+ (N-1)*(N-2)*(N-3)+…+ (N-1)* (N- 2)*…*(N-N/2)………………………(3)

We can rewrite the above formula as:

$$S_N = t_0 + [\sum_{i=1}^{N/2} t_{i-1}(N-i)] + t_{N-2};\ t_o = 1 \quad \text{……..(4)}$$

Table1 (column3) shows the number of states, using complement approach, for different values of N. In addition to the reduction in the number of states, number of paths in the state space is reduced by 50%.

Table1: The number of states in the state space for ordinary and complemented approaches for different values of N.

No of nodes	Ordinary	Complemented
4	22	10
6	446	86
8	18,740	1,100
10	1,349,290	18,731
12	148,421,912	397,112
14	23,153,818,286	10,057,646
16	4,862,301,840,076	295,891,276
18	1.32254610050069E+15	9,909524,690
20	4.52310766371236E+17	372,065,866,400

The reduction in the state space is less than the ordinary approach by:

(N-1)*…*(N-N/2-1)+ (N-1)*…*(N-N/2-2)+…+3(N-1)*…*2 ………………………….……………..(5)

Table2: A hypothetical cost to travel among four cities that used for figure1 and figure2.

	A	B	C	D
A	0	99	167	106
B		0	266	205
C			0	61
D				0

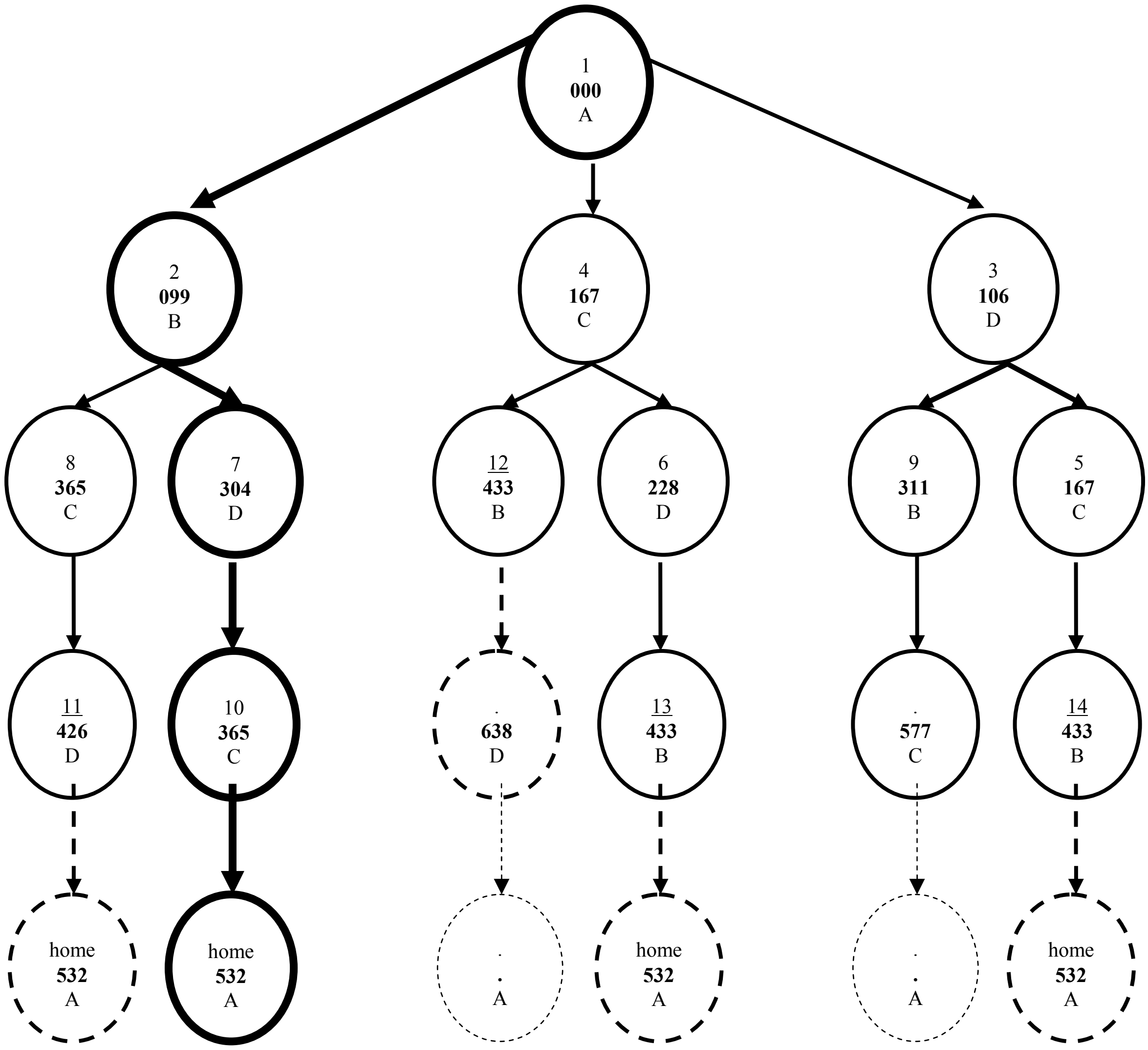

Figure1: The traveling salesman problem (TSP) state space for 4 cities. Number of states is 22. Number of paths is (N-1)!=3!=6.

The first number in a node represents the order of the node in branching. The second number represents the accumulated cost at that partial path. The character represents a city name.

The solid lines represent the searched state space to find the first complete path using B-n-B algorithm. The total searched states are 16, which represents 72.7% of the whole state space. The first complete optimal path ABDCA is shown in heavy solid line. The cost is=532. The heavy dotted line represents the branching all partial path with cost less than what we found in order to assure an optimal path. The total searched states are 20, which represents 90.9% of the whole state space.

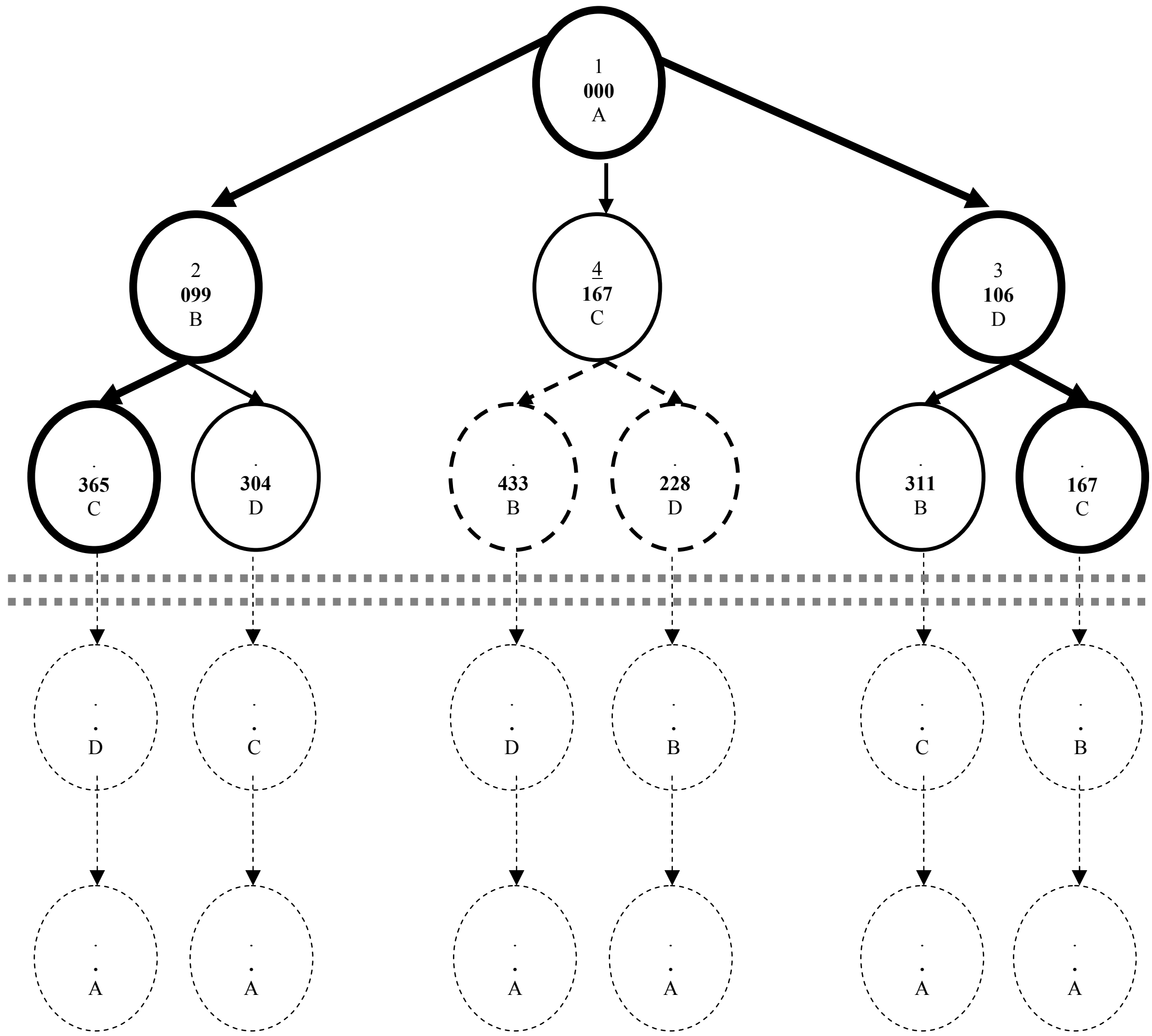

Figure2: Same as figure1 but we do not branch any partial path more than its half complete tour. This reduces the whole state space sharply. The number of states in the new state space is 10 (all states above the double dotted horizontal line), which represents 45.5% of the whole original state space.

The total searched states are 8, which represents 36.4% of the whole state space. The first complete path is shown in heavy solid line. It constructed by concatenation the half path ABC with its complement half path ADC to have the complete path ABCDA. The cost is=532. The heavy dotted line represents the branching all partial path with cost less than what we found in order to assure an optimal path.

This is a very large reduction for the state space. This reduction increases as N increases. Figure3 shows the percentage of the state space using complement approach relative to the ordinary approach. It decreases exponentially with number of nodes N. For a problem with 20 nodes we reduced the state space down to 0.000008%. The searched state space is reduced too since the optimal tour is constructed by concatenating the optimal half path with its optimal complement.

Since we are extending the partial path with lowest cost, the concatenation of the first two complemented haves represents the optimal tour.

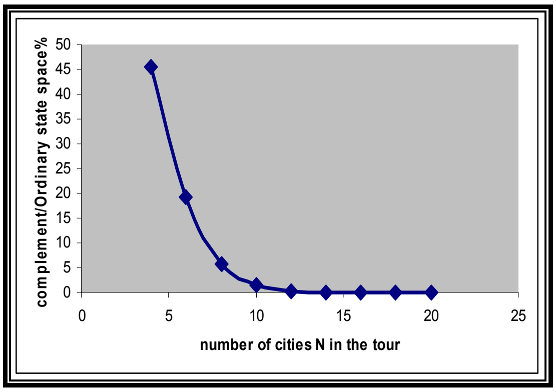

Figure3: complemented to ordinary percentage of the state space. It reduced exponentially with number of nodes N.

5. Conclusion

The complete tour is constructed by concatenating the optimal half partial path with its optimal half partial path. This limit a partial path no more than its half way to the complete tour. This leads to: 1- reducing the state space, 2- reducing the searched state space, and 3- reducing number of paths in the state space. This leads to reduction in memory requirement and the execution time.

5. References

[1] Lawler, E. L.; Lenstra, J. K.; Kan, A. H. G. R.; and Shmoys, D. B. (1985). *The Traveling Salesman Problem*. Essex, England: John Wiley & Sons.

[2] Applegate, D. L., Bixby, R. E., Chvatal, V., and Cooke, W. J., (2007), "The travel salesman problem", Princeton Univ. Press.

[3] Hassin, R, (1992), "Approximation schemes for the restricted shortest path problems." Math. Oper. Res., Vol.17, No,1, pp:36-42.

[4] Berman, P., and Karpiniski, M., (2005), "8/7 approximation algorithm for (1,2)-TSP," Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 69.

[5] Arora, S., (1998), “Polynomial time approximation schemes for Euclidian traveling salesman and other geometric problems”, *J. of ACM*, Vol. 45, No. 5, pp: 753-782.

[6] Chekuri, C., and P´al, M., (2007)," An *O*(log*n*) Approximation ratio for the asymmetric traveling salesman path problem," Theory of Computing, Vol.3, pp:197-209.

[7] Nagarajan, V., (2008), " On the LP relaxation of the asymmetric traveling salesman path problem," Theory of Computing,Vol.4, pp:191-193.

[8] Sahni, S., (1976),"General techniques for combinational approximations," Oper. Res., Vol.25, pp:920-936.

[9] Christofides, N., (1976)," Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report, GSIA, Carnegie-Mellon University.