Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Sciences and Engineering

Elixir Comp. Sci. & Engg. 62 (2013) 17774-17777

ISSN: 2229-712X

Parallel 2-Approximation algorithm to solve Travelling Salesman Problem

Rajnish Dashora* and Shiven Saiwal

School of Computing Sciences and Engineering, VIT University, Vellore, India.

ARTICLE INFO

Article history:
Received: 16 July 2013;
Received in revised form:
25 August 2013;
Accepted: 9 September 2013;

Keywords
Travelling Salesman Problem,
Approximation algorithm,
Parallel Algorithms,
Metric TSP,
Optimal tour,
Approximation ratio,
Multi-core processors,
OpenMP.

ABSTRACT

Travelling salesman problem is an NP complete problem and can be solved using approximation algorithms. It is a minimization problem starting and finishing at a specified vertex after having visited every other vertex exactly once. Often, the model is a complete graph. An algorithm that returns near-optimal solutions is called approximation algorithms. Through analyzing the Metric TSP, the performance of the approximation algorithm can be improved significantly using parallelization that implements a program to find a path with approximately minimum travelling cost through parallel depth-first search and the result can be verified using graphical analysis of spanning trees.

Introduction

Travelling salesman problem predicts the shortest possible route to connect different cities based on given distances between each city. Here, each city should be traversed only once and finally the person tracing the route should reach the starting point of the journey with minimal cost.

Solving a TSP requires application of several graph algorithms which are helpful in the analysis of the problem and hence, in finding the optimal solution of TSP.

We call the solution optimal because TSP is an NP hard problem which means that TSP cannot be solved in polynomial time, or, it is at least as hard as the hardest problem in NP [6]. TSP is a challenge to the field of algorithms.Many researches continue to find the solution for TSP and hence for the set of NP hard problems.

But optimal solutions can be found using algorithms such as

1) Exponential Time Algorithm
2) Pseudo Polynomial Time Algorithm

Approximation algorithm

Approximation algorithms are useful in finding near optimal solutions which are helpful in predictions, decision problems, operational research and search problems. These approximation algorithms have the output which do not deviate much with the optimal solution for the given problem and solves the purpose. Approximation algorithms are probably fast and run in polynomial time and provide us with a solution somewhat close to the optimal solution.

Approximation algorithms are used for maximization or minimization based on the problem. When it comes to Travelling Salesman Problem, it is a minimization problem as the task is to minimize the total cost of a round trip starting and finishing at a specified vertex after having visited each other vertex exactly once. Often, the model is a complete graph (i.e. each pair of vertices is connected by an edge). If no path exists between two cities, adding an arbitrarily long edge will complete the graph without affecting the optimal tour [12].

In the symmetric TSP, the distance between two cities is the same in each opposite direction, forming an undirected graph. This symmetry halves the number of possible solutions while in the asymmetric TSP, paths may not exist in both directions or the distances might be different, forming a directed graph.

For the problem of input size 'n' the approximation ratio ρ(n) is defined as the maximum of the ratio of the approximate solution to that of the optimal solution and optimal solution to the approximation .That is,

$$max\left(\frac{C}{C*}, \frac{C*}{C}\right) \le \rho(n) \qquad [7]$$

Hence the algorithms with the approximation ratio of ρ (n) are called as ρ (n) - Approximation algorithm. Approximation ratio is always greater than 1. Also in case of ρ (n) = 1 the algorithms gives the optimal solution.

Metric TSP

Here we consider a special case of Travelling Salesman Problem named Metric TSP which is a NP complete problem [1]. Metric TSP has some constraints,

If graph G denotes the network of the cities where all the vertices represent cities and the edges represent the path between them, First It should not have any self-loop; second

there is exactly one path [5] which is bi-directed with minimal edge cost between two vertices. Also, if a direct path exists between the two vertices then the path is smallest when compared with any other indirect path that is through other nodes implies vertices should hold triangle inequality constraint [2].

These constraints can be written mathematically as,

Graph G specified as an n x n Matrix D in which D [i, j] denotes the distance between the vertices I and j such that D forms a metric, i.e.

1) For all i D[i,i]=0

Tele:
E-mail addresses: dashora.rajnish@gmail.com

2) For all i.j D[i,j]=D[j,i]
3) For all i,j,k D[i,j] <= D[i,k] + D[k,j]

Output of the approximation algorithm will be a cycle in the graph passing through all vertices exactly once such that the sum of distances associated with the edges in the cycle is as small as possible.

Advantage of using approximation algorithm for TSP

Approximation algorithms are useful in approximation of optimal traversal for a given set of cities or nodes. When the accurate results are not necessary means you don't need to have the exact solution but an approximation or a solution near to the optimal solution is sufficient and the time matters[11]. As stated above in this TSP is an NP –Hard problem which do not have any algorithm which can find the optimal solution in polynomial time. Approximation algorithms provide with a near optimal solution with in polynomial time [4].

Related works and Existing Solution

Metric TSP is NP-complete hence there exists a 2-approximation algorithm for to approximate the solution with the approximation ratio $\rho(n)$ of 2where n is the number of cities to traverse. For this approximation algorithm, the weight of a minimum weight L spanning tree T of G (with D the weight matrix) is a lower bound on the length of an optimal tour on G which is proved if any edge is removed from that spanning tree the graph gets disconnected. Hence the cost of the path is at least equal to the weight of spanning tree.

The upper bound is the length of depth first search of the spanning tree with the minimal weight. That is twice the weight of the spanning tree equal to 2L.Hence the optimal solution C lies in {L, 2L} [3].

$L < C \leq 2L$

Existing serial 2-Approximation Algorithm

1) Find T = MST of G for weight matrix D. Weight(T) = L
2) Find E = Sequence of vertices visited in DFS of T
3) V appears more than once in E, Delete first appearance.
4) Repeat the previous step while possible.
5) Return E.

There is not a specific algorithms specified to use for the generation of spanning while solving TSP also the serial depth first search having high complexity of O(|V|+|E|) .Over all the 2-approximation algorithm for the metric TSP problem[8] is a feasible and operates very quickly still the existing serial algorithms is not as fast as it should be while executed on the multicore system environment because it cannot utilize the CPU's or threads available freely for computation and hence over loading a single core or processing unit which is a disadvantages of existing solution[15].

Proposed solution

When it comes to speed and time complexity the above serial algorithm takes more time to execute and not a feasible solution to the problem for getting approximation for metric TSP. This paper improve the 2-approimation algorithm and proposes a new algorithm for metric TSP which is again a 2-approimation but it takes lesser time to execute and to get a solution for minimization problem of Metric TSP [9].

This algorithms generates the spanning tree using Prim's algorithms which has time complexity of O (E + V log V), usefulness of Prims algorithms comes when there are more number of edges in to the graph ,it performs faster[10]. After getting the spanning tree, the tree is traversed using depth first traversal in which use of parallelism and recursion provides with an efficient algorithm for the Metric TSP also the complexity of the algorithm depends on the number of processor or core executing the algorithm [13].

Parallel 2-approximation algorithm for metric TSP

Parallel algorithm for the 2-approximation algorithm of metric TSP

Graph G denotes the network of cities to traverse
D is the Adjacency cost matrix of G
C array of vertices traversed as the solution

Algorithm:

```
1 sol_tsp (graph G with matrix D)
2 Generate spanning tree T for the Graph G with the root node a.
3 C [0] = a;
4 Call p_approx_tsp (a);
5 Store the return of the function call in C array starting from C [1].
6 Return C and cost of traversal.
7 //p_approx_tsp function used above will be
8 p_approx_tsp (p)
9 Z: 2-d array where Z[α] denotes the return of the function call p_approx_tsp (α)
10    i=0;
11    //Execute for in parallel or assign the call p_approx_tsp (α) to work pool to be executed by the threads running in parallel.
12    For all nodes α connected to p
13    call p_approx_tsp (α)
14    store return in Z[α]
15    increment i by 1;
16    //each p_approx_tsp call is independent hence can be executed in parallel
17    y: array of nodes as solution by the function
18    k=0;
19    for j = 0 . . . i
a.    h=0;
b.    while Z[j] not empty
  i.    y[k] = Z[j] [h];
 ii.    increment k by 1;
20    end for
21    y[k] =a;
22    return y;
```

Flow of algorithm

The above parallel algorithm for the approximation of metric TSP provides us with the solution with the approximation ratio ρ of 2 but the time to find the solution is reduced significantly.

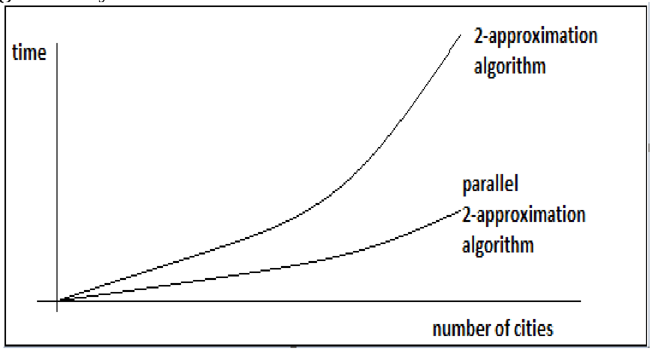

Figure 1: Time v/s Number of cities

The load is balanced on the processors which improves the performance of the code and reduces the overhead of complex calculations on a single processor. A small set of cities is traced for the solution for understanding the flow of parallel algorithm.

Step 1:

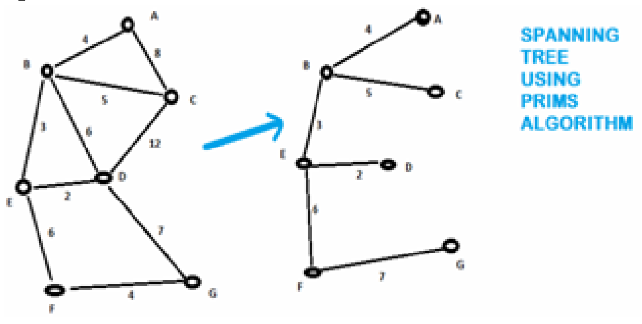

Figure 2: Spanning Tree using Prims Algorithm

With the root node A when the algorithm further proceeds we found the approximate solution for the metric TSP problem in a parallel and recursive manner.

Hence we get the solution as,

A G F D E C B A and the cost will be 53

Step 2:

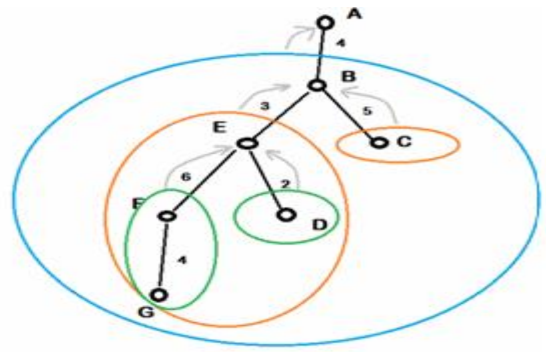

Figure 3: Finding the solution in recursive manner

This will be the approximately optimal solution,

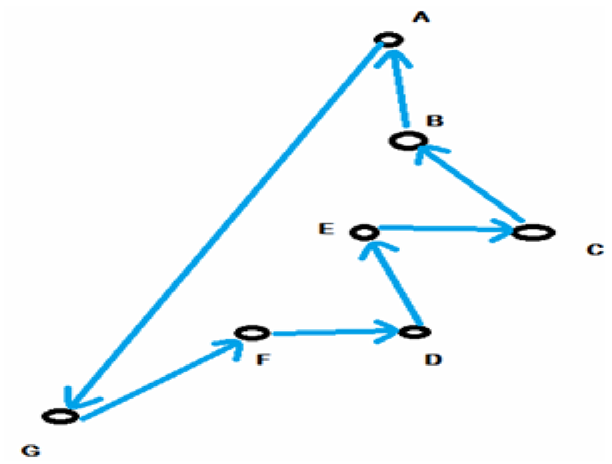

Figure 4: Approximate optimal solution

Further for the larger problem size the algorithm is scalable as well as for multicore processors with n number of cores and hence it can be improvisation of 2-approximation.

Experimental Work

Though 2-approximation algorithm is not an optimal approximation for travelling salesman problem yet when execution time for the serial 2-approximation and parallel 2-approximation are analyzed on multi-core processors this parallel algorithm is found to have a quicker response. Parallel implementation is done using openMP and the execution time for the serial and parallel implementations of the two approximation algorithms are analyzed using VTune Amplifier XE on core i5 (4 threads) and core i7 (8 threads) processors for 7 cities which are illustrated in Figure-4. 7 has been chosen as it is the largest prime number less than 10and the results of TSP solution are most notably dynamic when dealing with prime number of cities. Results are tabulated as,

Processor	Number of threads	Number of cities	Speed Up as compared to single core processor
Intel core i5 4570s processor	4	7	2.1
Intel core i7 4770s processor	8	7	4.33

Table 1: Summary of performance analysis of parallel 2-approximation algorithm

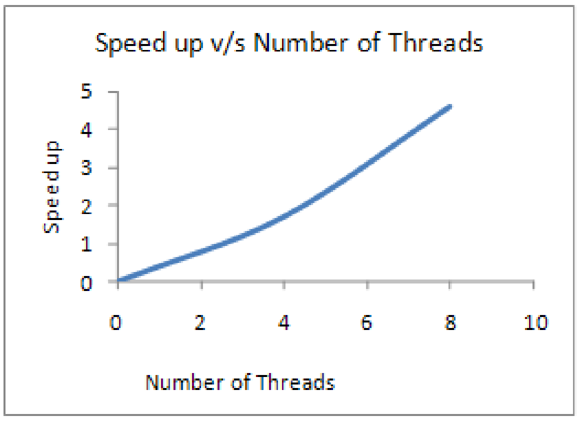

Figure 5: Speed up v/s Number of Threads

Results and conclusion

Hence the parallel 2-approximation algorithm will provide us with a near optimal solution with a significant reduction in the time complexity. The use of Prim's algorithm for generation of spanning tree helps to manage the dense graphs and the parallelization. Also, recursive computation further improves the computational performance. An overall improvement of 52.38% and 76.9% was realized on Intel corei5 and corei7 processors respectively.

Future work

Future works include the improvisation in the complexity of this algorithm. There exist 4/3 approximation and 1.5 approximation algorithms for metric TSP which are highly complex and hence using parallelization for those algorithms may help in reducing their complexity and execution time. Optimization of general TSP without constraints and further more applications of TSP, parallel algorithms and graph theory will be areas of research.

References

[1]G. Ausiello, A. Marchetti-Spaccamela, and M. Protasi. Towards a unified approach for the classification of NP-complete optimization problems.TheoretComput. Sci., 12:83–96, 1980.

[2] M. A. Bender and C. Chekuri. Performance guarantees for the TSP with parametrized triangle inequality. In Proc. WADS'99, volume 1663 of Lecture Notes in Computer Science, pages 80–85. Springer, 1999.

[3] H.-J. B¨ockenhauer, J. Hromkovi˘c, R. Klasing, S. Seibert, and W. Unger.An improved lower bound on the approximability of metric TSP and approximation algorithms for the TSP with sharpened triangle inequality. In Proc. STACS'00, Lecture Notes in Computer Science, pages 382–394. Springer, 2000.

[4] M. Demange, P.Grisoni, and V. T. Paschos. Diherential approximation algorithms for some combinatorial optimization problems.Theoret.Comput. Sci., 209:107–122, 1998.

[5] Kaplan, H.; Lewenstein, L.; Shafrir, N.; Sviridenko, M. (2004), "Approximation Algorithms for Asymmetric TSP by Decomposing Directed Regular Multigraphs", In Proc. 44th IEEE Symp. on Foundations of Comput. Sci, pp. 56–65.
[6] Woeginger, G.J. (2003), "Exact Algorithms for NP-Hard Problems: A Survey", Combinatorial Optimization – Eureka, You Shrink! Lecture notes in computer science, vol. 2570, Springer, pp. 185–207.
[7] www.wikipedia.com
[8] Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook (2006), “The Traveling Salesman Problem.”
[9] R.E.Burkard, “Travelling salesman and assignment problems: Asurvey”, Annals of Discrete Mathematics 4(1979) , 193-215 TSPLIB95/ ATSP.html
[10] Gilles brassard and paulbrately, “Fundamentals of algoritms”.
[11] Ellis horowitz,sartajsahni and sanguthevarrajasekaran, “Fundamentals of computer algorithms”.
[12] GramaAnanth, Gupta Anshul, Kapyris George, Kumar Vipin, Introduction to
Parallel Computing, Second Edition, PEARSON, Addison Wesley, 2003.
[13] Quinn Michael, Parallel programming in C with MPI and OpenMP, McGraw-Hill,2003
[14]S. Arora, Nearly linear time approximation schemes for Euclidean TSP and other geometric problems, J. of the ACM, 45(5):1{30, 1998.
[15]Peter S Pacheao,An Introduction to Parallel Programming, Morgan Kaufmann Elsevier 2011.