5/4-approximation for Symmetric TSP

Madhusudan Verma¹, Alok Chauhan*², Vijayakumar V³

¹madhusudan.verma2015@vit.ac, ²alok.chauhan@vit.ac.in, ³vijayakumar.v@vit.ac.in

¹, ², ³ VIT Chennai, India

Abstract: Travelling Salesman Problem (TSP) is one of the unsolved problems in computer science. TSP is NP-Hard. Till now the best approximation ratio found for symmetric TSP is \( \frac{3}{2} \) by Christofides’ Algorithm more than thirty years ago. There are different approaches to solve this problem. These range from methods based on neural networks, genetic algorithm, swarm optimization, ant colony optimization etc. The bound is further reduced from \( \frac{3}{2} \) but for graphic TSP. A factor of \( \frac{13}{9} \) was found for Graphic TSP. A newly proposed heuristic called k-RNN is considered here. It seems from experimental results that \( \frac{5}{4} \) is the approximation ratio. A performance analysis is done for this heuristic and it confirms experimental bound of \( \frac{5}{4} \).

Keywords: Christofides’ algorithm, 5-degree Minimum Spanning Tree, Approximation Ratio, Corresponding Edge, Symmetric TSP, k-RNN.

1. Introduction

TSP asks to visit node exactly once and all nodes in a graph. TSP is APX-Hard. However, Held-Karp LP relaxation is conjectured to have bound of \( \frac{4}{3} \). There is more general form of this problem known as Travelling Salesman Path Problem (TSPP) in which it is needed to find a path from two given points visiting all the nodes of graph exactly once. The best known algorithm for this problem is given by Hoogeveen. The bound found by this method is \( \frac{5}{3} \). However it is conjectured to have an integrality gap of \( \frac{3}{2} \) by the Held-Karp relaxation for this problem. One of the natural ways to attack this problem is to consider special cases of this problem. The most interesting is the Graphic TSP/TSPP. In Graphic TSP, we need to find a minimum cost circuit visiting nodes at least once. We can apply similar formulation to Graphic TSPP case. They are APX-hard, there are standard examples showing that the Held-Karp relaxation has a gap of at least \( \frac{4}{3} \) in the TSP case and \( \frac{3}{2} \) in the TSPP case. A significant progress has been made in approximating the graphic TSP and TSPP in recent times. Oveis Gharan gave an approximation of \( \frac{3}{2} - \varepsilon \) for Graphic TSP [3]. In which first an optimal solution of LP relaxation is computed. Then LP solution as \( \lambda \)-uniform distribution of spanning trees is written, followed by sampling of a Spanning Tree T from this distribution and at last a minimum cost matching on odd degree vertices of T is added. Following that, Mömke and Svensson
obtained a significantly better approximation ratio of \( \frac{14(\sqrt{2} - 1)}{12\sqrt{2} - 13} \approx 1.461 \) for graphic TSP, as well as factor \( 3 - \sqrt{2} + \epsilon \approx 1.586 + \epsilon \) for graphic TSPP, for any \( \epsilon > 0 \). Above approach uses matching in a truly ingenious way. Instead of adding edges of a matching to a spanning tree to make it Eulerian, as it was done in previous approaches, the matching edges are added and removed. This process is guided by a so-called removable pairing of edges which essentially encodes the information on which edges can be simultaneously removed from the graph without disconnecting it. An approximation ratio of \( \frac{5}{4} \) for symmetric TSP is found in present work. This algorithm is simple to understand as well as easy to implement.

2. Motivation

The challenge to improve the approximation ratio obtained by Christofides is a big motivation. Since TSP has much wider applications, the need to work on this problem is felt.

3. Comparison with related work

| Algorithm                                      | TSP Type | Approximation Ratio | Time Complexity |
|------------------------------------------------|----------|---------------------|-----------------|
| Christofides                                   | Symmetric| \( \frac{3}{2} \)    | O(n^3)          |
| Truncated Generalized Beta distribution Based on Christofides’ Algorithm [9] | Symmetric| \( 1 + \frac{1}{2} \left( \frac{\alpha+1}{\alpha+2} \right)^{K-1} \) where \( \alpha >> 1 \) is the shape parameter of TGB and \( K \) is the number of iterations | O(n^4) |
| 2-RNN [4]                                      | Symmetric| \( \frac{5}{4} \)    | O(n^3)          |
| Random Sampling [3]                            | Graphic  | \( \frac{3}{2} - \epsilon \) | unknown         |
| Novel use of matching [5]                     | Graphic  | \( \frac{13}{9} \)   | unknown         |
By ear-decomposition optimized using forest representations of hyper graphs[6]

| By ear-decomposition optimized using forest representations of hyper graphs[6] | Graphic | \( \frac{7}{5} \) | Polynomial time |
|---|---|---|---|

Finding a cycle cover with relatively few cycles for cubic bipartite graph [7]

| Finding a cycle cover with relatively few cycles for cubic bipartite graph [7] | Graphic | \( \frac{9}{7} \) | Polynomial time |
|---|---|---|---|

By consecutive path cover improvements [8]

| By consecutive path cover improvements [8] | Metric | \( \frac{8}{7} \) | Polynomial time |
|---|---|---|---|

**Table1:** Comparison of various TSP algorithms

4. **2-RNN Heuristic**

The heuristic is inspired by a new human centric co-existential philosophy propounded by Late Sri A Nagraj, India [10, 11]. Before explaining about 2-RNN, first let’s understand its general form which is k-RNN. The algorithm consists of the following steps [4]:

Step 1: For every permutation of the k vertices \( v_1, v_2, \ldots v_k \) create the partial tour \( T = (v_1, v_2, \ldots v_k) \) and mark the vertices \( v_1, v_2, \ldots v_k \) as visited.

Step 2: Set \( i = k \). While there are unvisited vertices left: Select \( v_{i+1} \) as the nearest unvisited neighbor of \( v_i \) and append \( v_{i+1} \) to \( T \). If there are multiple nearest neighbors, select any. Mark \( v_{i+1} \) as visited and increment \( i \) by 1.

Step 3: Among all \( \frac{n!}{(n-k)!} \) tours found, select the shortest as the result.

2-RNN is k-RNN with k=2.

Now at first glance it seems similar to nearest neighbor algorithm, but the difference here is that instead of starting from a node, here we start from an edge.
5. Preliminaries

**Statement 1** [1]: For every set of points in the plane, there exists a degree-5 MST.

**Algorithm 1**[1]: Algorithm for converting 5-degree MST to a spanning tree of degree 4

1. Root the MST at a leaf vertex r.
2. For each vertex \( v \in V \) do
   
   Compute the shortest path \( P_v \) visiting \( v \) and all its children.
3. Return \( T_4 \), the tree formed by the union of the paths \( \{P_v\} \).

**Theorem 1** [1]: Let MST be a minimum spanning tree of a set of points in \( \mathbb{R}^2 \). Let \( T_4 \) be the spanning tree output by the algorithm 1, then

\[
W(T_4) \leq 1.25 \times W(\text{MST})
\]

**Statement 2** [2]: Let TREE be the Minimum Spanning Tree of the given graph and OPTIMAL be the length of optimal tour in graph visiting and starting at the same node then,

\[
\text{Cost (MST)} \leq \left(1 - \frac{1}{n}\right) \text{OPTIMAL}
\]
Experimental Results for 2-RNN:

| Dataset | Optimum | 1-RNN | 2-RNN |
|---------|---------|-------|-------|
| a280    | 2579    | 2975  | 2953  | 14.50 |
| berlin52| 7942    | 8181  | 8.47  | 7968  | 5.05 |
| ber127  | 118282  | 133953| 12.25 | 128589| 8.71 |
| brunel  | 25939   | 27384 | 7.83  | 27213 | 7.16 |
| ch100   | 1059    | 8880  | 305.00| 2020  | 3.59 |
| ch130   | 6110    | 7129  | 16.68 | 6063  | 12.98|
| ch150   | 6538    | 7413  | 8.96  | 7113  | 8.96 |
| di291   | 9608    | 56861 | 15.51 | 56681 | 15.51|
| di55    | 6218    | 73369 | 18.00 | 72554 | 18.78|
| di18    | 17820   | 17620 | 11.66 | 17405 | 10.30|
| d493    | 35032   | 40186 | 14.81 | 40186 | 14.81|
| d657    | 48912   | 60174 | 25.03 | 59310 | 21.26|
| dantzig42| 699    | 864   | 23.61 | 826   | 18.17|
| eil101  | 629     | 746   | 18.60 | 743   | 18.12|
| eil51   | 426     | 482   | 10.15 | 472   | 10.80|
| eil76   | 538     | 608   | 15.01 | 598   | 11.15|
| fl1400  | 20117   | 25115 | 24.78 | 24719 | 22.82|
| fl17    | 11461   | 13987 | 17.08 | 13666 | 16.90 |
| fr26    | 937     | 965   | 2.90  | 950   | 2.35 |
| gr262   | 2478    | 2823  | 18.71 | 2767  | 18.36|
| gr120   | 6438    | 8438  | 21.55 | 8335  | 20.07|
| gr17    | 2815    | 2178  | 4.46  | 2178  | 4.46 |
| gr21    | 2707    | 3063  | 10.93 | 2958  | 9.27 |
| gr24    | 1732    | 1553  | 22.09 | 1400  | 10.06|
| gr48    | 5945    | 5840  | 15.74 | 5581  | 10.21|
| h148    | 11461   | 12137 | 5.90  | 12031 | 4.97 |
| kroA100 | 21282   | 24698 | 16.05 | 24582 | 15.51|
| kroA120 | 20524   | 31479 | 18.08 | 31320 | 18.08|
| kroA200 | 29398   | 34543 | 17.62 | 34543 | 17.62|
| kroB100 | 22141   | 25884 | 16.01 | 25255 | 14.06|
| kroB120 | 20130   | 31411 | 20.08 | 31224 | 20.64|
| kroB200 | 29437   | 35386 | 20.22 | 35983 | 19.46|
| kroC100 | 20794   | 23660 | 16.03 | 23603 | 13.75|
| kroC160 | 21248   | 24853 | 16.71 | 24603 | 15.54|
| knn150  | 20938   | 24782 | 12.30 | 24445 | 13.77|
| lin105  | 14379   | 16935 | 17.78 | 16147 | 13.30|
| lin138  | 42029   | 49201 | 17.06 | 49201 | 17.06|
| lin5318 | 41345   | 49318 | 18.00 | 49201 | 19.00|
| nrw1379 | 56368   | 68531 | 21.00 | 68738 | 19.84|
| p60     | 3443    | 43627 | 24.20 | 42935 | 23.94|
| pr261   | 2703    | 3279  | 18.68 | 3269  | 15.31|
| prb1173 | 56892   | 70115 | 20.28 | 69685 | 21.43|
| prb442  | 59778   | 58950 | 16.09 | 58682 | 15.57|
| r148    | 103159  | 138921| 21.04 | 139749| 19.84|
| st1012  | 92650   | 94863 | 1.55  | 93981 | 1.44 |
| st175   | 21407   | 22000 | 2.77  | 21906 | 2.33 |
| st35    | 48450   | 56866 | 3.27  | 50032 | 3.27 |
| swiss42 | 1273    | 1437  | 12.88 | 1425  | 11.94|

**Figure 1:** Results for 48 instances of the Symmetric TSP taken from TSPLIB [4].

6. Performance Analysis of 2-RNN

**Lemma 1:** Let T be the tree obtained after removing an edge e from the minimum tour T obtained by 2-RNN for a given complete graph G, and let M be the 5-degree MST for the graph G (Statement 1), then

\[ \text{Cost}(T) \leq \text{Cost}(M) \]

\[ \text{Cost}(T_i) \leq \text{Cost}(M) \]
Proof: We define an edge $e_c$ in $M$ as a corresponding edge for an edge $e$ in $T_i$ such that both $e_c$ and $e$ have the common vertex in $M$ and $T_i$ respectively, and $e_c \geq e$ otherwise we could replace $e$ by $e_c$ in $T_i$ and that will contradict the logic of 2-RNN because 2-RNN at each step selects the shortest next edge except when such selection results into a cycle, in that case 2-RNN checks for second shortest next edge and likewise. But such $e_c$s which cause formation of cycle in $T_i$ will not exist in $M$ in the first place as $M$ also avoids cycle. Therefore Cost ($T_i$) $\leq$ Cost(M).

Theorem 2: Let $T$ be the optimal tour $T$ by 2-RNN and $M$ be the 5-degree, MST of the given graph, then

$$\text{Cost}(T) \leq \left( \frac{n}{n-1} \right) \text{Cost}(M) \text{ ........................................... (2)}$$

Proof: Let $T_i$ be the tree obtained after removing edge $e_i$, then using lemma 1, Cost($T_i$) $\leq$ Cost(M)

There are $n$ such trees, so summing $n$ such trees,

$$\sum_{i=1}^{n} \text{Cost}(T_i) \leq \sum_{i=1}^{n} \text{Cost}(M)$$

$\Rightarrow$ $(n-1)\text{Cost}(T) \leq n\text{Cost}(M)$

$\Rightarrow \text{Cost}(T) \leq \left( \frac{n}{n-1} \right) \text{Cost}(M)$

Theorem 3:

Cost ($T$) $\leq \frac{5}{4}$ OPTIMAL

Proof: Now M can be converted to degree 4-spanning tree $T_4$ using algorithm 1

Therefore using inequality 2,

$$\text{Cost}(T) \leq \left( \frac{n}{n-1} \right) \text{Cost}(T_4)$$
Since \( \text{Cost} (T_4) \leq \frac{5}{4} \text{Cost(MST)} \) by theorem 1

Therefore \( \text{Cost}(T) \leq \frac{5}{4} \left( \frac{n}{n-1} \right) \text{Cost(MST)} \)

Using statement 2, \( \text{Cost(MST)} \leq \left( 1 - \frac{1}{n} \right) \text{OPTIMAL} \)

We get \( \text{Cost}(T) \leq \frac{5}{4} \left( 1 - \frac{1}{n} \right) \left( \frac{n}{n-1} \right) \text{OPTIMAL} \)

\( \Rightarrow \text{Cost}(T) \leq \frac{5}{4} \text{OPTIMAL} \)

**Conjecture:** It is conjectured that the approximation ratio for k-RNN algorithm is \( \frac{k^2 + 1}{k^2} \) for \( k > 1 \).

### 7. Conclusion and Future Work

If we choose any two nodes for initial tour, the bound for the ratio between tour by 2-RNN and optimal is \( \frac{5}{4} \). This can be baseline for finding the bound for the ratio if we choose \( k \) nodes as initial tour. Further research can be done to improve the time complexity of the algorithm from \( O(n^4) \).

### 8. References

1. Samir Khuller, Balaji Raghavachari and Neal E. Young. Low-Degree Spanning Trees of Small Weight. *SIAM Journal on Computing*, 25(2):355-368, 1996.

2. Daniel J. Rosenkrantz, Richard E. Stearns, and Philip M. Lewis. An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing*, 6(3):563-581, 1997.

3. Shayan Oveis Gharan, Amin Saberi and Mohit Singh. A Randomized Rounding Approach to the Traveling Salesman Problem. *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011.*
4. Nikolas Klug, Alok Chauhan, Vijayakumar V and Ramesh Ragala. k-RNN. Extending NN-heuristics for the TSP. Mobile Networks and Applications, pp. 1-4, January 2019.

5. Marcin Mucha. 13/9-approximation for Graphic TSP. Theory of Computing Systems, 55(4):November 2014.

6. András Sebő and Jens Vygen. Shorter Tours by Nicer Ears: 7/5-approximation for graphic TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraph. Combinatorica, 34 (5): January 2012.

7. Jeremy A. Karp and R. Ravi. A 9/7 -approximation algorithm for Graphic TSP in cubic bipartite graphs .Discrete Applied Mathematics, 209(C):164-216, 2016.

8. Piotr Berman and Marek Karpinski. 8/7-Approximation Algorithm for (1, 2)-TSP. SODA '06 Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp.641-648 , January 2006.

9. Wenhong Tian, Chaojie Huang and Xinyang Wang. A Near Optimal Approach for Symmetric Traveling Salesman Problem in Euclidean Space. DOI:10.5220/0006125202810287

10. A. Nagraj. Madhyasth Darshan Sahastitvavaad par Aadharit Samvaad (Part 1). Jeevan Vidhya Prakashan, 2011.

11. A. Nagraj. Madhyasth Darshan Sahastitvavaad par Aadharit Samvaad (Part 2). Jeevan Vidhya Prakashan, 2013.