Not All Insertion Methods Yield Constant Approximate Tours in the Euclidean Plane

Vineet Bafna [1]
Department of Computer Science
Penn State University
University Park, PA 16801

Bala Kalyanasundaram [2] Kirk Pruhs [3]
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260

March 24, 1993

[1]Supported in part by NSF under grant CCR-9114545
[2]Supported in part by NSF under grant CCR-9009318
[3]Supported in part by NSF under grant CCR-9209283

Abstract

An insertion heuristic for the traveling salesman problem adds cities iteratively to an existing tour by replacing one edge with a two-edge path through the new city in the cheapest possible way. Rosenkrantz, Stearns, and Lewis asked whether every order of inserting vertices gives a constant-factor approximation algorithm. We answer this question by showing that for some point sets, there is an order that yields tours with length $\Omega(\log n/\log\log n)$ times optimum, even if the underlying metric space is the Euclidean plane.

1 Introduction

Insertion methods are a class of algorithms, proposed by Rosenkrantz, Stearns and Lewis [11] for constructing a tour visiting a set V of points in a metric space M. The insertion method considers the points in V in some order, say $v_1, \ldots, v_n$, and constructs a sequence of partial tour, $T_1, \ldots, T_n$, where T_i is a tour of the points $v_1, \ldots, v_i$. T_1 is the point v_1, and T_2 consists of the two edges (v_1, v_2) and (v_2, v_1). The tour T_{i+1} is constructed by replacing some edge (x, y) in T_i by the two edges (x, v_{i+1}) and (v_{i+1}, y). The cost of this replacement is $d(x, v_{i+1}) + d(v_{i+1}, y) - d(x, y)$, where $d(\cdot, \cdot)$ is the distance function. The insertion method selects the replacement with minimum cost. The insertion method is greedy in the sense that T_{i+1} is the cheapest tour that visits the first i vertices in the same order as they are visited in T_i. Note that the insertion method does not specify the order in which the points in V are considered. Different orders may well produce different tours.

We say that a tour T is α *approximate* if the cost of T is at most α times the cost of the cheapest tour. The *performance guarantee* $PG_A(n)$ of an algorithm A is the supremum over all instances with n vertices of the cost of the tour produced by A divided by the cost of the optimal tour [7]. Rosenkrantz, Stearns, and Lewis [11] proved that some orderings of the input points yield a performance guarantee of at most two (independent of n). One example of such an ordering is nearest insertion. In this method, v_{i+1} is the point not in T_i that is closest to T_i, where the distance between a point x and a tour T is the minimum over $y \in T$ of $d(x, y)$. The proofs that these orderings yield a performance guarantee of at most two rest on finding a correspondence between the edges added by the insertion method and the edges in a minimum spanning tree. Note that the cost of the minimum spanning tree is at most the cost of the optimal tour, which in turn, has cost at most twice the cost of the minimum spanning tree [7].

In empirical trials conducted by Bentley [4] and Rosenkrantz *et al*, the orderings that yielded the best tours of points uniformly distributed in the unit square were farthest insertion and random insertion. In farthest insertion the point v_{i+1} is the point not in T_i that is farthest from T_i, and in random insertion v_{i+1} is chosen uniformly at random among the remaining points. However, the performance guarantees of these insertion methods are unknown [10].

Rosenkrantz *et al* proved that the performance guarantee of every the insertion method, given a worst-case point ordering, is at most $\lceil \log n \rceil + 1$. In contrast they stated that they did not know of any instance and corresponding ordering where the insertion method produced a tour with cost more than four times optimum. We show, in section 2, that there are instances and corresponding orderings of the input points on which the insertion method constructs tours that are $\Omega(\log n / \log \log n)$ approximate. Furthermore, in these instances the underlying metric space is the Euclidean plane. The construction is a modification of a construction used by Bentley and Saxe [5] to prove that the performance guarantee for the algorithm Nearest Neighbor was $\Omega(\log n / \log \log n)$.

Our original motivation for studying the performance guarantee of an arbitrary insertion method arose from our interest in the following online problem. We want to construct a telephone network, in the form of a spanning tree, connecting some set of cities. Furthermore, we want to minimize the amount of wire that is used. The well known optimal solution is that the connections should form a minimal spanning tree of the underlying distance graph [10]. However, over time it is likely that new cities will need to be added to the network. Since digging up existing phone lines would have significant cost, it would be infeasible to maintain the invariant that the cities are connected by the minimal spanning tree. Thus the problem is to maintain a spanning tree of small cost while only performing minimal modification each time a new point is added. It is not hard to see that the problem of maintaining a short tree online is equivalent to maintaining a short tour online in the sense that if there is a constant approximate algorithm for one then there is a constant approximate algorithm for the other.

In a general metric space, if one is not allowed to remove any part of the already existing tree, Imase and Waxman [8] and Chandra and Vishwanathan [6] proved that every algorithm must create spanning trees that are $\Omega(\log n)$ approximate for some instances. If the metric space is a plane, Alon and Azar [1] showed that every algorithm must create trees that are $\Omega(\log n/ \log\log n)$ approximate for some instances. It is not hard to show that all these results still hold if Steiner points are allowed.

One natural question to ask is how much one needs to modify the existing spanning tree (tour) to maintain a tree (tour) that is constant approximate. One natural way to measure the amount of modification is the number of edge deletions. For a general metric space, Imase and Waxman [8] showed that $O(n^{3/2})$ edge deletions are sufficient to maintain a constant approximate tree over n point expansions, for an amortized cost of $O(\sqrt{n})$ edge deletions per new point. Imase and Waxman conjectured that there is an algorithm that maintains a constant approximate tree with the worst-case number of edge deletions per new point being constant. The insertion method, with the points considered in chronological order, seemed like a natural candidate algorithm for maintaining a constant approximate tree (tour), while using only one edge deletion per new point. In this paper we thus rule out several variants of the insertion method algorithm as a possible candidate algorithms for solving this problem.

2 A Bad Insertion Method

We define three types of points, the *main points*, the *starter* points and the *correction points*. As the name implies, the main points are the ones most important to the construction. The main points are divided into $k+1$ rows (we assume k is even), with all points on a row being uniformly spaced on a horizontal line segment of length k^{6k}. There are $k^{6k-3l}+1$ points in row l, denoted R_l, for $0 \leq l \leq k$, and hence the distance between consecutive points on R_l is k^{3l}. The left endpoint of R_0 is at the origin, and the left endpoint of each row is on the y-axis. For $l > 0$, R_l lies a vertical distance of k^{3l-1} above R_{l-1}, and a vertical distance of k^{3l+2} below R_{l+1}. Hence the coordinates of the ith main point ($0 \leq i \leq k^{6k-3l}$) on R_l, denoted $p_{i,l}$, are $(ik^{3l}, \sum_{a=0}^{l-1} k^{3a+2})$.

The coordinates of the two starter points are $(0, \sum_{a=0}^{k} k^{3a+2})$, and $(k^{3k+3}, \sum_{a=0}^{k} k^{3a+2})$, ie. they would be the two leftmost points on R_{k+1} if we continued the pattern of main points. The starter points serve as a base case for our inductive construction. Along the line segment between the leftmost point in a row R_l, for odd l, and the point second from the left in R_{l+2} there are k^7 uniformly spaced correction points. We introduce these correction points so that the length of a diagonal edge between R_{l+2} and R_l does not exceed k^{3l}. It will become clear later that such small diagonal edges will not be replaced by the insertion method for points in rows below R_l. Figure 1 depicts the locations of the points in our construction.

Theorem 2.1 *The optimal tour for these points is $O(k^{6k})$.*

Proof: The theorem follows if we show that there is a spanning tree connecting the points with cost $O(k^{6k})$. The points on R_0 are connected from left to right, and each point in R_l, $l > 0$, is connected to the point in R_{l-1} directly below it. Thus the total cost of the edges in the spanning tree connecting the points in R_l to the points in R_{l-1} is the number of points in R_l times the distance between R_l and R_{l-1}, which is $(k^{6k-3l}+1)(k^{3l-1}) = \Theta(k^{6k-1})$. Hence the total cost of connecting the main points is $\Theta(k^{6k})$ for R_0, and $\Theta(k^{6k})$ for the aggregate cost of the other rows. The total cost for visiting the correction points and the starter points is clearly $o(k^{6k})$. $\diamond$

We now describe an ordering π of the points that causes the insertion method to create a tour of length $\Theta(kk^{6k})$. Points are generally revealed from top to bottom. The first two points revealed are the starter points. All the points on R_l are revealed before any of the points on R_{l-1} are revealed. The correction points between R_l and R_{l+2} are revealed after all the points on R_l are revealed, but before any points on R_{l-1} are revealed. If l is even then the points on R_l are revealed from left to right. If l is odd, then the first point in R_l that is revealed is the midpoint of R_l, $p_{m,l}$, where $m = k^{6k-3l}/2$. Then the points to the left of the midpoint of R_l are revealed from right to left, and finally, the points to the right of the midpoint are revealed left to right. Now the correction points between R_l and R_{l+2} are revealed in some arbitrary order.

Theorem 2.2 *The length of the tour generated by the insertion method for the ordering π is $\Theta(kk^{6k})$.*

To prove theorem 2.2, we consider the behavior of the insertion method row by row, and use the following inductive hypothesis.

Inductive Hypothesis: Assume that we are about to reveal the first point on R_{l-1}. Then the connections of the points in R_l, and above, are as follows:

1. The connections in R_l induce a line.

2. The connections in the even rows induce a line.

3. With the exception of the leftmost point, the connections in the odd rows induce a line.

4. There is a vertical edge going from the leftmost point in each even row to the leftmost point in the odd row above it.

5. There is a vertical edge going from the rightmost point in each odd row to the rightmost point in the even row above it.

6. The leftmost point in an each odd row R_i is connected via a series of *diagonal edges* through the correction points to the point second from the left in R_{i+2}.

7. If l is even, there is a *swing edge* between rightmost point in R_l and the point second from the left in R_{l+1}.

Figure 2 and figure 3 show the shape of the partial tour just before R_{l-1} is to be revealed. Figure 2 shows the case when the last row revealed was odd, and Figure 3 shows the case when last row revealed is even. The following six claims capture the behavior of the insertion method when the points in R_{l-1} and R_{l-2} are revealed according to π. Observe that R_{k+1}, which contains the starter points, is a odd row. Our induction will proceed two rows at a time. Hence we assume that l is odd in the following claims, and the partial tour is as pictured in figure 2.

Figure 2

Figure 3

Claim 1: When the leftmost point in R_{l-1}, $p_{0,l-1}$, is revealed, then the edge $(p_{0,l}, p_{1,l})$ is replaced.

Claim 2: When $p_{i,l-1}$ $(i > 0)$ is revealed, then the edge $(p_{i-1,l-1}, p_{1,l})$ is replaced.

Notice that this results in a partial tour, shown in figure 3, consistent with the inductive hypothesis.

Claim 3: When the midpoint in R_{l-2} is revealed, the swing edge, which connects $p_{1,l}$ to the rightmost point in R_{l-1}, is replaced.

Claim 4: When $p_{i,l-2}$ (for i to the left of the midpoint) is revealed, the edge connecting $p_{i+1,l-2}$ to R_l is replaced.

Claim 5: When $p_{i,l-2}$ (for i to the right of the midpoint) is revealed, the edge connecting $p_{i-1,l-2}$ to R_{l-1} is replaced.

Claim 6: When the k^7 correction points between R_{l-2} and R_l are revealed, the edge from $p_{0,l-2}$ to $p_{1,l}$ is subdivided into k^7 edges, each of length less than k^{3l-6}.

Once again notice that the partial tour is now consistent with the induction hypothesis. The following lemmas establish some geometrical properties that we use to prove claims 1 through 6.

Lemma 2.1 *Let (a, b) and (c, d) be edges and x be a point such that the line segments (a, x) and (b, x) intersect the line segment (c, d). Then the insertion method will choose (c, d) over (a, b).*

Proof: Let (a, x) intersect (c, d) at e, and (b, x) intersect (c, d) at f. Assume without loss of generality that e is closer to c than f. We need to prove that $d(c, x) + d(x, d) - d(c, d) \leq d(a, x) + d(x, b) - d(a, b)$. First note that $d(a, b) \leq d(a, e) + d(e, f) + d(f, b)$. Then substituting $d(a, e) = d(a, x) - d(e, x)$, and $d(b, f) = d(b, x) - d(f, x)$ we derive,

$$d(e, x) + d(x, f) - d(e, f) \leq d(a, x) + d(x, b) - d(a, b)$$

Note that,

$$d(c, x) + d(x, d) - d(c, d) = d(c, x) + d(x, d) - d(c, e) - d(e, f) - d(f, d)$$

Then by the triangle inequality this is less than $d(e, x) + d(x, f) - d(e, f)$. $\diamond$

Lemma 2.2 *Let (a, b) be an edge, with length at most k^{3l}, that has both endpoints on or above R_l. Then the cost of replacing (a, b) when a point x in R_j, for $j < l$, is revealed is at least $2k^{3l-3}$, twice cost of an edge in R_{l-1}.*

Proof: Let a' be the intersection of the line segment (a, x) with R_l, and b' be the intersection of the line segment (b, x) with R_l. By lemma 2.1 the cost of replacing (a, b) is at least the cost of replacing (a', b'). The cost of replacing (a', b') is minimized if x is on R_{l-1}, and if x lies on the perpendicular bisector of (a', b'). Then we need to show that $2\sqrt{[k^{3l}/2]^2 + [k^{3l-1}]^2} - k^{3l} \geq 2k^{3l-3}$. By squaring both sides one can easily see that this holds. $\diamond$

Lemma 2.3 *The cost of replacing a swing edge (a, b) between R_i and R_{i-1} when the midpoint m of R_{i-2} is revealed is less than 2.*

Proof: Drop a perpendicular from m to a point x on (a, b). Note that $d(x, m) \leq \sqrt{d(a, x)}$ and $d(x, m) \leq \sqrt{d(b, x)}$. Then notice that $d(a, m) \leq \sqrt{[d(a, x)]^2 + d(a, x)} < d(a, x) + 1$. Finally the claim follows since an analogous argument shows that $d(b, m) < d(b, x) + 1$. $\diamond$

We are now ready to prove the claims.

Proof of Claim 1: Among the horizontal edges in any row, the cheapest one to replace is the leftmost edge. Lemma 2.1 then guarantees that the best choice among the leftmost horizontal edges is $(p_{0,l}, p_{1,l})$. The vertical and diagonal edges above R_{l+1} are ruled out by lemma 2.2. Finally, by lemma 2.1 the cost of replacing a diagonal edge between R_l and R_{l+2} can be shown to be large by lemma 2.1. $\diamond$

Proof of Claim 2: The cost of insertion as per the claim is strictly less than $2k^{3l-3}$, which by lemma 2.2 is less than the cost of replacing an edge with both endpoints on or above R_{l-1}. The cost of replacing another horizontal edge in R_{l-1} is at least $2k^{3l-3}$. $\diamond$

Proof of Claim 3: By lemma 2.3 the cost of replacing the swing edge is strictly less than 2. By lemma 2.2 the cost of replacing any other edge would be greater than 2. $\diamond$

The proofs of claim 4 and claim 5 are almost identical to the proof of claim 2.

Proof of Claim 6: The cost of the replacement is 0, while any other replacement would have positive cost. $\diamond$

Proof of Theorem 2.2: In the resultant tour, almost all of the points in R_l are connected to both their left and right neighbors. Hence, the length of the subtour visiting R_l is $\Theta(k^{6k})$. The claim then follows since there are k rows. $\diamond$

Theorem 2.3 *Some insertion methods have a performance guarantee of $\Omega(\log n / \log \log n)$.*

Proof: Since $n = \Theta(k^{6k})$, $k = \Theta(\log n / \log \log n)$. $\diamond$

3 Conclusion

There are several open questions remaining to be answered, including: Can one prove an $O(\log n / \log \log n)$ bound on the performance guarantee of the insertion method in the

Euclidean plane? Can one strengthen the bound to $\Omega(\log n)$ for a general metric space? Can a constant approximate spanning tree (or tour) be maintained in the Euclidean plane with only one edge deletion per new point? How many edge deletions are needed in a general metric space?

Acknowledgments: We would like to thank Jim Saxe for informing us of the construction in [5]. We have recently learned that Azar [2] has independently discovered the result presented in the paper as well as an $\Omega(\log n / \log\log n)$ on the performance guarantee of random insert.

References

[1] N. Alon and Y. Azar, "On-line Steiner Trees in the Euclidean Plane," to appear in *Proceedings of 8th Symposium on Computational Geometry*, 1992.

[2] Y. Azar, "Lower Bounds for Insertion Methods for TSP", Manuscript.

[3] R. Baeza-Yates, J. Culberson, and G. Rawlins, "Searching with Uncertainty", to appear in *Information and Computation.*

[4] J. Bentley, "Experiments on Traveling Salesman Heuristics", *Proceedings of 1st ACM/SIAM Conference on Discrete Algorithms*, pp. 91–99, 1990.

[5] J. Bentley and J. Saxe, "An Analysis of Two Heuristics for the Euclidean Traveling Salesman Problem", *Proceedings of the Eighteenth Annual Allerton Conference on Communication, Control, and Computing*, pp. 41–49, 1980.

[6] B. Chandra, and S. Vishwanathan, "Constructing Reliable Communication Networks of Small Weight Online", Manuscript.

[7] M. Garey and D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-completeness", W.H. Freeman, New York, 1979.

[8] M. Imase, and B. Waxman, "Dynamic Steiner Tree Problem", *SIAM Journal of Discrete Mathematics*, **4**, pp. 369–384, 1991.

[9] B. Kalyanasundaram, and K. Pruhs, "Online Construction of Short Traveling Salesman Tours", Technical Report 92-2, Computer Science Department, University of Pittsburgh, 1992.

[10] E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Smoys, *The Traveling Salesman Problem*, Wiley, New York, 1985.

[11] D. Rosenkrantz, R. Stearns, and P. Lewis, "An Analysis of Several Heuristics for the Traveling Salesman Problem", *SIAM Journal of Computing*, **6**, pp. 563–581, 1977.

[12] G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set", *Pattern Recognition*, **12**, pp. 261–268, 1980.