An Approximation Algorithm for the Curvature-Constrained Traveling Salesman Problem*

Jerome Le Ny
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
jleny@mit.edu

Eric Feron
School of Aerospace Engineering
Georgia Tech
Atlanta, GA 30332
ef57@mail.gatech.edu

Abstract

We present an approximation algorithm for the Traveling Salesman Problem when the vehicle is constrained to move forward along paths with bounded curvature (Dubins' vehicle). A deterministic algorithm returns in time $O(n^3)$ a tour within $\left(\left(1 + \max\left\{\frac{8\pi\rho}{D_{min}}, \frac{14}{3}\right\}\right)\log n\right)$ of the optimum tour, where n is the number of points to visit, ρ is the minimum turning radius and D_{min} is the minimum Euclidean distance between any two points. A randomized version returns a tour with an expected approximation ratio of $\left(\left(1 + \frac{13.58\rho}{D_{min}}\right)\log n\right)$. This very simple algorithm reduces the Traveling Salesman Problem for Dubins' vehicle to an Asymmetric Traveling Salesman Problem on a directed graph.

1 Introduction

In the Traveling Salesman Problem (TSP), we are given n nodes and for each pair (i, j) of distinct nodes, a distance d_{ij}. We want to find a closed path that visits each node exactly once and incurs the least cost, which is the sum of the distances along the path. The distances need not be symmetric; i.e., we can have two nodes i, j with $d_{ij} \neq d_{ji}$, in which case the problem is called the Asymmetric Traveling Salesman Problem (ATSP). In the metric TSP, the distances satisfy the triangle inequality. A subcase of the metric TSP is the (planar) Euclidean TSP (ETSP), in which the nodes lie in $\mathbb{R}^2$ and the distance is the usual Euclidean distance.

Computing good TSP tours efficiently is of interest in the area of aerial surveillance. Because we are increasingly interested in developing autonomous vehicles, the question of how these vehicles should behave usually leads to optimizing a given objective function, for example minimizing the distance traveled when the task is to explore a set of locations. An important difficulty arises, however, when the problem involves planes, underwater vehicles, cars and other vehicles with significant dynamics: the paths obtained from algorithms solving the Euclidean TSP are infeasible. Kinodynamic planning refers to the path-planning problem when the kinematic constraints of the vehicle are taken into account. The methods developed in this field aim at finding a trajectory from an initial

[*] This work was supported by Air Force - DARPA - MURI award 009628-001-03-132, The University of California award 025-G-CB222, and Navy ONR award N00014-03-1-0171.

position and configuration to a final position and configuration, usually while avoiding potential obstacles. In this paper, we study a different problem. We want to optimize trajectories visiting a specified set of points, but the configuration of the vehicle at these points is free as long as the kinematic constraints are satisfied.

We focus on algorithms computing tours for Dubins' vehicle, a problem which was recently considered in [10, 9]. Dubins' model [3] is a simple but efficient way to handle the dynamic characteristics of fixed-wing aircrafts. It gives a complete characterization of the optimal paths between two configurations for a vehicle with limited turning radius moving in a plane at constant speed.

The complexity issue of our algorithms is an important part of the analysis, since the path planner is usually only one component of a general scheduling system, and therefore its task should not become prohibitively time-consuming. The TSP is NP-hard, even in the Euclidean case [8], and therefore we are interested in efficient *approximation algorithms*. More precisely, an α-approximation algorithm ($\alpha \geq 1$) for a minimization problem with optimum OPT is an algorithm that produces in polynomial time a feasible solution whose value Z is within a factor α of the optimum, i.e., such that

$$OPT \leq Z \leq \alpha\, OPT.$$

It is known that in the most general case, there can be no approximation algorithm for the TSP unless P=NP. But if the distances satisfy the triangle inequality, Christofides' algorithm [2] gives a 3/2-approximation for the symmetric TSP, and there is a $(\log n)$-approximation for the ATSP [4], where n is the number of points, and log denotes the logarithm of base 2. No constant factor approximation for the ATSP is currently known. For the ETSP, Arora [1] gives a polynomial-time approximation scheme that can approximate the optimal tour within $(1+\epsilon)$ for any $\epsilon > 0$. The analysis of our algorithm will show that we have a $O(\frac{\rho \log n}{D_{min}})$ [1] approximation for the TSP for Dubins' vehicle (DTSP), where D_{min} is the minimum Euclidean distance between any two points in the set and ρ is the minimum turning radius of the vehicle. It should be noted that compared to the situation considered in [9], our algorithm provides a worst-case approximation guarantee for any configuration of the points to visit.

This paper is organized as follows: Section 2 recalls some facts about point-to-point Dubins' paths. Then Section 3 presents a simple approximation algorithm for the DTSP and compares it to the approach taken in [10]. Section 4 gives a randomized version of the algorithm that achieves a slightly better approximation factor. Section 5 presents computational results, and finally Section 6 comments on the tightness of our analysis and suggests possible improvements.

2 Point-to-Point Dubins' Paths

We consider a point vehicle moving at *unit speed* (without loss of generality) in the plane, with a constraint on its maximal turning rate. More formally, given a continuously differentiable path $P : I \rightarrow \mathbb{R}^2$ parametrized by arc length $s \in I$, the average curvature of P in the interval $[s_1, s_2] \subseteq I$ is defined by $\|P'(s_1) - P'(s_2)\|_2 / |s_1 - s_2|$. We require that the average curvature of the vehicle's path be at most $1/\rho$ in every interval. Denote the *configuration* of the vehicle by (X, θ), where X is the location of the vehicle in the plane

[1] We say $f(n) = O(g(n))$ if there exists $c > 0$ such that $f(n) \leq cg(n)$ for all n, and $f(n) = \Omega(g(n))$ if there exists $c > 0$ such that $f(n) \geq cg(n)$ for all n.

and $\theta \in (-\pi, \pi]$ is its heading, i.e., the angle that the velocity vector makes with the x-axis. The goal is to design an efficient approximation algorithm that, given a set of point locations in a bounded square (in an obstacle-free environment), returns a permutation of the points specifying the order of the visits, as well as headings for the vehicle at each point.

Dubins [3] characterized curvature constrained shortest paths between an initial and a final configuration. Let P be a feasible path. We call a nonempty subpath of P a C-segment or an S-segment if it is a circular arc of radius ρ or a straight line segment, respectively. We paraphrase the following result from Dubins:

Theorem 1 ([3]). *An optimal path between any two configurations is of type CCC or CSC, or a subpath of a path of either of these two types. Moreover, to be optimal, a CCC path must have its middle arc of length greater than $\pi\rho$.*

In the following, we will refer to these minimal-length paths as Dubins' paths. When a subpath is a C-segment, it can be a left or a right hand turn: denote these two types of C-segments by L and R respectively. Then we see from Theorem 1 that to find the minimum length path between an initial and a final configuration, it is enough to find the minimum length path among six paths, namely among $\{LSL, RSR, RSL, LSR, RLR, LRL\}$. Each of these paths can be explicitly computed (see for instance [11]) and therefore finding the optimum path and length between any two configurations can be done in constant time.

Note that, in our case, the configuration of the vehicle at each point is not completely specified. Only the position of the points is known, the headings of the vehicle at these points must be found. Since we do not expect to obtain the exact optimum headings, we will need a lemma based on a result due to Jacobs and Canny [6] on the difference in length between two paths when only the initial and terminal headings are different. In general, if we consider a path of Dubins' length p_{ij} between two configurations (X_i, θ_i) and (X_j, θ_j) and make an error up to $\delta \in (-\pi, \pi]$ on the initial and final headings, we can derive a multiplicative bound on the perturbed path $\hat{p}_{ij}$ as follows.

Lemma 2. *Let d_{ij} be the Euclidean distance between X_i and X_j. We have:*

$$\hat{p}_{ij} \leq \left(1 + 2\rho \max\left\{\frac{3|\delta| + \pi|\sin\frac{\delta}{2}|}{d_{ij}}, \frac{|\delta| + 4\arccos\left(1 - |\sin\frac{\delta}{2}|/2\right)}{\pi\rho}\right\}\right) p_{ij}. \qquad (1)$$

Proof. See [6] section 4.8 for a proof of this lemma [2]. Let us mention that the two terms in the max on the right-hand side correspond to the cases where the initial Dubins' path is a CSC path with opposite initial and final turning directions and a CCC path respectively. Perturbations of a CSC path with identical initial and final turning directions are dominated by the first term. Moreover, for the second term, the fact that a CCC optimal path must have minimum length $\pi\rho$ was used. □

3 An Approximation Algorithm for the DTSP

Savla et al. [10] give an algorithm for computing TSP tours for Dubins' vehicle, called the "alternating algorithm". It works as follows: given a set of n points, the optimal Euclidean TSP tour is computed, and the order of visits for the ETSP is used for the

[2] It appears that eq. (4.38) and (4.48) in [6] contain a typographical error, namely a ρ is missing.

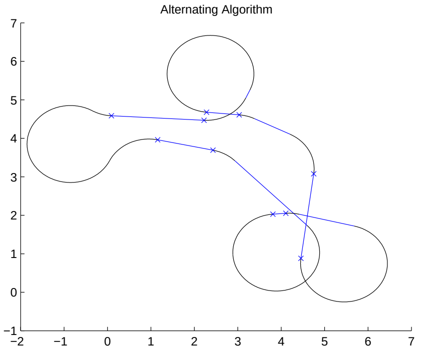


Figure 1: DTSP tour obtained from the alternating algorithm for a set of 10 points.

curvature-constrained tour. It is then necessary to obtain a feasible path through these ordered points. Following the edges of the tour, all odd-numbered edges are retained (i.e. the subpath is a straight line) as well as the corresponding headings, and the even-numbered edges are replaced with Dubins' paths (see Fig. 1).

The alternating algorithm is not an approximation algorithm for the DTSP in the sense of the definition given in the introduction. It could easily be made a polynomial time algorithm by using Arora's $(1+\epsilon)$-approximation of the ETSP instead of an exact algorithm. But a more critical issue is that there could be little relationship between the Euclidean and Dubins' metrics, especially when the Euclidean distances are small with respect to the turning radius, which is the case we are mainly interested in for the DTSP. An algorithm for the Euclidean problem will tend to schedule very close points in a successive order, which can imply long maneuvers for the aircraft. We will come back to this point in section 5. In the following we suggest a simple approximation algorithm which does not rely on the Euclidean solution.

The major difficulty is to determine the headings. We argue however that we have much freedom in selecting them, provided we can be satisfied with a relatively weak performance bound. A first deterministic algorithm can be described as follows:

1. Fix the headings at all points to be 0.
2. Compute the $n(n-1)$ Dubins' distances between all pairs of points.
3. Construct a complete graph with one node for each point and edge weights given by the Dubins' distances.
4. We obtain a *directed* graph where the edges satisfy the triangle inequality. Compute the solution of the asymmetric traveling salesman problem on this graph, using the $\log n$-approximation algorithm of Frieze et al. [4].

The complexity of the three first steps is $O(n^2)$. The algorithm for solving the ATSP runs in $O(n^3)$, so overall the running time of our algorithm is $O(n^3)$.

To analyze the performance guarantee, we use the bound (1). At each point, we can make an error up to $|\delta| = \pi$ with respect to the optimal heading. Let p_{ij} be the distances if the optimal headings were selected, and $\hat{p}_{ij}$ the distances after we fix the headings. We obtain:

$$\hat{p}_{ij} \leq \left(1 + \max\left\{\frac{8\pi\rho}{D_{min}}, \frac{14}{3}\right\}\right) p_{ij} = C\ p_{ij}, \tag{2}$$

where $D_{min} = \min_{i \neq j}\{d_{ij}\}$, and C is defined by the equation.

Note that theoretically there is no reason for D_{min} to be bounded from below, and therefore the upper bound (2) can be arbitrarily bad. In practice however, in the context of aerial surveillance, we can restrict our study to the situation where the points have a minimum distance between them, for example equal to the coverage radius of the sensors of the aircraft, which allow very close points to be observed at the same time. This can be justified by the following greedy procedure. Suppose we can observe at each instant the area inside a disk of radius D_{min} around the aircraft. For a given set of points, start by picking a point arbitrarily, and discard all the points which are within a distance D_{min} of this first point. Next, pick a second point arbitrarily among the remaining points, and continue similarly, until all the points have been considered. The points selected have a distance greater than D_{min} between them, and finding a trajectory visiting these points is enough to cover the complete initial set. Therefore it is enough to run our algorithm for a set of points with minimum pairwise distance bounded from below by D_{min}, and this implies that for practical purposes D_{min} can be considered as a constant independent of n.

Call OPT the optimal value of the DTSP and σ^* the corresponding optimal permutation specifying the order of visits. We have $OPT = \sum_{i=1}^{n-1} p_{\sigma^*(i)\sigma^*(i+1)} + p_{\sigma^*(n)\sigma^*(1)} := L(\{p_{ij}\}, \sigma^*)$, where the definition of the functional L should be clear from the equation. Considering the permutation σ^* for the graph problem (where the edge weights are the perturbed distances $\{\hat{p}_{ij}\}$) and $\hat{\sigma}^*$ the optimal permutation for the graph problem, we have

$$L(\{\hat{p}_{ij}\}, \hat{\sigma}^*) \leq L(\{\hat{p}_{ij}\}, \sigma^*) \leq C\, L(\{p_{ij}\}, \sigma^*)$$

Now on the graph with weights $\hat{p}_{ij}$, we can solve the traveling salesman problem in polynomial-time with an approximation ratio of $\log n$, so calling $\hat{\sigma}$ the corresponding solution we have:

$$L(\{\hat{p}_{ij}\}, \hat{\sigma}) \leq \log n\, L(\{\hat{p}_{ij}\}, \hat{\sigma}^*) \leq C\, \log n\, L(\{p_{ij}\}, \sigma^*) = (C\, \log n)\, OPT.$$

Therefore, we obtain with the specific assignment of headings (or any assignment in fact) an approximation guaranteed to be within a factor $\left(1 + \max\left\{\frac{8\pi\rho}{D_{min}}, \frac{14}{3}\right\}\right) \log n$ of the optimum.

We can make several remarks about this algorithm:

- If we *randomize* the heading assignments, we obtain a better guarantee in expectation, more precisely we can lower the constant in front of the logarithm, by avoiding the worst-case analysis assuming $\delta = \pi$. This is described in section 4.

- A deterministic generalization is to add more discretization levels to decrease the value of the error δ. At each point, we consider a set of K possible headings, and we want to select one of them optimally. To each point in the original problem, we associate a cluster of K nodes corresponding to the K different headings. We then have to solve a problem called the "Generalized Asymmetric Traveling Salesman Problem", i.e., find a tour through n sets of K vertices, visiting one node in each cluster (that is, selecting one possible heading at each point). The Generalized Traveling Salesman Problem can be reduced to the Traveling Salesman Problem [7], although in practice this might become too complex.

- As we have pointed out, the presence of D_{min} could make the approximation ratio arbitrarily bad in theory. It seems difficult to avoid this term in the performance bound, as long as the analysis is carried independently on each point-to-point path as we did above. Consider the path from $(0, 0, \theta_i = 0)$ to $(\epsilon, 0, \theta_j = 0)$, with $\epsilon > 0$: this is just a straight line. But if we make an error δ on θ_i, as $\epsilon \to 0$ we obtain a Dubins' path of positive length since the initial and final configurations are different and the aircraft has to maneuver to change its heading. Therefore the ratio of the Dubins' distance of the perturbed path to that of the original path becomes infinite, however small the error on the initial angle was.

- The $\log n$ factor appears to be inherently linked to our reduction to a directed graph. Note however that discretization and reduction of a kinodynamic planning problem to a directed graph formulation is standard. The asymmetry is due exactly to the dynamic constraints (for example, making a U-turn is costly). Working with a directed graph is not an issue as long as the graph problem is simple like computing a shortest path, but the situation is worse for more complicated problems such as the TSP, since much less is known about digraphs. Reducing the $\log n$ factor therefore appears to be quite challenging unless a better approximation algorithm is devised for the ATSP or a method departing from the traditional discretization approaches is used (or a hardness result is obtained).

4 Randomized Version

We can modify the algorithm described in part 3 to obtain an interesting randomized algorithm. Instead of assigning all headings to be 0, we choose the headings randomly and independently in $(-\pi, \pi]$ for each point.

Consider an optimal tour, and two successive points X_i, X_j in this tour. The Dubins' path between these two points has length p_{ij}, and the optimal headings are θ_i and θ_j. Following [6] in the derivation of the bound (1), we know that if we make an error of $\delta \in (-\pi, \pi]$ on θ_i, the difference in path length is bounded by:

$$\Delta p \leq \rho \max \left\{ 3|\delta| + \pi |\sin(\frac{\delta}{2})|,\ |\delta| + 4 \arccos \left(1 - |\sin \frac{\delta}{2}|/2 \right) \right\}. \tag{3}$$

This leads to the inequality (1) by taking into account the error on θ_j as well. Now (3) is derived for a change from θ_i to $\theta_i + \delta$ in the initial heading. Of course, we do not know the optimal θ_i so the natural idea is to choose $(\theta_i + \delta)$ uniformly in $(-\pi, \pi]$, in which case the error δ is distributed uniformly in $(-\pi, \pi]$ as well. This implies that Δp becomes a random variable whose expectation is bounded by:

$$E[\Delta p] \leq \rho \int_{-\pi}^{\pi} \max \left\{ 3|\delta| + \pi |\sin \frac{\delta}{2}|,\ |\delta| + 4 \arccos \left(1 - |\sin \frac{\delta}{2}|/2 \right) \right\} \frac{d\delta}{2\pi} \leq 6.79\rho.$$

Replacing the corresponding expression in (1), we obtain as a final upper bound:

$$E[\hat{p}_{ij}] \leq \left(1 + \frac{13.58\rho}{d_{ij}} \right) p_{ij}. \tag{4}$$

It is also possible to refine the bound using the fact that a CCC path has length at least $\pi\rho$ as in section 3, but for our purpose D_{min} and ρ will be of the same order and (4) is then enough.

We can now reproduce the analysis of part 3, using only the linearity of expectation. Thus, we see that we have a randomized algorithm which, given a set of n points, returns a tour whose expected length is within $\left(1 + \frac{13.58\rho}{D_{min}}\right) \log n$ of the optimum.

5 Computational Experiments

In this section we describe some computational experiments for the randomized algorithm described in part 4. The algorithm was implemented in MATLAB. For the computation of the ATSP tours, we did not implement Frieze's algorithm, rather we used LKH, Helsgaun's implementation of the Lin-Kernighan heuristic, available as a C-code [5]. Our main program calls this routine when asked to solve an ATSP. For the sake of comparison, we also implemented the alternating algorithm, using LKH for computing the Euclidean TSP tours as well. LKH has excellent performance on real-world problems, and solves the small instances that we considered exactly without difficulty. Therefore our implementation of the algorithm returns a tour of expected length less that $\left(1 + \frac{13.58\rho}{D_{min}}\right)$ times the optimum length for the problems considered, i.e. the $\log n$ factor did not appear. In practice, for a given instance of the problem, we generated 10 tours using 10 different sets of random headings, and returned the best tour obtained. The difficulty in verifying the performance of the algorithm is that it is hard to evaluate the true optimum.

We ran simulations for different sizes of point sets; the sets consisted of points generated randomly and uniformly inside a square of side length 5. In all simulations, the turning radius of the vehicle was fixed to 1. In order to compare the performance of the alternating algorithm and our algorithm, for n fixed, we generated 10 different sets and compared the average lengths returned by the two algorithms on these samples. The results are shown on Fig. 2. As expected, when the density of the points increases, the performance of the alternating algorithm decreases with respect to the performance of our randomized algorithm. For points far apart however, the Dubins problem becomes similar to the Euclidean problem, and therefore we can expect the alternating algorithm to perform almost optimally. In practice, to obtain an algorithm which can handle various point configurations, we can compute tours with both algorithms and choose the best one.

For sufficiently dense sets of points, it becomes clear that the ordering of the Euclidean tour is not optimal in the case of the DTSP. In Fig. 1, we can see that modifying the ordering of the group of 3 points in the top loop should provide a better tour. In general, as previously mentioned, we expect that the Euclidean tour will schedule close points successively, which may result in long maneuvers. The alternating algorithm tends to create numerous loops that become problematic with dense sets of points (see Fig. 3).

We conclude this section by commenting on the tightness of the performance bounds obtained in parts 3 and 4. For random sets as considered above, the worst case bound, based on the point-to-point worst case performance, is far from the actual performance, typically by an order of magnitude or two. This bound is very conservative, for example the term $1/D_{min}$ is present even if the two closest points are not scheduled in successive order. This calls for a more global analysis of the performance. Note also that in our experiments, we simply chose the points uniformly at random, without imposing any restriction on D_{min}. The fact that the square can be covered by $O(n)$ boxes of side length $n^{-\frac{1}{2}}$ tells by a pigeonhole argument that $D_{min} = O(n^{-\frac{1}{2}})$. This means that our approximation ratio is $\Omega(\sqrt{n} \log n)$ for target points distributed uniformly in the square. On the

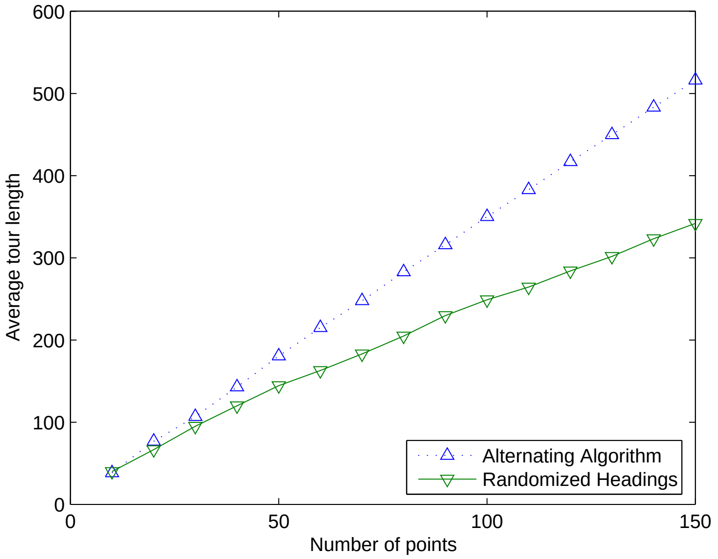
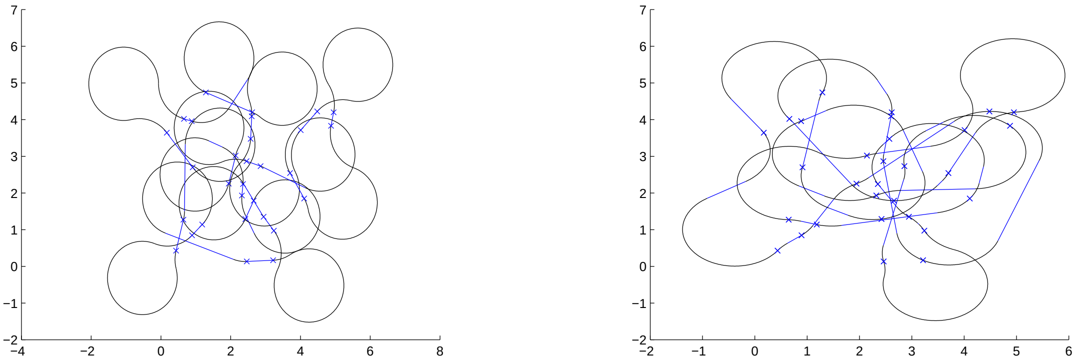


Figure 2: Performance of the alternating algorithm and the randomized headings algorithm for sets of points of increasing density in a square of side length 5.



(a) Alternating Algorithm

(b) Randomized Headings

Figure 3: Paths generated by the alternating algorithm and the randomized headings algorithm for a set of 30 points in a square of side length 5.

other hand, Savla et al. prove in [9] that the expected length of the optimum Dubins' tour in this case is $\Omega(n^{2/3})$. This implies that our upper bound on the length returned by the randomized headings algorithm is $\Omega(n^{7/6} \log n)$, while experimental results suggest that we should expect a sublinear bound (see Fig. 2).

6 Conclusions

We have presented a simple algorithm to compute efficiently tours within sets of points for the Dubins' vehicle, which models as a first approximation the dynamics of a fixed-wing aircraft. The Dubins' paths cannot be followed exactly by a real vehicle because instantaneous changes of acceleration are necessary. However, our algorithm is useful for obtaining a first approximation of the trajectory for instance sizes that could not be handled by more precise models. Through simulation, we have demonstrated that the performance of the randomized version of the algorithm is in general better than for the alternating algorithm, which is based on computing Euclidean tours. This result is supported by the idea that, for dense sets of points, the Dubins' metric has a behavior departing significantly from the Euclidean metric. We have provided a performance-bound guarantee for the algorithm, however typically the performance obtained is much better than this worst case bound. Our analysis relied on a point-to-point result, and we expect that a significantly better analysis cannot be obtained except by adopting a more global approach. For the randomized algorithm, it would also be interesting to obtain a result with high probability, to improve on our expected bound.

References

[1] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[2] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report, CSIA, Carnegie-Mellon Univ., 1976.

[3] L.E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[4] A. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12:23–39, 1982.

[5] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. Website. http://www.akira.ruc.dk/~keld/research/LKH/.

[6] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In Z. Li and J. Canny, editors, *Nonholonomic Motion Planning*, pages 271–342. Kluwer Academic, 1992.

[7] C.E. Noon and J.C. Bean. An efficient transformation of the generalized traveling salesman problem. *Information Systems and Operational Research*, 31(1), 1993.

[8] C. H. Papadimitriou. Euclidean TSP is NP-complete. *Theoretical Computer Science*, 4:237–244, 1977.

[9] K. Savla, F. Bullo, and E. Frazzoli. On the stochastic traveling salesperson problem for Dubins' vehicle. In *IEEE Conference on Decision and Control*, Seville, Spain, December 2005. Submitted.

[10] K. Savla, E. Frazzoli, and F. Bullo. On the point-to-point and traveling salesperson problems for Dubins' vehicle. In *Americal Control Conference*, Portland, OR, June 2005.

[11] A.M. Shkel and V.J. Lumelsky. Classification of the Dubins' set. *Robotics and Autonomous Systems*, 34:179–202, 2001.