Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Task and Motion Planning Algorithm for the Dubins Travelling Salesperson Problem ⋆

Pantelis Isaiah, Tal Shima

The Technion—Israel Institute of Technology, Haifa, Israel, 32000
(e-mail: pantelis@technion.ac.il, tal.shima@technion.ac.il)

Abstract: A new motion planning algorithm for the so-called Dubins Travelling Salesperson Problem is derived, and compared via simulations with a number of existing algorithms from the literature. In its general form, the new algorithm is dubbed "k-step look-ahead algorithm" and stems naturally from the formulation of the Dubins Travelling Salesperson Problem as a minimum-time control problem. When the minimum turning radius of the Dubins vehicle is comparable to the average intercity distance, the simulations yield a comparison favourable to the new algorithm. The examples in the paper are confined to small instances of the Dubins Travelling Salesperson Problem, however the main idea behind the k-step look-ahead algorithm can be combined with different optimisation methods, if larger instances of the DTSP are to be considered.

1. INTRODUCTION

The Dubins Travelling Salesperson Problem (DTSP) is a useful abstraction for the study of problems related to motion planning of uninhabited vehicles [Yang and Kapila, 2002; McGee and Hedrick, 2006; Rathinam et al., 2007; Savla et al., 2008; Tang and Özgüner, 2005; Edison and Shima, 2011; Cons et al., 2013]. As in the case of the classic Euclidean Travelling Salesperson Problem (ETSP) in $\mathbb{R}^2$ [Papadimitriou and Steiglitz, 1998; Cormen et al., 2001], the sought after solution is a tour of minimum length that passes through every city (target) once, however, in the case of the DTSP, the tour is required to be a C^1 curve of bounded curvature. The additional requirement on the regularity and the curvature of the tour has a fundamental implication on the very nature of the problem. Specifically, the ETSP belongs to the realm of combinatorial optimisation, whereas the DTSP does not. A precise formulation of the DTSP is given below, however the crux of the matter is that, in the case of the DTSP, even if the order of the targets is given and fixed, the length of the tour depends on the heading of the Dubins vehicle when it passes through each target (in other words, the slope of the tour at each target). Therefore, the solution space for the DTSP has the cardinality of the continuum.[1] A possible remedy is to discretise the interval $[0, 2\pi)$ for each target, however this approach has at least two shortcomings that are intimately related. First, the minimum-time function for the Dubins dynamics is discontinuous and, hence, a fine partition of the aforementioned interval is necessary if an optimal trajectory is to be approximated; second, an algorithm that checks a large number of orientations for each target is bound to be impractical even for small instances (i.e., number of targets) of the DTSP.

Among the contributions of the present work is the systematic reduction of the DTSP to a problem of combinatorial optimisation by means of a receding horizon principle that is suitably adjusted to the Dubins dynamics and leads to feasible tours (for the DTSP) that can be much shorter than those obtained by other methods. For example, our simulations demonstrate substantial improvement over algorithms that rely on a solution to the ETSP to find a feasible tour for the DTSP [Savla et al., 2008; Ma and Castañón, 2006]. When the horizon equals the number of targets, our algorithm—the k-step look-ahead algorithm (k-step LAA)—returns a solution to the DTSP which is globally optimal. Although the computation of globally optimal solutions is impractical even for small instances of the DTSP, it has the potential of shedding some light on the structure of solutions to the DTSP which, until now, has been tackled only with heuristics. Both the ETSP and the DTSP are NP-hard problems [Papadimitriou, 1977; Le Ny et al., 2012], however an intuitive understanding of optimal solutions to the DTSP will lead to better motion planning algorithms and seems to be currently lacking.

Ultimately, every algorithm for DTSP relies on a mechanism for assigning headings to the targets and, in this sense, on a discretisation scheme. Once such an assignment has been made, several existing algorithms—specifically, algorithms for the Asymmetric Travelling Salesperson Problem (ATSP)—can be directly applied to find a feasible tour for the DTSP. For the purpose of illustration and comparison, the k-step LAA was combined with a best-first search method to generate the simulations presented in this paper. Partly because of this specific choice of implementation, the applicability of the overall algorithm is limited to small instances of the DTSP and leaves open

⋆ This work was supported by the U.S. Air Force Office of Scientific Research, Air Force Material Command, under grant FA8655-12-1-2116.

[1] Although the solution space consists of closed, planar *curves*, to each n-tuple $(\theta_1, \ldots, \theta_n) \in \mathbb{S}^n$ of headings, where n is the number of targets, corresponds a finite number of tours of the same minimum length.

for further investigation the fusion of the k-step LAA with different optimisation algorithms or approximation algorithms for the ATSP. That said, there are application areas where a small number of targets does not represent an unrealistic scenario. Lastly, some computational aspects of the process of discretising the DTSP are analysed in Le Ny et al. [2012].

2. PROBLEM STATEMENT

Consider the control-affine system

$$\gamma'(t) = f(\gamma(t)) + u(t)g(\gamma(t)) \qquad (\Sigma)$$

on $M = \mathbb{R}^2 \times \mathbb{S}^1$, where f, g are the real analytic vector fields (i.e., elements of $\Gamma^\omega TM$) with coordinate representations $X(x, y, \theta) = (\cos\theta, \sin\theta, 0)$ and $Y(x, y, \theta) = (0, 0, 1)$, in the chart on TM induced by the chart $(U, \phi) = (\mathbb{R}^2 \times \mathbb{S}^1 \backslash \{(-1, 0)\}, (x, y, v, w) \mapsto (x, y, \theta = \operatorname{atan}(w/v)))$ on M. [2] The admissible controls are the locally integrable maps $u : \mathbb{R} \supset I \ni t \mapsto u(t) \in [-1/\rho, 1/\rho] \subset \mathbb{R}$, where ρ is a fixed, positive, real number and, given a control u, γ denotes the corresponding locally absolutely continuous trajectory of Σ. The projection on $\mathbb{R}^2$ of a trajectory γ will be called the **path** that corresponds to γ. An **optimal path** is the projection of an optimal trajectory. A trajectory $\gamma : \mathbb{R} \supset [a, b] \to M$ of (Σ) is said to be **closed** if $\gamma(a) = \gamma(b)$ and the path that corresponds to a closed trajectory will be called a **tour**. If I is an interval in $\mathbb{R}$, we denote by $C^0(I; M)$ the continuous curves in M defined on I and by $C^\infty_{\mathrm{pw}}(I; M)$ the curves γ for which there exists a finite set $S_\gamma \subset I$ such that γ is C^∞ on $I \setminus S_\gamma$.

The control system (Σ) can be viewed as the kinematic model of a point that moves with constant, unit speed, along a planar curve whose curvature is bounded by $1/\rho$. We shall abide by the common convention of referring to (Σ) as the "Dubins vehicle" and to the following minimum-time control problem as the Dubins Travelling Salesperson Problem (DTSP).

DTSP: *Let n be a positive integer. Given a point $p \in M$ and n submanifolds (targets) of the form $N_i = \{(x_i, y_i)\} \times \mathbb{S}^1 \subset M$, where $(x_i, y_i) \in \mathbb{R}^2$ and $i \in \{1, \ldots, n\}$, minimise the time $T > 0$ over the set $\Gamma^{\mathrm{cl}}_\Sigma(p, N_1, \ldots, N_n)$ of closed trajectories $\gamma \in C^0 \cap C^\infty_{\mathrm{pw}}([0, T]; M)$ of (Σ) that satisfy $\gamma(0) = \gamma(T) = p$ and $\mathrm{Im}\gamma \cap N_i \neq \emptyset$, for every $i \in \{1, \ldots, n\}$.*

To avoid trivial complications with the statements and the notation that follow, we always assume that the targets are distinct from each other and that the initial condition p does not lie in any target. Moreover, unless mathematical consistency warrants otherwise, we refer to a target $\{(x, y)\} \times \mathbb{S}^1$ simply as a point (x, y) in the plane. A tour corresponding to a trajectory $\gamma \in \Gamma^{\mathrm{cl}}_\Sigma(p, N_1, \ldots, N_n)$ will be called an **admissible tour**. It should be noted that an initial state for the Dubins vehicle, that is, a given position and a given orientation, is assumed to be given. It is, of course, possible to consider the initial heading of the Dubins vehicle as an independent variable of the optimisation problem as in Savla et al. [2008], for example. This minor discrepancy leads to minor only modifica- tions for any given algorithm to be applicable to either one formulation of the DTSP.

3. EXISTENCE OF SOLUTIONS AND LACK OF UNIQUENESS

Before embarking on the pursuit of an algorithm for the DTSP, it is necessary to consider the question of existence of a solution. To this end, we now recall the classification, by Dubins [1957], of the time-optimal trajectories of the control system (Σ) between any two given states. Dubins's result underlies much of the analysis of the DTSP not only in the present work, but also in a large part of the literature on problems that use the Dubins vehicle as a model.

Let C_ϕ denote a circular arc of ϕ radians and of radius ρ in the plane, and S_d a straight-line segment of length d in the plane. A C^1 concatenation of such arcs and straight-line segments is denoted by juxtaposition of the corresponding symbols. For example, $C_\alpha S_d C_\beta$ denotes a C^1 curve that consists of an arc of α radians, followed by a straight-line segment of length d, followed by an arc of β radians. Consider, now, the Dubins Problem (DP) which is the following minimum-time problem.

DP: *Given two points $p, q \in M$, minimise the time $T > 0$ over the set of trajectories $\gamma \in C^0 \cap C^\infty_{\mathrm{pw}}([0, T]; M)$ of (Σ) such that $\gamma(0) = p$ and $\gamma(T) = q$.*

The following theorem is proven in Dubins [1957]; Boissonnat et al. [1991]; Sussmann and Tang [1991].

Theorem 1. (Dubins). A solution to the Dubins Problem exists and an optimal path has to be either of the form $C_\alpha C_\beta C_\delta$ or of the form $C_\alpha S_d C_\beta$, where $0 \leq \alpha, \delta < 2\pi$, $\pi < \beta < 2\pi$, and $d \geq 0$.

A path that corresponds to a solution to the DP is called a **Dubins path**. It follows directly from Theorem 1 that the set $\Gamma^{\mathrm{cl}}_\Sigma(p, N_1, \ldots, N_n)$ in which we are seeking a solution to the DTSP is nonempty: given any permutation $\sigma \in S_k$ and any n-tuple $(p_1, \ldots, p_n) \in N_{\sigma(1)} \times \cdots \times N_{\sigma(n)}$, we can connect with a minimum-time trajectory every pair of points (p_i, p_{i+1}), $i \in \{0, \ldots, n\}$, where, for convenience, we set $p_0 = p_{n+1} = p$.

A trajectory $\gamma \in \Gamma^{\mathrm{cl}}_\Sigma(p, N_1, \ldots, N_n)$ merely satisfies the constraints of the DTSP and is not necessarily optimal. However, continuity of the solution map [3] [Aubin, 1991, Theorem 3.5.1] associated with the control system (Σ) implies that an optimal solution exists [Doyen and Quincampoix, 1997, Proposition 2.3, Theorem 3.1]. In other words, the minimum time $\tilde{T}$ for the DTSP is achieved by some admissible trajectory $\tilde{\gamma} \in \Gamma^{\mathrm{cl}}_\Sigma(p, N_1, \ldots, N_n)$. Let us now turn to the question regarding the uniqueness of solutions.

Since solutions to the DP are not unique, solutions to the DTSP are not unique, a fortiori. For example, the two tours shown in Figure 1 for an instance of the DTSP with a single target are of the same minimum length (and, therefore, correspond to the same minimum-time).

[2] To cover the entire state space M, a second chart can be chosen in an obvious manner.

[3] Recall that the solution map associated with a differential inclusion $\dot{x}(t) \in F(x(t))$, for a.e. $t \in [0, T]$, $x(t_0) = x_0$, is, by definition, the set-valued map that assigns to x_0 the, possibly empty, set of solutions to $\dot{x} \in F(x)$ with initial condition x_0.

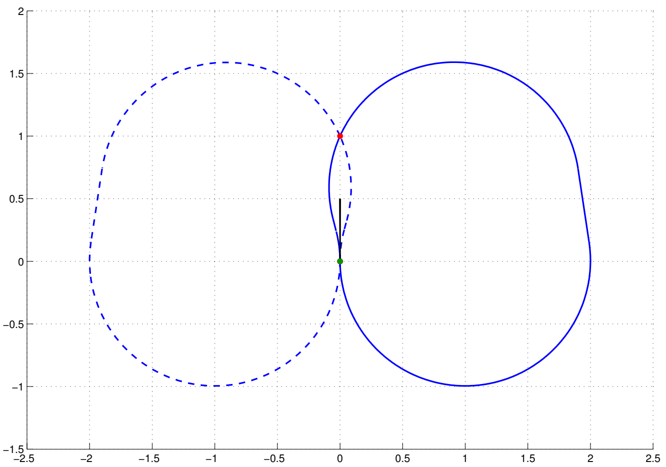

Fig. 1. Solutions to the DTSP are not unique, in general. In this (trivial) instance of the DTSP, the initial condition (green) is $(0, 0, \pi/2)$, there is one target (red) at $(0, 1)$, and the minimum turning radius is $\rho = 1$. Both tours—one shown with a solid line and the other with a dashed line—are of minimum length (which is approximately $7.5 < 2 + 2\pi$).

The formulation of the DTSP as a minimum-time control problem hints at possible algorithms for the computation of feasible solutions. The description of such an algorithm is the content of the next section.

4. AN ALGORITHM FOR THE DTSP

A direct discretisation of the DTSP, that is, partitioning the unit circle $\mathbb{S}^1$ for each one of the targets and considering all possible tours for each combination of headings leads to a computationally intractable problem even for a DTSP with a few targets. One of the reasons why this approach is computationally prohibitive is that the minimum-time function for (Σ) is discontinuous and a fine partition of $\mathbb{S}^1$ is necessary for optimal tours to be approximated (numerical experiments with one or two targets and increasingly finer partitions show that there are cases where several hundreds of points in $\mathbb{S}^1$ are necessary for the length of a tour to practically stop decreasing). This difficulty motivates the main idea behind the k-step LAA that consists, roughly, of (i) solving a finite number of smaller problems with $k < n$ targets, (ii) of keeping only part of the solution for that smaller problem, and (iii) of iterating this procedure. We call the problem of finding a minimum-time trajectory of (Σ) through k targets the "k-step look-ahead Dubins problem" (k-step LADP). A precise formulation is as follows.

k-step LADP: *Let k be a positive integer. Given a point $p \in M$ and k distinct submanifolds (targets) of the form $N_i = \{(x_i, y_i)\} \times \mathbb{S}^1 \subset M$, where $(x_i, y_i) \in \mathbb{R}^2$ and $i \in \{1, \ldots, k\}$, minimise the time $T > 0$ over the set of trajectories $\gamma \in C^0 \cap C^\infty_{\mathrm{pw}}([0, T]; M)$ of (Σ) such that $\gamma(0) = p$, $\gamma(T) \in N_k$, and $\mathrm{Im}\gamma \cap N_i \neq \emptyset$, for every $i \in \{1, \ldots, k-1\}$.*

A few comments about the k-step LADP are in order. Let $\tilde{\gamma}$ denote a solution to the k-step LADP and t_i denote the first time instant when $\tilde{\gamma}(t_i) \in N_i$, where $i = 1, \ldots, k$ and $t_k = T$.

First, it should be noted that the restriction $\tilde{\gamma}|_{[t_{k-1}, t_k]}$, that is, the part of the optimal trajectory between the last two targets can be computed much more efficiently if, instead of Theorem 1, we use the following result of Boissonnat and Bui [1994].

Lemma 2. A path that corresponds to a solution to the 1-step LADP is either of the form $C_\alpha C_\beta$ or of the form $C_\alpha S_d$, where $0 \leq \alpha < 2\pi$, $\pi < \beta < 2\pi$, and $d \geq 0$.

In our terminology, Lemma 2 classifies the solutions to the 1-step LADP and waives the need to check all possible headings at the final target when an optimal solution to the k-step LADP is computed numerically.

Second, given an instance of the DTSP with n targets, a solution to the k-step LADP with $k = n + 1$ and the final condition $\gamma(T) = p$ substituted for the condition $\gamma(T) \in N_k$ is a globally optimal solution to the DTSP.

Third, the philosophy—and the main advantage—behind the k-step LADP is that, even if k is kept small (e.g., 1 or 2) relative to n (the number of targets in the DTSP), satisfactory admissible tours can still be obtained by iteratively solving a, perhaps large, number of computationally tractable problems.

The focal point of the present paper is the "k-step look-ahead algorithm" (k-step LAA) that relies heavily on the k-step LADP, hence the name. Instead of a formal algorithmic description that would obscure the essential ideas, the k-step LAA is now described by means of a representative example that is simple enough to keep the presentation clear.

Suppose we are seeking an admissible tour to an instance of the DTSP with four targets and we intend to apply the 2-step LAA. In other words, we set $n = 4$ and $k = 2$. The first step is to construct a rooted tree whose root R represents the initial condition $p \in M \times \mathbb{S}^1$ of the Dubins vehicle and induces an orientation on the tree away from the root. The children of the root represent, temporarily, the four possible targets. This is shown in Figure 2. We say "temporarily" because this is a step-by-step description of the 2-step LAA and it is important to realise that, eventually, every node of the tree will represent a state of the Dubins vehicle: a target with a heading assigned to it. Consequently, different nodes may correspond to different headings at the same "physical" target.

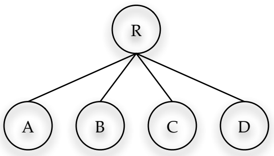

Fig. 2. The root R represents the initial condition of the Dubins vehicle and the four children represent, temporarily, the four targets.

Next, we assign headings to each one of the targets. To this end, each child of R is replicated as many times as the number of possible subsequent targets. For example, after visiting target A, there are three options: to visit either target B or C or D. This process leads to the tree in Figure 3 and it is a consequence of setting $k = 2$ because, now, each node XY, where $X, Y \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$ and $X \neq Y$, can be used to represent the target X with the heading assigned to it by solving the 2-step LADP with initial condition p, first target X, and second target Y (In the general case, i.e., when n and k are not necessarily equal to 4 and 2, respectively, R would have $n!/(n-k)!$ children).

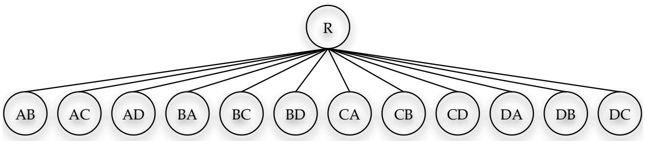

Fig. 3. Each node is duplicated as many times as the number of possible subsequent targets. The labels on the nodes represent the order in which the targets are visited.

In the notation of the previous sections, if $\tilde{\gamma}$ is a solution to such a 2-step LADP, $\tau \in]0, \tilde{T}[$ is the time when $\tilde{\gamma}(\tau) \in X$, and (x, y, θ) are local coordinates, then the child XY is assigned the heading $\theta(\tau)$. The length of the Dubins path that corresponds to $\tilde{\gamma}|_{[0,\tau]}$ is assigned as weight to the edge that connects p to XY. This assignment of headings allows us to view the node XY as the state of the Dubins vehicle that consists of the position of the target X and the heading $\theta(\tau)$. Because the heading at a target X depends on the target Y that is visited next, the grandchildren of the root p are not arbitrary. Rather, a child of a node XY has to be of the form XYZ, where $\{X, Y, Z\} \subset \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$. Figure 4 illustrates this idea which is a design choice: we could allow the children of the node XY to be of the form XWZ with W not necessarily equal to Y, however such a choice would vitiate the anticipative nature of the algorithm. It would lead, however, to a larger tree and more candidate admissible tours for the DTSP.

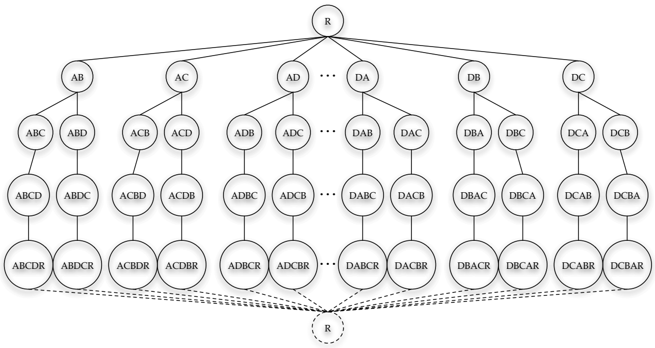

Fig. 4. The rooted tree constructed in the course of the 2-step LAA as applied to an instance of the DTSP with four targets. The edges and the node shown with dashed lines are included for clarity and they are not formally part of the tree.

Having assigned a heading to every child XY of p, we can proceed in the same manner and compute the weights between the children and the grandchildren of p. The weight of the edge between two nodes XY and XYZ is computed by solving the 2-step LADP with initial condition XY (recall that nodes represent states), first target Y and second target Z. By repeating this procedure, the tree in Figure 4 is constructed. To conform with the definition of a rooted tree, the dashed part in Figure 4 should not be considered as being formally part of the tree; it is included as a visual aid to the description of the algorithm. Towards the lower end of the tree an off-by-one issue has to be resolved, but this can be done in a straightforward manner. Specifically, the heading of a node (state) of the form $XYZWR$ has to be computed by solving a 2-step LAA with the final condition $\gamma(T) = p$ substituted for the condition $\gamma(T) \in N_k$ since the last target is always the initial condition p. In the general case, it is also necessary to reduce the look-ahead horizon k towards the final stages of the construction of the tree, simply because k will exceed the number of targets that are left to be considered. Once the tree in Figure 4 has been constructed, the final step is to find a shortest path (in the tree) from the root to the (fictitious) terminal node. For concreteness, we assume that Dijkstra's algorithm is used to this end. Dijkstra's algorithm was actually incorporated in the implementations of the 1-step and 2-step LAA that were used for the simulations in Section 5.

A number of straightforward observations regarding the k-step LAA are summarised in the following proposition.

Proposition 3. Given a DTSP with n targets, the k-step LAA returns an admissible tour of length at most

$$\mathrm{ETSP}(n) + (n+1)\kappa\pi\rho,$$

where κ is a constant and $\mathrm{ETSP}(n)$ denotes the length of the solution to the corresponding ETSP.

Proof. The k-step LAA constructs a finite rooted tree and assigns a non-negative weight to every edge and a heading to every target. Therefore, the fact that the k-step LAA terminates with an admissible DTSP tour is a direct consequence of the correctness of Dijkstra's algorithm [Cormen et al., 2001, Thm 24.6].

An instance of the DTSP can also be viewed as an instance of the ETSP and a solution σ_{ETSP} to the latter (i.e., a permutation of the targets) corresponds to a path, not necessarily a shortest one, from R to the terminal node. Because Dijkstra's algorithm returns a *shortest* path from R to the terminal node, the length $L^k_\rho(n)$ of any admissible DTSP tour through n targets found by the k-step LAA algorithm is bounded above by the length $L^k_{\rho,\mathrm{ETSP}}(n)$ of the DTSP tour that visits the targets following the order σ_{ETSP}. To quantify this bound, we observe that every admissible DTSP tour returned by the k-step LAA is a concatenation of Dubins paths. A Dubins path between two targets is of length at most $d + \kappa\pi\rho$, where d is the Euclidean distance between the targets and $\kappa \in [2.657, 2.658]$ is a constant [Savla et al., 2008, Thm 3.4], and, therefore,

$$L^k_\rho(n) \leq L^k_{\rho,\mathrm{ETSP}}(n) \leq \mathrm{ETSP}(n) + (n+1)\kappa\pi\rho, \quad (1)$$

where $\mathrm{ETSP}(n)$ denotes the length of the Euclidean travelling salesperson tour for the same set of targets.

Remark. The bound(1) is not sharp, unless $\rho \to 0$, and the reason is that its derivation does not take into consideration essential features of the k-step LAA such as the ordering of the targets independently of the solution to the ETSP and the use of a receding horizon principle.

The simulations in Section 5, especially Figure 5, provide quantitative evidence that, on the average, the left-hand side of (1) can be significantly smaller than the right-hand side.

5. SIMULATIONS

The k-step LAA can also be used simply as a receding horizon algorithm on a sequence of n targets $(N_i)_{i=1}^n$ that have been ordered by some other method, e.g., by applying an algorithm for the ETSP. This version of the k-step LAA will be called "k-step ETSP-LAA". Specifically, suppose that the targets have been reindexed so that N_i is the i-th target. Starting from the initial condition p, a solution $\tilde{\gamma}$ is found to the k-step LADP that corresponds to the first k targets. Then, only the part $\tilde{\gamma}|_{[0,t_1]}$ that connects p to N_1 is kept and the point $\tilde{\gamma}(t_1)$ is considered as a new initial condition from which the k-step LADP for the targets N_2 to N_{k+1} can be solved. This procedure is repeated until an admissible tour is constructed with the horizon k being reduced as necessary when less than k targets are left.

In what follows, the following five algorithms are compared by means of Monte Carlo simulations and the results are shown in Figures 5 to 7.

(1) **1-step ETSP-LAA:** The targets are ordered by applying an algorithm for the ETSP and the horizon is set to $k = 1$.
(2) **2-step ETSP-LAA:** The targets are ordered by applying an algorithm for the ETSP and the horizon is set to $k = 2$.
(3) **1-step LAA:** The k-step LAA of Section 4 for $k = 1$.
(4) **2-step LAA:** The k-step LAA of Section 4 for $k = 2$.
(5) **AA:** (alternating algorithm) This is the algorithm described in Savla et al. [2008].

Remark. In Ma and Castañón [2006], algorithms (1) and (2), above, are called "two-point algorithm" and "look-ahead algorithm", respectively.

For each number of targets shown in the x-axes of Figures 5, 6, and 7, 100 instances of the DTSP were randomly generated so that the initial condition and the targets were contained in $[-2.5, 2.5]^2 \times [0, 2\pi) \subset \mathbb{R}^3$ and $[-2.5, 2.5]^2 \subset \mathbb{R}^2$, respectively, with uniform distribution. Next, each algorithm was applied to all randomly generated DTSPs and the length of each tour was normalised by the length of the solution to the ETSP for the same set of targets. The y-axes correspond to the average of these normalised lengths (hence, the length of the solution to the ETSP is always equal to 1). The three figures correspond to three different minimum-turning radii. As expected, when the minimum-turning radius is small (i.e., $\rho = 0.1$) relative to the distance between the targets, the difference between the output of the five different algorithms is negligible. In all cases, however, the 2-step LAA yields the best results. Lastly, to get a sense of what the actual tours returned by the algorithms look like, an example with six targets is shown in Figure 8.

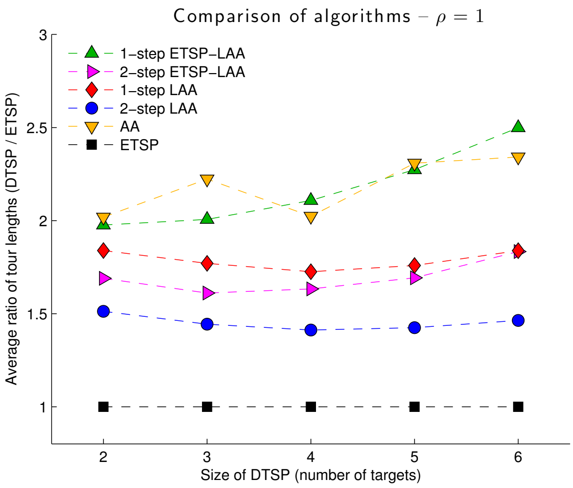

Fig. 5. Comparison of five algorithms via Monte Carlo simulations. For each number of targets shown in the x-axis, the five algorithms were applied to a common set of 100 randomly generated instances of the DTSP. The initial condition of the Dubins vehicle and the positions of the targets were generated with uniform distribution in $[-2.5, 2.5]^2 \times [0, 2\pi) \subset \mathbb{R}^3$ and $[-2.5, 2.5]^2 \subset \mathbb{R}^2$, respectively, and the minimum turning radius ρ was set equal to 1. The y-axis corresponds to the average normalised length of the DTSP tours.

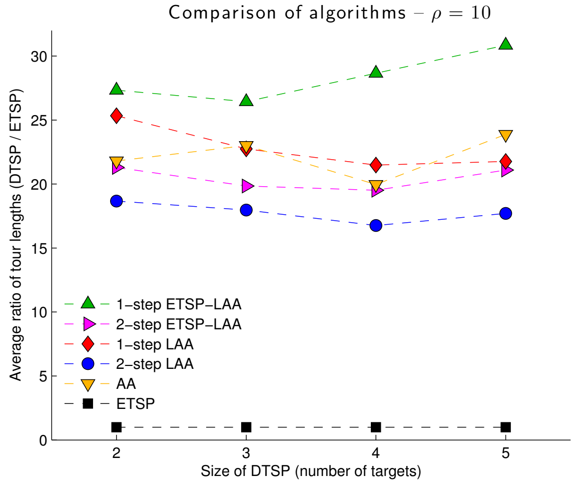

Fig. 6. Same as Figure 5, but with $\rho = 10$.

6. EXTENSIONS

Several improvements are conceivable that would allow the solution of larger instances of the DTSP using the ideas presented in this paper. For example, the k-step LAA can be combined with existing solvers for the Asymmetric Travelling Salesperson Problem (ATSP). The idea is to use the k-step LAA to generate an adjacency matrix (or list) that is subsequently provided as input to an ATSP solver. A time-consuming part of the k-step LAA is the solution of the k-step LADPs that are necessary in order to assign headings to the targets or, equivalently, in order to assign weights to the edges of the tree of the previous section. A detailed analysis of the information provided by the Maximum Principle can further reduce the number of candidate solutions to the k-step LADP and, hence, accelerate the algorithm. Work in this direction is currently in progress.

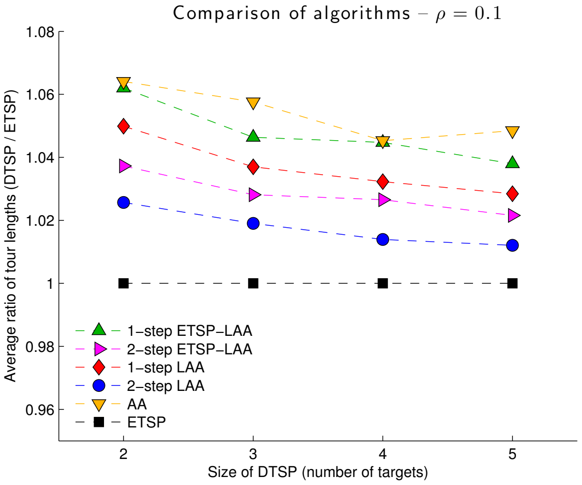

Fig. 7. Same as Figure 5, but with $\rho = 0.1$.

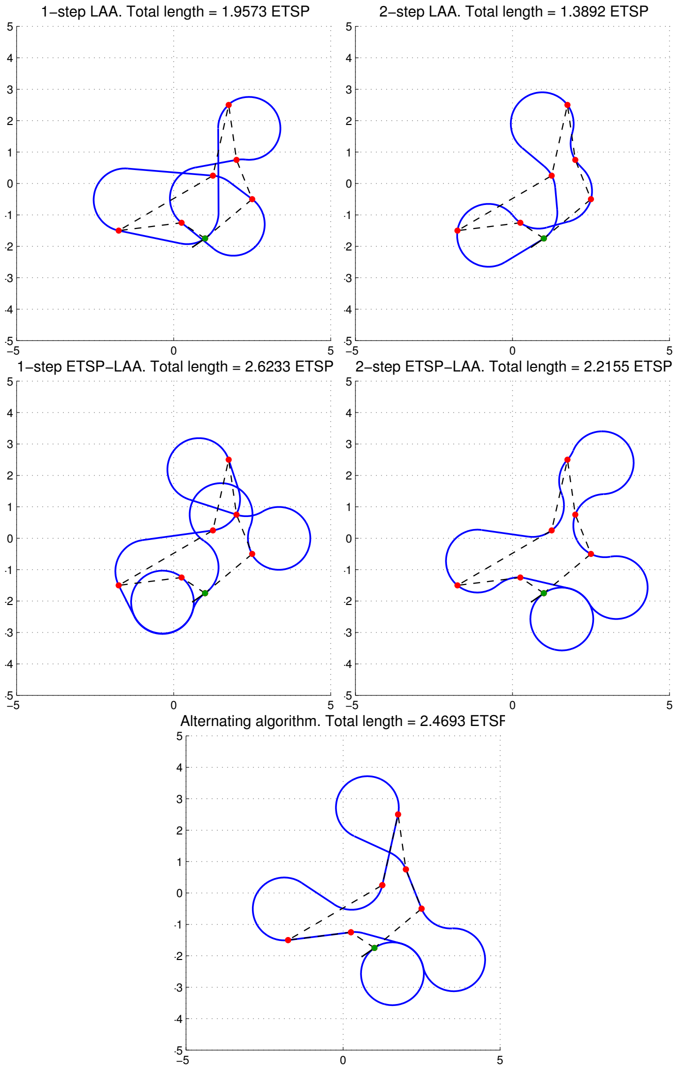

Fig. 8. One instance of the DTSP solved by five different algorithms. The setup is the same as in Figure 5.

REFERENCES

J.-P. Aubin. *Viability Theory*. Birkhäuser, 1991.

J.-D. Boissonnat and X.-N. Bui. Accessibility region for a car that only moves forwards along optimal paths. Rapport de recherche 2181, INRIA Sophia Antipolis, Janvier 1994.

J.-D. Boissonnat, A. Cérézo, and J. Leblond. Shortest paths of bounded curvature in the plane. Robotique, Image et Vision 1503, INRIA Sophia Antipolis, Juillet 1991.

M. S. Cons, T. Shima, and C. Domshlak. Integrating task and motion planning for unmanned aerial vehicles. *Unmanned Systems*, 2(1):1–20, 2013.

T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction To Algorithms*. MIT Press, 2001.

L. Doyen and M. Quincampoix. Multi-target control problems. *Journal of Optimization Theory and Applications*, 93(1):121–139, April 1997.

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*, pages 497–516, 1957.

E. Edison and T. Shima. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research*, 38(1):340–356, 2011.

J. Le Ny, E. Feron, and E. Frazzoli. On the Dubins Travelling Salesman Problem. *IEEE Transactions on Automatic Control*, 57(1):265–270, January 2012.

X. Ma and D. A. Castañón. Receding horizon planning for dubins travelling salesman problems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5453–5458, San Diego, CA, USA, December 13-15 2006.

T. G. McGee and J. K. Hedrick. Path planning and control for multiple point surveillance by an unmanned aircraft in wind. In *Proceedings of the 2006 American Control Conference*, pages 4261–4266, June 2006.

C. H. Papadimitriou. The Euclidean Traveling Salesman Problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, June 1977.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover books on mathematics. Dover Publications, 1998.

S. Rathinam, R. Sengupta, and S. Darbha. A resource allocation algorithm for multivehicle systems with nonholonomic constraints. *IEEE Transactions on Automation Science and Engineering*, 4(1):98–104, January 2007.

K. Savla, E. Frazzoli, and F. Bullo. Travelling salesperson problems for the dubins vehicle. *IEEE Transactions on Automatic Control*, 53(6):1378–1391, July 2008.

H. Sussmann and G. Tang. Shortest paths for the reeds-shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control. *Rutgers Center for Systems and Control Technical Report*, 10:1–71, 1991.

Z. Tang and Ü. Özgüner. Motion planning for multitarget surveillance with mobile sensor agents. *IEEE Transactions on Robotics*, 21(5):898–908, October 2005.

G. Yang and V. Kapila. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 1301–1306, December 2002.