Angle Bisector Algorithm and Modified Dynamic Programming Algorithm for Dubins Traveling Salesman Problem

Eswara Venkata Kumar Dhulipala and Deepak Patil

Abstract—A Dubin's Travelling Salesman Problem (DTSP) of finding a minimum length tour through a given set of points is considered. DTSP has a Dubins vehicle, which is capable of moving only forward with constant speed. In this paper, first, a worst case upper bound is obtained on DTSP tour length by assuming DTSP tour sequence same as Euclidean Travelling Salesman Problem (ETSP) tour sequence. It is noted that, in the worst case, *any algorithm that uses of ETSP tour sequence* is a constant factor approximation algorithm for DTSP. Next, two new algorithms are introduced, viz., Angle Bisector Algorithm (ABA) and Modified Dynamic Programming Algorithm (MDPA). In ABA, ETSP tour sequence is used as DTSP tour sequence and orientation angle at each point i_k are calculated by using angle bisector of the relative angle formed between the rays $i_k i_{k-1}$ and $i_k i_{k+1}$. In MDPA, tour sequence and orientation angles are computed in an integrated manner. It is shown that the ABA and MDPA are constant factor approximation algorithms and ABA provides an improved upper bound as compared to Alternating Algorithm (AA) [17]. Through numerical simulations, we show that ABA provides an improved tour length compared to AA, Single Vehicle Algorithm (SVA) [15] and Optimized Heading Algorithm (OHA) [2], [13] when the Euclidean distance between any two points in the given set of points is at least 4ρ where ρ is the minimum turning radius. The time complexity of ABA is comparable with AA and SVA and is better than OHA. Also we show that MDPA provides an improved tour length compared to AA and SVA and is comparable with OHA when there is no constraint on Euclidean distance between the points. In particular, ABA gives a tour length which is at most 4% more than the ETSP tour length when the Euclidean distance between any two points in the given set of points is at least 4ρ .

Note to Practitioners—This work is motivated from the applications such as terrain mapping and intelligence gathering etc. in which a curvature constrained vehicle such as a fixed wing air craft visits a set of target regions by starting and terminating at base station. While doing so it is required to optimize a cost such as fuel, time and fare etc. For simplified analysis, in the literature, Dubins vehicle i.e. a curvature constrained vehicle is used in place of fixed wing air craft and a point in place of target region and the resulting problem is called as Dubins Traveling Salesman Problem (DTSP). The cost of DTSP depends on the order of traversal of points as well as the angle at which the Dubins vehicle leaving and entering the points. This makes the DTSP difficult and raises the importance to the algorithms that provides a good approximate solution. To address the same, we present two algorithms ABA and MDPA that provides a good approximate solution. Through simulations, we observe that for number of points up to 70, ABA provides better tour length than AA, SVA and OHA when the Euclidean distance between any two points in the given set of points is at least 4ρ . Time complexity of ABA is comparable with AA and SVA and is better than the OHA. Also, through

simulations, we show that MDPA provides a better tour length as compared to AA, SVA and ABA and is comparable with OHA when there is no constraint on Euclidean distance between the points. MDPA is limited to 13 points due to its exponential time complexity.

I. INTRODUCTION

In a *Traveling Salesman Problem* (TSP), for a given set of points, each pair of points associate a cost with them and the salesman visits each point only once. Salesman is required to start and terminate at the same point. TSP requires to find a sequence of visiting the points, called as tour sequence that minimizes the total cost of travel. A particular case of TSP is Euclidean TSP (ETSP), in which cost associated between pair of points is Euclidean distance. TSP finds applications in many areas including but not limited to (i) Surveillance involves starting at a base station and monitoring of given set of targets and then coming back to the base station. (ii) Delivering goods involve starting at ware house and visiting all the delivery locations and coming back to ware house. TSP does not explicitly consider system dynamics into account. However, in applications like an aerial surveillance where an aircraft acts as a salesman has inherent dynamics. For more efficient path planning, in recent years, researchers introduced the system dynamics into TSP [17], [14], [6], [15], [2], [16], [10], [5].

In recent years, Dubins vehicle, a vehicle capable of moving only forward with constant speed is used in TSP due to its vast applications in motion planning in various domains including aerial surveillance. If Dubins vehicle is used as a salesman in TSP then it is called as Dubins Traveling Salesman Problem (DTSP). TSP involves finding a sequence of points such that visiting all the points and coming back to starting point incurs minimum cost. Unlike TSP, complexity of DTSP is twofold since it not only involves finding sequence of points but also orientation angle at each point. As a result, similar to TSP, DTSP is also NP hard [14]. This raises the importance of algorithms that provide a good approximate solution in reasonable amount of time. A good approximation solution is the one that is quantifiably near to the optimal tour length. Consequently, various algorithms to provide an approximate solution to DTSP in reasonable time have been proposed [17], [14], [15], [12], [11].

The contributions of this paper as follows. First, using a result from [17], we show that the worst case upper bound for any algorithm that uses ETSP tour sequence to be $(1 + \kappa\pi/\eta)OPT_E$ where OPT_E is the optimal tour length of ETSP and κ, η are positive constants. So, any algorithm that

uses ETSP tour sequence is a constant factor approximation algorithm irrespective of the orientation angles. This is useful in providing a quick solution by angle assignment and also opens up a possibility for optimizing angle assignment. Secondly, we introduce two new algorithms, viz., (i) ABA, which uses ETSP tour sequence and assigns orientation angles and (ii) MDPA, which is based on Dynamic Programming (DP) [3], [8] for calculating tour sequence and orientation angles simultaneously. We show that the ABA is a constant factor approximation algorithm when the Euclidean distance between given set of points is at least 4ρ and the upper bound on tour length is $2.57OPT_E$. The upper bound on tour length by ABA is an improved bound as compared to an upper bound by Alternating Algorithm (AA) [17]. We show that the MDPA is a constant factor approximation algorithm when the Euclidean distance between any two points in the given set of points is at least 2ρ and the upper bound on tour length is $3.04OPT_E$. The upper bound on tour length by MDPA is same as upper bound on tour length by Single Vehicle Algorithm (SVA) [15]. However, through simulations we show that the on an average tour length provided by MDPA is lower than the one provided by SVA. ABA is quick as only orientation angle assignment is required once the tour sequence is decided. However, MDPA, though giving a better tour length compared to ABA, AA, and SVA, has exponential time complexity due to use of DP. Finally, through simulations, we demonstrate the superior performance of ABA and MDPA. In numerical simulations, to compare the nearness of solution from the optimal tour length, we consider optimal tour length of ETSP (instead of DTSP) since optimal tour length of DTSP is unknown (and greater than ETSP's). A variety of efficient approximation algorithms and heuristics are available for computing ETSP tour viz., dynamic programming [3], [8], Christofides algorithm[4] and LKH algorithm[9] etc.

In simulations, we show that the ABA gives a tour length which is at most 4% more than ETSP tour length for points up to 70 and $\rho \in \{0.1, 0.5, 1\}$ when the Euclidean distance between the given set of points is at least 4ρ . For the same case, ABA also gives shorter tour lengths than AA, SVA and a discretization based algorithm called as Optimized Heading Algorithm (OHA) [2], [13] in which orientation angles at each point are discretized and then the problem is converted to an Asymmetric TSP (ATSP)¹ which is solved using existing techniques. MDPA gives shorter tour lengths compared to AA, SVA and ABA and is competitive with respect to OHA for points up to 13 and $\rho \in \{0.1, 0.5, 1\}$ when there is no constraint on Euclidean distance between the given set of points. MDPA runs in reasonable time for only up to 13 points on a standard laptop due to its use of DP leading to exponential time complexity [3], [8].

II. NOTATIONS AND PROBLEM SETUP

Dubins vehicle is a vehicle that has only forward motion capability with constant speed [7]. We assume that the speed

of Dubins vehicle to be unity, without loss of generality. The kinematic equations of Dubins vehicle are –

$$\frac{dx}{dt} = \cos \theta, \quad \frac{dy}{dt} = \sin \theta, \quad \frac{d\theta}{dt} = \frac{u}{\rho}, \quad |u| \leq 1 \quad (1)$$

where u , (x, y) and θ are control input, position and orientation angle respectively. Position $(x, y) \in \mathbb{R}^2$ along with the orientation angle $\theta \in \mathbb{S}^1$ is called the *configuration* of Dubins vehicle and is represented as $(x, y, \theta) \in \mathbb{R}^2 \times \mathbb{S}^1$. For any set S , denote $|S|$ as the cardinality of the set. We consider the Dubins Traveling Salesman Problem (DTSP) which is formally stated as follows.

Problem 1 (DTSP): Consider a collection of points $(1, 2, \dots, n)$ in the two dimensional Euclidean plane and an orientation angle α_1 at the first point 1. Let $\mathcal{I}_{n-1} := \{2, 3, \dots, n\}$ and $\mathcal{S}_{n-1}$ be the set of all possible permutations of $\mathcal{I}_{n-1}$. Find $(i_2, i_3, \dots, i_n) \in \mathcal{S}_{n-1}$ and corresponding orientation angles $(\alpha_2, \dots, \alpha_n, \alpha_{n+1})$ such that system (1) passes through the sequence of points $(1, i_2, \dots, i_n, 1)$ at the orientation angles $(\alpha_1, \alpha_2, \dots, \alpha_n, \alpha_{n+1})$ with minimum tour length. Here α_{n+1} is the orientation angle at 1 when (1) moves from i_n to 1.

Shortest path between two configurations is the building block to obtain a solution to Problem 1. Finding the shortest path between two configurations is commonly termed as Dubins problem [7]. For a given initial and final configurations, a relaxed version of Dubins problem known as *relaxed Dubins problem* is studied in [19] in which the orientation angle at final configuration is kept free. Next, we discuss the worst case upper bound on DTSP tour length if ETSP tour sequence is used as DTSP tour sequence.

III. WORST CASE UPPER BOUND

We make use of the following result from [17].

Theorem 1 ([17, Theorem 3.4]): If d and D be the Euclidean distance between given two configurations and Dubins shortest path length between the given initial and final configurations respectively then

$$D \leq d + \kappa\pi\rho, \quad \kappa \in [2.657, 2.658]$$

From $\eta\rho \leq d$ and Theorem 1, we get $D \leq (1 + \kappa\pi/\eta)d$.

Theorem 2: If an algorithm uses ETSP tour sequence as DTSP tour sequence and Euclidean distance between any two points in the given set of points is at least $\eta\rho$, $\eta > 0$ then the length L_ρ of DTSP tour with minimum turning radius ρ by the algorithm satisfies

$$L_\rho \leq \left(1 + \frac{\kappa\pi}{\eta}\right) OPT_E$$

Remark 1: In [17] authors make use of ETSP tour sequence in AA algorithm and show that the worst case tour length obtained by AA algorithm is $(1 + (5/6)(\kappa\pi/\eta))OPT_E$. In that respect, we would like to note that although the worst case bound in AA is slightly lower than the one in Theorem 2, in principle, by assigning orientation angles at each point at random the worst case bound only degrades slightly.

¹ ATSP is a TSP in which cost between the any two points is asymmetric.

For a particular value of η and $\kappa = 2.658$, we have the following.

Corollary 1: For a given set of points, if the Euclidean distance between any two points in the given set of points is at least 2ρ and DTSP tour sequence is same as ETSP tour sequence then $L_\rho \leq 5.18OPT_E$.

Corollary 2: For a given set of points, if the Euclidean distance between any two points in the given set of points is at least 4ρ and DTSP tour sequence is same as ETSP tour sequence then $L_\rho \leq 3.10OPT_E$.

From Corollaries 1 and 2, we observe that when an algorithm chooses ETSP tour sequence as DTSP tour sequence and even if the algorithm assigns orientation angles randomly then the algorithm is still a constant factor approximation algorithm.

Remark 2: Note that the bound in Theorem 2 is obtained by following ETSP tour sequence for DTSP tour and it is tighter in comparison to the one obtained in [14]. It is worth noting that, from Corollaries 1 and 2, we can obtain the upper bound on DTSP tour length by solving for ETSP. Moreover, to solve ETSP, we have efficient approximation algorithms [4] and also heuristic techniques for example, see [9]. From Corollaries 1 and 2, in the worst case one can just use ETSP tour sequence as DTSP tour sequence and assign orientation angles randomly and at the same time we can ensure the constant factor approximation upper bounds with out much computation requirements other than solving for ETSP.

IV. ANGLE BISECTOR ALGORITHM AND MODIFIED DYNAMIC PROGRAMMING ALGORITHM

In ABA, we fix the sequence of points same as ETSP tour sequence and determine the orientation angles. In MDPA, we assume no pre-ordered sequence of points to visit instead we compute the tour sequence and orientation angles simultaneously. We first describe ABA:

A. Angle Bisector Algorithm

Angle bisector algorithm steps are as follows:

- i) For the given set of points be $(1, 2, \dots, n)$, compute the ETSP tour sequence $(1, i_2, \dots, i_n, 1)$ and use it as the tour sequence for Problem 1. We assume that the α_1 is unknown unlike the assumption in the Problem 1, with out loss of generality.
- ii) Now, using the first three consecutive points $1, i_2$ and i_3 in the computed ETSP tour sequence, choose the orientation angle α_2 at i_2 as shown in Fig. 1. Alternatively, we call $1, i_2, i_3$ as A, B, C respectively. Let θ_{12} be the angle formed by the ray BA. Similarly, let θ_{32} be the angle formed by ray BC. Fig. 1 shows θ_{12} and θ_{32} . Then
$$\alpha_2 = \begin{cases} \frac{(\theta_{32} + \theta_{12})}{2} - \frac{\pi}{2} & \text{if } \theta_{32} < \theta_{12} \\ \frac{(\theta_{32} + \theta_{12})}{2} + \frac{\pi}{2} & \text{otherwise} \end{cases}$$
- iii) Repeat the step 2 for every three consecutive points in the ETSP tour sequence.
- iv) Lastly we compute orientation angle at 1 using $i_n, 1$ and i_2 .

Fig. 1. Orientation angle at i_2 for ABA

Fig. 2. DTSP tour by ABA for $\rho = 0.5$

Example 1. We consider 10 point instance from [11] and is given in Table I. ETSP instances are solved by LKH algorithm [9]. An implementation of LKH algorithm is available at <http://akira.ruc.dk/~keld/research/LKH/>. The best DTSP tour length obtained for considered 10 point instance with $\rho = 0.5$ is 17.8073 [11]. Fig. 2 shows the DTSP tour obtained by ABA for considered 10 point instance with $\rho = 0.5$. The DTSP tour length obtained by ABA is 24.0106.

Next, we show that the ABA is a constant factor approximation algorithm when the Euclidean distance between any

Fig. 3. DTSP tour by MDPA for $\rho = 0.5, \alpha_1 = \pi/6$

TABLE I
INSTANCE OF 10 POINTS

Point	1	2	3	4	5	6	7	8	9	10
x	4.45	4.10	3.81	2.42	1.15	0.09	2.22	2.28	3.03	4.75
y	0.88	2.05	2.02	3.69	3.95	4.58	4.46	4.67	4.60	3.07

Fig. 4. Four consecutive points in ETSP tour sequence

two points in the given set of points is at least 4ρ . Let $L_{ABA,\rho}$ be the tour length of DTSP by ABA with minimum turning radius ρ .

Theorem 3: If the Euclidean distance between any two points in the given set of points is at least 4ρ then the length of DTSP tour by ABA satisfies $L_{ABA,\rho} \leq 2.57OPT_E$.

Proof: To obtain the upper bound on tour length by ABA, first, we obtain the upper bound Dubins path length between two configurations. To do the same, we make use of the results presented in [18]. ABA uses the ETSP tour sequence as DTSP tour sequence. Let A, B, C and D be the four consecutive points in the ETSP tour sequence. Fig. 4 shows A, B, C and D . With out loss of generality, we assume that the line formed by B and C is an horizontal line. According to ABA, the possible orientation angles for B belongs to the quadrants I and IV on right side of line L_1 in Fig. 4. We call the quadrant I on right side of line L_1 as quadrant I of B . Similarly, we call the quadrant IV on right side of line L_1 as quadrant IV of B . According to ABA, the possible orientation angles for C belongs to the quadrants I and IV on right side of line L_2 in Fig. 4. We call the quadrant I on right side of line L_2 as quadrant I of C . Similarly, we call the quadrant IV on right side of line L_2 as quadrant IV of C .

Now, we focus on Dubins path length between B and C . Let b_{ij} denotes the class of paths having initial orientation angle in quadrant i of B and final orientation angle in quadrant j of C . In [18], equivalency group is defined as follows. Under orthogonal transformations, a class of paths are transformed into another class of paths having same length. The equivalency of two class of paths is denoted by ' $\simeq$ ', for example $b_{14} \simeq b_{41}$. From [18], in Fig. 4, $b_{11} \simeq b_{44}$ and $b_{14} \simeq b_{41}$. So, now it is enough to focus on the classes b_{11} and b_{14} .

For b_{ij} , let α be the initial orientation angle and it belongs to the quadrant i of B . Similarly, let β be the final orientation angle and it belongs to the quadrant j of C . From [18], for b_{11} , the optimal path is RSL and maximum path length of RSL occurs when $\alpha = \pi/2$ and $\beta = \pi/2$. Let $L_{11, RSL}$ be the length of path RSL for b_{11} . Let $B = (x_1, y_1)$ and $C = (x_2, y_2)$. Substituting $\alpha = \pi/2$ and $\beta = \pi/2$ in the RSL path length then we get the following upper bound.

$$L_{11, RSL} \leq d + 2\pi\rho$$

where d is the Euclidean distance between B and C . Since we have $d \geq 4\rho$,

$$L_{11, RSL} \leq \left(1 + \frac{\pi}{2}\right) d \quad (2)$$

From [18], for b_{14} , the optimal paths are RSL and RSR . The maximum of RSL occurs when $\alpha = \pi/2$ and $\beta = 0$ or 2π . The maximum of RSR occurs when $\alpha = \pi/2$ and $\beta = 3\pi/2$. For b_{14} , let $L_{14, RSL}$ and $L_{14, RSR}$ be the path lengths of RSL and RSR respectively. Substituting α and β values in the respective path lengths we get the following upper bounds.

$$\begin{aligned} L_{14, RSL} &\leq d + \rho + \frac{3\pi\rho}{2} \\ L_{14, RSR} &\leq d + (\pi + 2)\rho \end{aligned}$$

Since we have $d \geq 4\rho$,

$$L_{14, RSL} \leq \left(1 + \frac{1}{4} + \frac{3\pi}{8}\right) d \quad (3)$$

$$L_{14, RSR} \leq \left(1 + \frac{1}{2} + \frac{\pi}{4}\right) d \quad (4)$$

Let L_{BC} be the length of the Dubins path between B and C having initial orientation angle in the union of quadrants I and IV of B and final orientation angle in the union of quadrants I and IV of C . From (2), (3) and (4), we have,

$$\begin{aligned} L_{BC} &\leq \max \left(\left(1 + \frac{\pi}{2}\right) d, \left(1 + \frac{1}{4} + \frac{3\pi}{8}\right) d, \left(1 + \frac{1}{2} + \frac{\pi}{4}\right) d \right) \\ &\approx 2.57d \end{aligned}$$

We note that B and C can be any two consecutive points in the ETSP tour sequence. So, the upper bound on L_{BC} represents an upper bound Dubins path length between any two consecutive points in the ETSP tour sequence. So, we have,

$$L_{ABA,\rho} \leq 2.57OPT_E$$

We note that the upper bound in Theorem 3 is less than the one provided by AA [17].

B. Modified Dynamic Programming Algorithm

We adopt the DP proposed independently by Bellman in [3] and by Held and Karp in [8] for TSP for solving Problem 1. First, we review the DP for TSP and subsequently, we modify it such that it can handle DTSP. We use the notations given in [8].

1) *DP for TSP*: Let $(1, 2, \dots, n)$ be the given set of points. Let a_{ij} be the cost of moving from point i to point j . For a set $\mathbf{S} \subseteq \{2, 3, \dots, n\}$ and a point $l \in \mathbf{S}$, define $C(\mathbf{S}, l)$ to be the minimum cost incurred for starting from point 1, visiting all the points in the set $\mathbf{S}$ once and finally terminating at point l . Then, $C(\mathbf{S}, l)$ satisfies:

$$\begin{aligned} C(\{l\}, l) &= a_{1l}, \quad l \in \{2, 3, \dots, n\}, \quad \text{for } |\mathbf{S}| = 1 \\ C(\mathbf{S}, l) &= \min_{m \in \mathbf{S} \setminus \{l\}} [C(\mathbf{S} \setminus \{l\}, m) + a_{ml}], \quad \text{for } |\mathbf{S}| > 1 \end{aligned} \quad (5)$$

In second equation of (5), m is a point that precedes the terminal point l and $C(\mathbf{S} \setminus \{l\}, m)$ i.e, the minimum cost of visiting all cities in $\mathbf{S} \setminus \{l\}$ and terminating at m is known. Second equation of (5) iterates over all possible m and find a m that gives $C(\mathbf{S}, l)$. Let $\mathcal{C}$ be the minimum cost of a complete tour.

$$\mathcal{C} = \min_{l \in \{2, 3, \dots, n\}} [C(\{2, 3, \dots, n\}, l) + a_{l1}] \quad (6)$$

An optimal tour sequence $(1, i_2, i_3, \dots, i_n)$ is identified by the following equations.

$$\mathcal{C} = C(\{2, 3, \dots, n\}, i_n) + a_{i_n 1} \quad (7)$$

and, for $2 \leq k \leq n-1$,

$$C(\{i_2, \dots, i_{k+1}\}, i_{k+1}) = C(\{i_2, \dots, i_k\}, i_k) + a_{i_k i_{k+1}} \quad (8)$$

The optimal tour sequence is obtained as follows. First the quantities $C(\mathbf{S}, l)$ and $\mathcal{C}$ are computed by using (5) and (6) recursively. Second, the optimal tour sequence is obtained by using (7) and (8). The time complexity of DP for TSP is exponential [3], [8]. The DP assumes that $a_{ij}, i \neq j$ are available apriori to the application of algorithm. The following modification of DP for DTSP computes certain a_{ij} s when needed as the algorithm progresses.

2) *MDPA for DTSP*: We adopt the following procedure to compute $a_{ij}, i \neq j$ such that DP can handle DTSP.

- i) Compute a_{1j} for all $j = 2, 3, \dots, n$ as follows. Let α_1 be the orientation angle at 1 and is known. The orientation angle at 1 is known and we have freedom to choose the orientation angles at $2, 3, \dots, n$. So, we obtain $a_{1j}, j = 2, 3, \dots, n$ by solving the relaxed Dubins problem.
- ii) Similar to step i, whenever an a_{ij} is unavailable we compute a_{ij} by solving relaxed Dubins problem since the orientation angle at i is known and we have freedom to choose the orientation angle at j .

Let $\alpha(\mathbf{S}, l)$ denote the orientation angle at $l \in \mathbf{S}$ obtained by visiting all the points in $\mathbf{S}$ and terminating at l and starting at 1. We note that for every $C(\mathbf{S}, l)$, we store $C(\mathbf{S}, l), l$ and $\alpha(\mathbf{S}, l)$, which helps in further proceeding of DP for DTSP. The MDPA for DTSP results into a DTSP tour which is concatenations of solution of relaxed Dubins problem, which is similar to SVA. Pseudo code for MDPA is shown in Algorithm 1.

Algorithm 1 MDPA

Input: Set of points $1, 2, \dots, n$

Output: Tour sequence $1, i_2, \dots, i_n, 1$ with the corresponding orientation angles $\alpha_1, \alpha_2, \dots, \alpha_n, \alpha_{n+1}$ respectively where α_1 is known apriori.

- 1: **for** $\forall \mathbf{S} \subseteq 2^{\{2, 3, \dots, n\}} \setminus \emptyset$ **do**
 - 2: $C(\mathbf{S}, l) = a_{1l}, \quad l \in \{2, 3, \dots, n\}, \quad \text{for } |\mathbf{S}| = 1$
 - 3: $C(\mathbf{S}, l) = \min_{m \in \mathbf{S} \setminus \{l\}} [C(\mathbf{S} \setminus \{l\}, m) + a_{ml}], \quad \text{for } |\mathbf{S}| > 1$ where a_{ml} is obtained by solving relaxed Dubins problem.
 - 4: $\alpha(\mathbf{S}, l)$: Orientation angle at l obtained after visiting all the points in $\mathbf{S}$ by terminating at $l \in \mathbf{S}$ and starting at 1.
 - 5: $\mathcal{C} = \min_{l \in \{2, 3, \dots, n\}} [C(\{2, 3, \dots, n\}, l) + a_{l1}]$
 - 6: $\alpha(\{2, 3, \dots, n\}, 1)$: Orientation angle at 1 visited after i_n
 - 7: Find i_n such that $\mathcal{C} = C(\{2, 3, \dots, n\}, i_n) + a_{i_n 1}$
 - 8: $\alpha_{n+1} = \alpha(\{2, 3, \dots, n\}, 1)$
 - 9: **for** $k = n-1$ to 2 **do**
 - 10: Find i_k such that $C(\{i_2, i_3, \dots, i_k, i_{k+1}\}, i_{k+1}) = C(\{i_2, i_3, \dots, i_k\}, i_k) + a_{i_k i_{k+1}}$
 - 11: $\alpha_{k+1} = \alpha(\{i_2, i_3, \dots, i_k, i_{k+1}\}, i_{k+1})$
 - 12: Return tour sequence $1, i_2, \dots, i_n, 1$ with the corresponding orientation angles $\alpha_1, \alpha_2, \dots, \alpha_n, \alpha_{n+1}$ respectively.
-

Example 2. Consider the 10 point instance from [11] to demonstrate MDPA for DTSP. We use MATLAB implementation of DP for TSP given in [1] to implement MDPA for DTSP. Fig. 3 shows the DTSP tour obtained by MDPA for the considered 10 point instance for $\rho = 0.5$. From Fig. 3, for $\rho = 0.5$, the DTSP tour length obtained by MDPA is 17.1125, which is an improved tour length over the one obtained by Particle Swarm Optimization (PSO) in [11].

From Figs. 2 and 3, we observe that the MDPA gives shorter tour length than ABA. ABA follows the ETSP tour sequence and for the 10 point instance given in Table I, ETSP tour sequence is 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1. In Fig. 3, for $\rho = 0.5$, DTSP tour sequence by MDPA for the 10 point instance given in Table I is 1, 2, 10, 9, 8, 5, 6, 7, 4, 3, 1 which is not same as ETSP tour sequence. We observe that MDPA adjusts its tour sequence to provide improved tour length.

Next, we show that MDPA is also a constant factor approximation algorithm if the Euclidean distance between any two points in the given set of points is at least 2ρ . To do the same, we make use of upper bound on DTSP tour length by SVA from [15]. Let $L_{MDPA, \rho}$ and $L_{SVA, \rho}$ be the DTSP tour length by MDPA with minimum turning radius ρ and the DTSP tour length by SVA with minimum turning radius ρ respectively.

Theorem 4: If the Euclidean distance between any two points in the given set of points is at least 2ρ then the length of DTSP tour by MDPA satisfies

$$L_{MDPA, \rho} \leq 3.04 OPT_E$$

Proof: MDPA uses the solution to the relaxed Dubins problem to obtain DTSP tour. DTSP tour by MDPA is concatenations of solution to the relaxed Dubins problems which is similar to SVA [15]. With out loss of generality we assume that the DTSP tours by MDPA and SVA start at the same point. MDPA chooses the one that gives minimum tour length among the tours which are concatenations of solution to the relaxed Dubins problem. The tours that MDPA checks to obtain minimum length tour also includes the DTSP tour by SVA according to the MDPA procedure we adopted. So, MDPA gives either the shorter tour length then SVA or the same tour length as SVA. So, we have,

$$L_{MDPA,\rho} \leq L_{SVA,\rho} \quad (9)$$

From [15], the upper bound on DTSP tour length by SVA satisfies

$$L_{SVA,\rho} \leq 3.04OPT_E \quad (10)$$

So, from (9) and (10) we have the theorem statement. ■

V. PERFORMANCE AND RUN TIME ANALYSIS

For the numerical simulations, we run 100 iterations and consider average value of tour length and computation time. In each iteration, same set of points is given as input to every algorithm AA, SVA, ABA, MDPA and OHA. We note that the for OHA, we consider 5 angles at each point which are equally spaced in the interval $[0, 2\pi]$. We consider two cases for the analysis. First one is when there is no constraint on Euclidean distance between any two points in the given set of points and is limited to 13 points due to the exponential time complexity of MDPA. Second one is when the Euclidean distance between any two points in the given set of points is at least 4ρ with $\rho \in \{0.1, 0.5, 1\}$. In second case, we consider up to 70 points and for performance comparison, we consider only AA, SVA, ABA and OHA. Since the MDPA does not execute in reasonable time for more than 13 points, it is excluded in second case. All the algorithms presented in this paper are implemented in MATLAB R2020b and run on a laptop configured with Intel core i5, 1.70 GHz CPU and 8 GB RAM.

A. Case 1: no constraints

Points are generated according to uniform probability distribution in a square region $[-2.5 \ 2.5] \times [-2.5 \ 2.5]$. Figs. 5, 6 and 7 show the performance and computation time of AA, SVA, ABA, MDPA and OHA for $\rho \in \{0.1, 0.5, 1\}$. In Figs. 5, 6 and 7, DTSP tour length is normalized by corresponding ETSP tour length. From Fig. 5 (a), for $\rho = 0.1$, the OHA and ABA outperforms remaining algorithms and is at most 3% more than the ETSP tour length. From Figs. 6 and 7, MDPA is competitive with OHA and performs better than AA, SVA and ABA. We note that the time complexity of MDPA is higher than the OHA. From Figs. 5, 6 and 7, run time of MDPA rises exponentially, which is consistent with exponential time complexity of DP for TSP [8]. From Figs. 5, 6 and 7, we observe that computation time of AA, SVA and ABA are almost same. Table II shows the standard deviations

on DTSP tour length by AA, SVA, ABA, MDPA and OHA for $n = 3, \dots, 13$ and $\rho \in \{0.1, 0.5, 1\}$. From Table II, the standard deviations by AA, SVA, ABA, MDPA and OHA for $\rho = 0.1$ are similar. From Table II, for $\rho = 0.5, 1$, the MDPA and OHA give less value of standard deviation as compared to AA, SVA and ABA. This means that most of the times DTSP tour length by MDPA and OHA is not far from the mean as compared to AA, SVA and ABA. Finally, we conclude that in case 1, MDPA outperforms AA, SVA and ABA and is competitive with respect to OHA at the price of higher time complexity. We note that the MDPA ensures an upper bound on DTSP tour length as given by the Theorem 4 and is not the case with OHA. OHA is sensitive to number of discretizations and the corresponding discretized angles [2]. So MDPA has an advantage of proven upper bound as compared to OHA. From Fig. 5, for lower turning radius i.e. $\rho = 0.1$, ABA outperforms AA, SVA and MDPA and is competitive with respect to OHA. This can be thought as a superior performance of ABA when Euclidean distance between any two points in the given set of points is higher than the turning radius. To further show the superior performance of ABA over other algorithms, in the case 2, we consider the points such that the Euclidean distance between any two points in the given set of points is at least 4ρ and $\rho \in \{0.1, 0.5, 1\}$. We note that, when Euclidean distance between any two points in the given set of points is less than 4ρ and $\rho \in \{0.1, 0.5, 1\}$, the DTSP tour length by ABA is higher as compared to AA, SVA and OHA and are not reported here.

B. Case 2: at least 4ρ distance

Points are generated according to uniform probability distribution in a square region $[-20 \ 20] \times [-20 \ 20]$. In this case, we consider increased area of the region i.e. $[-20 \ 20] \times [-20 \ 20]$ as compared to case 1 to generate higher number of points compared to case 1 and also to ensure Euclidean distance between any two points in the give set of points is at least 4ρ , $\rho \in \{0.1, 0.5, 1\}$. When the Euclidean distance between any two points in the give set of points is at least 4ρ , $\rho = 1$, we could generate up to 70 points in the region considered and hence, in this case we consider points up to 70 for simulation. We note that the MDPA is not employed in the present case since the MDPA does not run for more than 13 points in a reasonable time on a standard laptop. Figs. 8, 9 and 10 show the performance and computation time of AA, SVA, ABA and OHA for $\rho \in \{0.1, 0.5, 1\}$. In Figs. 8, 9 and 10, DTSP tour length is normalized by corresponding ETSP tour length. From Figs. 8 (a), 9 (a) and 10 (a), for $\rho = 0.1, 0.5$ and 1, the ABA outperforms AA, SVA and OHA. From Figs. 8 (a), 9 (a) and 10 (a), for $\rho = 0.1, 0.5$ and 1, DTSP tour length by ABA is at most 4%, 2% and 4% more than the ETSP tour length respectively. From Figs. 8 (b), 9 (b) and 10 (b), for $\rho = 0.1, 0.5$ and 1, the computation time of AA, SVA and ABA are almost same and OHA takes more time as compared to AA, SVA and ABA. From Table III, for $\rho = 0.1$ and 0.5, the standard deviations by AA, SVA, ABA and OHA

TABLE II
STANDARD DEVIATIONS BY AA, SVA, ABA, MDPA AND OHA FOR $n = 3, \dots, 13$ AND $\rho = \{0.1, 0.5, 1\}$

n	$\rho = 0.1$					$\rho = 0.5$					$\rho = 1$				
	AA	SVA	ABA	MDPA	OHA	AA	SVA	ABA	MDPA	OHA	AA	SVA	ABA	MDPA	OHA
3	2.2393	2.2610	2.2361	2.2473	2.2437	2.2081	2.4175	2.1949	2.4071	2.2701	2.3369	2.4483	3.2303	2.3172	2.2333
4	2.2244	2.2613	2.2322	2.2601	2.2329	2.1874	2.02717	2.1176	1.9873	2.0058	2.4234	3.6300	3.9959	2.6866	1.9990
5	1.9527	1.9314	1.9334	1.9361	1.9331	2.0171	2.2669	2.2024	2.0539	1.8963	3.2733	3.8328	4.6786	2.4607	2.3437
6	1.9704	1.9473	1.9444	1.9633	1.9469	2.1463	2.3469	2.4598	2.0634	1.8381	3.5177	4.4341	5.5945	2.3211	1.9458
7	1.8000	1.8208	1.7957	1.8208	1.8250	2.6267	2.4840	2.9310	2.0269	1.8404	3.9675	4.9725	6.0547	2.5930	1.9834
8	1.6291	1.6010	1.5773	1.6151	1.6026	2.5238	2.6961	2.9602	2.2419	1.6827	3.2916	4.7994	6.2753	2.7088	2.1703
9	1.5089	1.4876	1.4849	1.4824	1.4476	2.5509	2.8477	3.9675	1.9087	1.6826	4.3137	5.3392	6.9317	2.1478	2.1960
10	1.6655	1.6825	1.6390	1.6501	1.6570	2.7816	3.0500	3.8202	1.8676	1.9508	4.5706	5.5668	6.6408	2.5096	2.0488
11	1.5846	1.6192	1.5832	1.6261	1.6095	2.8852	2.9586	4.1059	1.8370	1.7460	4.9266	5.7128	6.7950	2.4445	2.5121
12	1.5048	1.5017	1.4824	1.5089	1.4844	2.8138	3.3313	4.1829	2.1415	1.9949	3.8783	5.6132	8.3713	2.4019	2.2571
13	1.8599	1.9125	1.8177	1.9076	1.8869	2.7411	3.4784	4.0436	1.8812	1.9535	3.9745	6.2464	8.1174	2.5446	2.2557

Fig. 5. Demonstration of performance and run time of AA, SVA, ABA, MDPA and OHA for $\rho = 0.1$

Fig. 6. Demonstration of performance and run time of AA, SVA, ABA, MDPA and OHA for $\rho = 0.5$

(a) n vs. Ratio of tour length

(b) n vs. Average computation time by AA, SVA, ABA and OHA

(c) n vs. Average computation time by MDPA

Fig. 7. Demonstration of performance and run time of AA, SVA, ABA, MDPA and OHA for $\rho = 1$

are similar. From Table III, for $\rho = 1$, the standard deviations by ABA and OHA are similar. From Table III, for $\rho = 1$, the standard deviations by ABA and OHA are less as compared to AA and SVA. This means that the most of the times DTSP tour length by ABA and OHA is not far from the mean as compared to AA and SVA. In conclusion, ABA performs better than AA, SVA and OHA when the Euclidean distance between any two points in the given set of points is at least 4ρ .

VI. CONCLUSION

First, we noted that any algorithm that makes use of ETSP tour sequence is a constant factor approximation algorithm. This is beneficial in making it viable to explore easy to compute choices of orientation angles at each point. In fact, even if one assigns orientation angles randomly at each point,

(a) n vs. Ratio of tour length

(b) n vs. Average computation time

Fig. 8. Demonstration of performance and run time of AA, SVA, ABA and OHA for $\rho = 0.1$

(a) n vs. Ratio of tour length

(b) n vs. Average computation time

Fig. 9. Demonstration of performance and run time of AA, SVA, ABA and OHA for $\rho = 0.5$

TABLE III
STANDARD DEVIATIONS BY AA, SVA, ABA AND OHA FOR $n = 10, \dots, 70$ AND $\rho = \{0.1, 0.5, 1\}$

n	$\rho = 0.1$				$\rho = 0.5$				$\rho = 1$			
	AA	SVA	ABA	OHA	AA	SVA	ABA	OHA	AA	SVA	ABA	OHA
10	13.9920	14.0037	13.9718	13.9627	12.1544	12.1359	12.1237	12.1099	13.0498	13.1531	12.5759	12.5775
20	12.3793	12.3810	12.3497	12.3846	11.4688	11.5664	11.3303	11.3398	9.0770	9.1108	8.7442	8.8044
30	12.0439	12.0242	11.9988	12.1147	9.0185	9.0254	8.9234	8.9507	9.2432	9.2525	8.5854	8.5880
40	9.8541	9.8507	9.8522	9.9057	8.8362	8.8809	8.7706	8.8072	7.1369	7.3896	6.3913	6.4243
50	10.9911	10.9996	10.9803	10.9229	8.9732	8.9855	8.7082	8.7518	7.3265	7.3232	5.8702	5.8499
60	9.6952	9.7139	9.6421	9.5832	7.9213	7.9665	7.7645	7.8112	6.5098	6.9776	4.7853	4.8604
70	9.5610	9.5394	9.5314	9.6339	8.4124	8.5059	7.8976	7.8111	5.8703	6.3828	3.8801	3.6694

(a) n vs. Ratio of tour length

(b) n vs. Average computation time

Fig. 10. Demonstration of performance and run time of AA, SVA, ABA and OHA for $\rho = 1$

the worst case tour length is less than $(1 + k\pi/\eta)OPT_E$. Next, we introduced two new algorithms for DTSP, ABA and MDPA. We showed that the ABA and MDPA are competitive with the existing algorithms and in particular, ABA outperforms AA, SVA, and OHA when the Euclidean distance between any two points in the given set of points is at least 4ρ . In particular, for ABA, we showed that the DTSP tour length is at most $2.57OPT_E$ when the Euclidean distance between any two points in the given set of points is at least 4ρ and is an improved bound as compared to the one provided by AA. On the other hand, MDPA is suited for points up to 13 without constraint on Euclidean distance between the points but with higher time complexity. Also MDPA ensures that the tour length is not more than the $3.04OPT_E$ when the Euclidean distance between any two points in the given set of points is at least 2ρ . Exploring the use of MDPA for offline computations and for cloud

based computational paradigms, in situations where cost of utilizing external on-demand computational resources is cheaper compared to cost of performance degradation is currently being pursued.

REFERENCES

- [1] Ewing Kang (2020). EwingKang/Dubins-Curve-For-MATLAB (<https://www.github.com/EwingKang/Dubins-Curve-For-MATLAB>), GitHub. Retrieved April 8, 2020.
- [2] Luitpold Babel. New heuristic algorithms for the dubins traveling salesman problem. *Journal of Heuristics*, pages 1–28, 2020.
- [3] Richard Ernest Bellman. Dynamic programming treatment of the traveling salesman problem. 1961.
- [4] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.
- [5] Izack Cohen, Chen Epstein, Pantelis Isaiiah, Saar Kuzi, and Tal Shima. Discretization-based and look-ahead algorithms for the dubins traveling salesperson problem. *IEEE Transactions on Automation Science and Engineering*, 14(1):383–390, 2016.
- [6] Eswara Venkata Kumar Dhulipala and Deepak Patil. Time optimal tour through a class of points for a double integrator traveling salesman. In *2019 18th European Control Conference (ECC)*, pages 3248–3253. IEEE, 2019.
- [7] Lester 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(3):497–516, 1957.
- [8] Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.
- [9] Keld Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.
- [10] Sleiman Itani, Emilio Frazzoli, and Munther A Dahleh. Travelling salesperson problem for dynamic systems. *IFAC Proceedings Volumes*, 41(2):13318–13323, 2008.
- [11] Richard J Kenefic. Finding good dubins tours for uavs using particle swarm optimization. *Journal of Aerospace Computing, Information, and Communication*, 5(2):47–56, 2008.
- [12] Xiang Ma and David A Castanon. Receding horizon planning for dubins traveling salesman problems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5453–5458. IEEE, 2006.
- [13] Satyanarayana G Manyam and Sivakumar Rathinam. On tightly bounding the dubins traveling salesman’s optimum. *Journal of Dynamic Systems, Measurement, and Control*, 140(7), 2018.
- [14] Jerome Ny, Eric Feron, and Emilio Frazzoli. On the dubins traveling salesman problem. *IEEE Transactions on Automatic Control*, 57(1):265–270, 2012.
- [15] Sivakumar Rathinam, Raja Sengupta, and Swaroop Darbha. A resource allocation algorithm for multivehicle systems with nonholonomic constraints. *IEEE Transactions on Automation Science and Engineering*, 4(1):98–104, 2007.
- [16] Ketan Savla, Francesco Bullo, and Emilio Frazzoli. Traveling salesperson problems for a double integrator. *IEEE Transactions on Automatic Control*, 54(4):788–793, 2009.
- [17] Ketan Savla, Emilio Frazzoli, and Francesco Bullo. Traveling salesperson problems for the dubins vehicle. *IEEE Transactions on Automatic Control*, 53(6):1378–1391, 2008.

- [18] Andrei M Shkel and Vladimir Lumelsky. Classification of the dubins set. *Robotics and Autonomous Systems*, 34(4):179–202, 2001.
- [19] B. Thomaschewski. Dubins problem for free terminal direction. Preprint M09/2001, Technical University of Ilmenau, Ilmenau, Germany, 2001.