On the Stochastic TSP for the Dubins Vehicle

Sleiman Itani, Munzer A. Dahleh

***Abstract*— In this paper, we study the Stochastic Travelling Salesperson Problem for a vehicle that is constrained to move forwards with a bound on the minimum turning radius. For n uniformly distributed targets, it is known that the expected length of the optimal tour has a lower bound that is $kn^{2/3}$ (where k is a constant that depends on the geometry of the area). We design a novel algorithm that performs within a constant factor of that lower bound. This proves that the expected length of the optimal tour of the Stochastic Dubins Travelling Salesperson Problem (SDTSP) is of the order of exactly $n^{2/3}$.**

I. INTRODUCTION

The Travelling Salesperson Problem is defined as follows: Given a graph of $n < \infty$ points , find the Hamiltonian circuit that incurs the minimum cost. Here, the cost is defined as the sum of the weights of the individual edges of the circuit. The weights of the edges can be anything that is non-negative. They can be symmetric or asymmetric, satisfy the triangular inequality or not, or they can have special structure that makes the problem solvable in polynomial time in some instances [4] [6] [8]. The more structure that is added, the easier the problem becomes. The Euclidean TSP is widely studied and has polynomial time algorithms that produce a tour whose length is within $(1+\epsilon)$ times the shortest tour for any ϵ [8].On the other hand, we know of no way that can closely approximate TSP problems with little or no structure in polynomial time.

The Dubins vehicle is a nonholonomic vehicle with a lower bound on its turn radius, which we will call ρ. It is an appropriate model for many vehicles and robots, especially aerial and marine vehicles. The name follows the mathematician who was the first to prove that under average curvature constraints (or minimum radius constraints), the following is true: Given any two points p_i and p_j, and the headings at those points, the paths with minimum distance always exist. He also proved that the optimal path always takes one of two forms (C,C,C), (C,S,C) or any truncation of those two forms. Here, C stands for a circular arc (clockwise or anticlockwise) whose radius is the minimum turning radius (ρ), and S stand for a straight line segment [3].

In both forms of the Dubins path ((C,C,C) and (C,S,C)); the first (last) curve is a part of the circle of radius ρ that is tangent to the given heading at the first (last) point. Two such circles can be constructed for each point, on different sides of the line indicating the heading. One of these circles has clockwise orientation and the other has anti-clockwise orientation so that the orientation of the circle would also match the orientation of the heading at the point. So, for a given pair of points and their headings, there are four (C,S,C) curves that can be constructed to connect the points. Only two (C,C,C) curves are possible because for a (C,C,C) curve, since the orientation of the first and last arcs has to be the same. The optimal path is usually determined by trying all six configurations and choosing the one that has the least length. There are some studies that specify which path is optimal for any configuration of points and headings [9]. It is important to note that the Dubins metric (not really a metric) becomes close to the Euclidean metric when the points are very far away, but it has radically different characteristics as the Euclidean distance between the points becomes comparable to ρ.

The TSP problem for the Dubins vehicle (DTSP) is a TSP problem where the path can be traversed by a Dubins vehicle. This implies two requirements, the first is that between any two points, the path taken is a Dubins curve. The second requirement is that the headings of the two Dubins curves that meet at the same point are the same. So for a fixed heading at each point, the DTSP is reduced to an asymmetric TSP problem where the distances satisfy the triangular inequality. Finding the optimal headings is hard, because changing the heading at one point changes the distance from that point to all other points. This means that we have two NP problems (TSP and minimizing a non-convex function) that are interwind. The Stochastic DTSP problem is the problem in which the points are generated by a random process, and the cost is the expected value of the tour length.

The DTSP and the Stochastic DTSP are coming into the picture because of advances in robotics and the growth of interest in Unmanned Aerial Vehicles (UAV's). The possible use of robots and UAV's in search and rescue missions, surveillance and many other applications that require optimized planning of a route make this problem important for the near future. The Dubins vehicle is the natural simple model for many of those vehicles and robots, and thus it is the appropriate model to use for path planning . Studying the DTSP and the Stochastic DTSP might also offer insight to the solution of different problems where there are still constraints on the curvature. This is a more general class of applications that allows the cost function to be modified to account for areas of danger, priority between customers and many other applications that seem to be dawning with the feasibility of making commercial autonomous vehicles and robots.

Our work will mainly build on the work of Savla, Frazzoli and Bullo in [1],[2]. They studied the same problem, established a tight lower bound and an upper bound, and provided and algorithm that results in a tour of expected length $O(n^{2/3}\ln((n))^{1/3})$[1]. So the problem that we are addressing is the following: Given a rectangle R(of area A, height H and width W) and n points randomly distributed in A (where n is large), we want to find the expected optimal length of the Dubins vehicle path through all of the points, and an algorithm that performs close to the expected optimal length.

The rest of this paper will be organized in the following manner: In section 2, we will introduce some of the notation and prepare the ground for our study by citing some results that are relevant. We will also provide some points that are direct extensions to some results, and prove other facts that we will be needing. In section 3, we will solve an essential auxiliary problem and introduce the " Scanning Algorithm", which will be an integral element in our constant factor algorithm. In section 4, we describe the "Iterated Level Algorithm" and prove that it performs within a constant factor of the lower bound. Section 5 has the conclusions and 6 has the future work.

II. Background and notation

In this section we mainly introduce the notation we will use and provide direct extensions of previous results that are necessary for our proofs. Most of the results are from those on the point to point Dubins vehicle problem and some related problems. Since the Dubins curve, and thus Dubins distance, depend on the headings at each target; a target is usually augmented with an angle and is represented in $(\mathbb{R}^2, S)$ where S=[0,2π] represents the angle that the heading vector at the point makes with the x-axis.

A. Optimal tour length lower bound:

From [1], given n targets that are uniformly distributed in a region, the expected length of the optimal TSP tour by a Dubins vehicle over all of the points is $\Omega(n^{2/3})$. For the same problem, if the vehicle does not have any dynamic constraints (it can change directions instantaneously); then the expected length of the optimal TSP tour is $\Theta(\sqrt{n})$.

B. Distance between close points for a Dubins vehicle:

Let a straight line L, and a Dubins vehicle with minimum turning radius ρ at a point (p, ϕ) such that the heading at p is parallel to L (if L is parallel to the x-axis, then $\phi = 0$) . Denote the Euclidean distance from p to L (the length of the projection of p on L) by d and the point that is the projection of p on L by $p\prime$. Denote the shortest curve that the vehicle has to follow to go from (p, ϕ) to L and keep moving on L by C.

The requirement that the vehicle has to keep moving on L means that C should be tangent to L at the point of their intersection (p_f). Let $|C|$ be the length of C from p till p_f and $|L|$ the distance between $p\prime$ and p_f .

[1]We say a function $f(n)$ is $O(g(n))$ if there is a $c > 0$ such that $\lim_{n\to\infty} \frac{f(n)}{g(n)} \leq c$ (the limit could be 0), we say $f(n)$ is $\Omega(g(n))$ if $g(n)$ is $O(f(n))$ and we say $f(n)$ is $\Theta(g(n))$ if there is a $c > 0$ such that$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$. We say $f(l)$ is $o(l)$ if $\lim_{l\to 0} \frac{f(l)}{l} \to 0$

Proposition 1:
as $d \to 0$, $|C| \leq |L| + o(|L|)$.

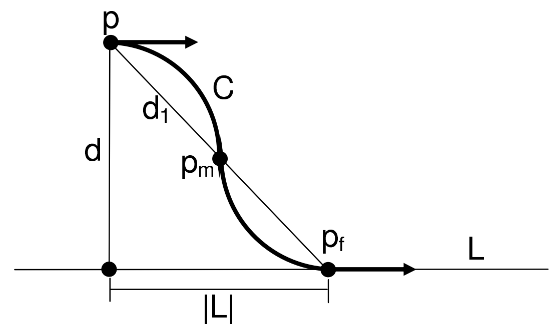

Figure 1: Optimal path from a point to a straight line

This means that if a Dubins vehicle is moving parallel to a line and close to it, it can go onto the line and continue moving on it with almost no loss compared to the case where the vehicle is on the line from the beginning.

To prove this proposition we will have three steps:

1) Determine C.
2) Prove that $|L| = 2\sqrt{\rho d} - o(d)$.
3) Prove that $|C| \leq 2\sqrt{\rho d} + o(d)$.

The first step is a known result: If $d < \rho$; then the shortest trajectory of the vehicle is made up of two circular arcs with radius ρ [5]. The first arc is tangent to the straight line passing through p and parallel to L, the second is tangent to L and both arcs are tangent to each other at the point of their intersection.

The second and third steps can be achieved by using Figure 1 and a bit of basic geometry: Let p_m be the point of intersection of the two arcs in C. By symmetry, p_m is the middle point on C between p and p_f. Denote the Euclidean distance between p and p_m by d_1; we have the following:

$$|L|^2 = 4d_1^2 - d^2$$

$$= 4(2\rho^2 - 2\rho^2 \frac{(\rho - d/2)}{\rho}) - d^2 = 4\rho d - d^2$$

Therefore as $d \to 0$,

$$|L| = 2\sqrt{\rho d} - o(d)$$

$$|C| = 2\rho * \arcsin(\frac{|L|/2}{\rho}),$$

so as $d \to 0$, $|C| \leq 2\sqrt{\rho d} + o(d)$

More importantly, as $d \to 0$, $|C| \leq |L| + o(|L|)$ This proves the proposition.

III. Scanning Algorithm

In this section, we will try to solve the following auxiliary problem:

Let R be a rectangular area with height H and Width W that is covered by m^2 congruent "initial" rectangles (i-rectangles), each of which has a height $l = \frac{H}{m}$ and width

$w = \frac{W}{m}$. *Also, let i of the m^2 i-rectangles have one target in each of them and $(m-i)$ be empty. For large i and m, and if the distribution of the non-empty rectangles is uniform, design an algorithm that determines a tour for a Dubins vehicle so that it visits at least constant factor of the i targets and has an expected tour length that is $O(i^{2/3})$.*

We now introduce the Scanning Algorithm, which will solve the problem above. We begin with some terminology and definitions that will be needed in the algorithm and bound proofs.

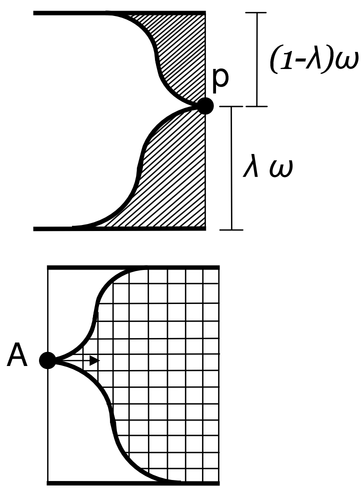

Figure 2

This figure shows segments of strips. The border lines are guidance lines. The curves are the shortest paths from the vehicle to the guidance lines.

1) (Top)p will be retrieved unless the vehicle is in the shaded area.
2) (Bottom) The Scanning Area of the vehicle at point A.

A. Algorithm Terminology and Definitions:

In the Scanning Algorithm, we will divide the whole rectangle R into strips. Let $k_1 > 0$ be a parameter of the algorithm (to be determined later). With no loss of generality, the strips have been chosen to be parallel to H. It will be shown that it is optimal that the strips are parallel to the longer of the sides. After dividing R into $\frac{\rho^{1/3}W(k_1 i)^{2/3}}{\sqrt[3]{(WH)^2}}$ strips, the width of each strip will be:

$$w_s(i) = \frac{\sqrt[3]{(WH)^2}}{\rho^{1/3}(k_1 i)^{2/3}}. \tag{1}$$

We will now define some terms that we will use in the algorithm:

1) **Guidance Lines:**
   The lines that are between the strips are called guidance lines, since the vehicle will move in between them.
2) **Retrieval Area:**
   When the vehicle is in strip i, the retrieval area is the set of points in strip i that the vehicle can reach without going out of strip i (Figure 2).

B. Algorithm Description:

The scanning Algorithm can be described as follows:

We will assume that H is horizontal and W is vertical. We divide the area into strips that have width $w_s(i)$, and number them from the bottom up. The vehicle moves bottom along the length of the strips, on the same horizontal level as the closest target in the retrieval area. It moves to that level using a minimal length curve as in Figure 1. When the vehicle finishes strip j, it moves to strip $j+1$ and moves on it in the opposite direction.

C. Constant factor:

Consider a target p in any strip, with distances $\lambda w_s(i)$ and $(1-\lambda)w_s(i)$ from the edges of the strip (Figure 2). If the shaded region ($\mathbb{S}$) in Figure 2 is empty, the vehicle will visit p. By symmetry, the areas of the $\mathbb{S}$ is $\sqrt{\rho}(\lambda w_s(i))^{3/2} + \rho[(1-\lambda)w_s(i)]^{3/2}$.

To bound the probability that $\mathbb{S}$ is empty, we use the following lemma:

Lemma 1: For large i and m, the probability that $\mathbb{S}$ is empty under the given distribution of targets is greater than the probability that $\mathbb{S}$ is empty if the targets were uniformly distributed.

Proof: The proof considers the generation of the targets, conditioned on the fact that p already exists.

Under the uniform distribution, every target other than p has a probability of $\frac{\sqrt{\rho}(\lambda w_s(i))^{3/2}+[(1-\lambda)w_s(i)]^{3/2})}{WH} = \frac{\lambda^{3/2}+(1-\lambda)^{3/2}}{k_1 i}$ of being in $\mathbb{S}$.

Under the given distribution, the first target has a probability of $\frac{\sqrt{\rho}(\lambda w_s(i))^{3/2}+[(1-\lambda)w_s(i)]^{3/2})-\alpha}{WH}$, where α is the area of the intersection of the *i*-rectangle containing p and $\mathbb{S}$. The consequent targets have less probability than the first, because of the assumption that i and m are large.

Therefore, the probability that $\mathbb{S}$ contains $i_0 > 0$ targets (other than p) is less than the probability that $\mathbb{S}$ contains i_0 targets if the targets were uniformly distributed. The result follows directly. ■

The number of targets in $\mathbb{S}$ under the uniform distribution is a Poisson variable with parameter:

$$\frac{\lambda^{3/2}+(1-\lambda)^{3/2}}{k_1}$$

Therefore, the probability that $\mathbb{S}$ is empty (given p) is greater than:

$$e^{-\frac{\lambda^{3/2}+(1-\lambda)^{3/2}}{k_1}}$$

Since λ is uniformly distributed between 0 and 1, the probability that any target p will be visited by is greater than:

$$\int_0^1 e^{-\frac{\lambda^{3/2}+(1-\lambda)^{3/2}}{k_1}} d\lambda = 2\int_0^{1/2} e^{-\frac{\lambda^{3/2}+(1-\lambda)^{3/2}}{k_1}} d\lambda$$

Now, since $-\lambda^{3/2} - (1-\lambda)^{3/2}$ is monotonic increasing between 0 and $1/2$, and the exponential function is convex, we can use Jensen's inequality, and get our bound:

$$P \geq e^{-\frac{4}{5k_1}}$$

This means that the Scanning Algorithm will allow the vehicle to retrieve at least $e^{-\frac{4}{5k_1}} i$ of the i targets.

D. Bound on the length of the SA tour:

Lemma 1:

The maximum distance travelled by the scanning algorithm is less than:

$$\frac{WH + Wc_1}{w_s(i)} + H + c_2 + o(\frac{1}{\sqrt{w_s(i)}})$$

Proof:

First, we will start by finding the distance travelled when crossing one guidance line: For a certain pass on one of the guidance lines in which we retrieve k targets, we denote the retrieval cost (as defined before) of the j^{th} target by L_j and the distance on the guidance line that was skipped by the retrieval by l_j. We have:

$$\sum_{j=1}^{k} l_j + l_f = H.$$

Where l_f is the distance travelled on the guidance line. We also have, because of the assumption that i is large ($w_s(i) < \rho$):

$\forall i$,

$$L_j \leq l_j + o(\sqrt{w_s(i)}).$$

Therefore

$$\sum_{j=1}^{k} L_j + l_f \leq H + o(\sqrt{w_s(i)}).$$

Where the reason the last term is $o(\sqrt{w_s(i)})$ can be proved by proving that the longest pass will be when $(\sqrt{\frac{H}{l_c(i)}})$ points are retrieved and the distance to each of them from the guidance line is $w_c(i)$. The total length will therefore be of $o(\sqrt{w_s(i)})$.

Therefore the distance travelled in traversing one guidance line is bounded by

$$D_{GL} = H + o(\sqrt{w_s(i)}).$$

To turn from one guidance line to the other, an additional distance bounded by $D_{turn} = c_1 + w_s(i)$ is needed; where $c_1 < 2.658\pi\rho$ [2].

The number of guidance lines is

$$N_g \leq \frac{W}{w_s(i)} + 1$$

To go back to the beginning of first guidance line, we must travel at most the diagonal of the square and some distance for changing direction. This total distance is bounded by

$$c_2 = \sqrt{W^2 + H^2} + c_1$$

Thus the total distance travelled in one loop over the whole square is bounded by:

$$N_g[D_{GL} + D_{turn}] + c_2$$

which is equal to:

$$\frac{WH + Wc_1}{w_s(i)} + H + c_2 + o(\frac{1}{\sqrt{w_s(i)}})$$

■

Therefore, The length of the distance travelled by the vehicle following the Scanning Algorithm is bounded by two times the maximum distance travelled in any certain pass over the guidance lines.

Therefore for large i and from Lemma 1 and the fact that

$$w_s(i) = \frac{\sqrt[3]{(WH)^2}}{\rho^{1/3}(k_1 i)^{2/3}}$$

$$D_{SA} \leq \frac{WH + Wc_1}{w_s(i)} + O(\frac{1}{w_s(i)})$$

$$= \frac{(WH + Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}(k_1 i)^{2/3}$$

Therefore, the Scanning Algorithm visits a constant factor of the i targets in $O(i^{2/3})$.

IV. Iterated Level Algorithm

In this section, we will solve the following problem: Given n targets that are uniformly distributed in a rectangle R that had width W and height H; design an algorithm for a Dubins vehicle that will allow it to visit all of the n targets with a tour whose length is $O(n^{2/3})$.

Here, we will present the Iterated Level Algorithm that will have a performance within a constant factor of the established lower bound. To do that, we will create an $O(n^{2/3})$ algorithm (the Level Algorithm) that removes a constant factor of the targets while keeping remaining targets uniformly distributed. This algorithm can be iterated till the remaining number of targets is less than $\sqrt{n}$ and the remaining targets can be easily removed by a tour with length $O(\sqrt{n})$. We will first present the Level Algorithm and then prove the necessary bound on the produced tour's expected length. We will then prove that the distribution of the remaining targets at the end of the Level Algorithm will be bounded by a uniform distribution. The bounds on the Expected length of the Iterated Level Algorithm will follow.

A. Level Algorithm Description

The Level Algorithm has three steps:

1) Initialization step: Divide the whole area into *i*-rectangles (the ones introduced in the Scanning Algorithm). Number the targets in each of the *i*-rectangles and create "levels", which are collections of targets that have the same number (from different *i*-rectangles).
2) Pre-processing step: Remove all targets from levels that have a few targets (the exact number will be shown below).

3) Level processing step: Trace the non-empty levels top-down and at each level, applying the Scanning Algorithm to each of them.

Initialization:

Here, we will introduce another constant k_2 (analogous to k_1). We will study the performance of the algorithm for any k_1 and k_2, and in the end we will choose k_1 and k_2 to minimize the bound on the expected tour length.

In the initialization step, we first divide the area into the *i*-rectangles, each with width $w = \frac{k_2^{2/3}(WH)^{2/3}}{\rho^{1/3}(2n)^{2/3}}$ and length $l = \frac{2k_2^{1/3}\rho^{1/3}(WH)^{1/3}}{(2n)^{1/3}}$. We randomly number the targets in each *i*-rectangle. Targets from different *i*-rectangles that have the same number will form a "level". We think of each *i*-rectangle as a stack, and the targets in it as elements on top of each other in the stack, ordered by the numbers we gave them. It is useful to imagine the area A now as a histogram, where above every *i*-rectangle there is a stack that contains the targets in that *i*-rectangle. It is obvious that a lower level cannot contain less targets than any of those above it.

We will study the probability distribution of X, the random variable indicating the number of points in any one of the *i*-rectangles, as $n \to \infty$. X has a binomial distribution of n trials and probability of success $\frac{k_2}{n}$. This means that, as $n \to \infty$:

$$p.d.f.(X) \to Poisson(n\frac{k_2}{n})$$
$$= Poisson(k_2)$$

Therefore the probability that the number of points in a certain *i*-rectangle is i is given by:

$$P(X=i) = \frac{e^{-k_2}k_2^i}{i!}$$

Pre-processing:

The reason pre-processing is needed is that in the Scanning Algorithm, we assumed that the number of targets is large. If the area of the rectangles is large (their number is small), the bounds that we have proved will not hold. We therefore need to clear all levels that have a small number of targets (less than $n^{1/2}$).

Doing this is actually not difficult at all, since we know that the number of levels is less than $\ln(n)$ almost surely. This means that the total number of targets in all levels that have less than $n^{1/2}$ is less than $n^{1/2}\ln(n)$. Therefore, they can all be cleared using a tour that has a tour length that is $O(n^{1/2}\ln(n))$.

Level Processing:

The algorithm passes over each of the remaining levels once . When processing level i, we only consider the targets in that level and apply the Scanning Algorithm to them ,with $k_1 = ki!$. It is obvious that the setting satisfies all of the assumptions in the auxiliary problem. Applying the Scanning Algorithm guarantees that we take a constant factor of $c = e^{-\frac{4}{5ki!}}$ of the targets in level i. We also guarantee that if the number of targets in the i^{th} level is t_i, the length of the tour of the Scanning Algorithm in the i^{th} level is $O((i!t_i)^{2/3})$.

B. Level Algorithm Constant Factor Guarantee:

The Scanning Algorithm guarantees that the vehicle visits a constant factor of the targets at each level. Here, we will prove that applying the Scanning Algorithm to all of the levels will make the vehicle visit a constant factor of the total number of targets.

Let Y be the random variable representing the number of targets in an *i*-rectangle after the Level Algorithm has been applied. For Y to take a certain value j, the rectangle should have contained $i \geq j$ targets initially, and exactly $i-j$ targets were taken in the i passes that this rectangle was involved in. The scanning algorithm guarantees that the probability that a target is removed is $c(i) = e^{-\frac{4}{5ki!}}$. Thus the probability that Y takes a certain value j is given by:

$$P(Y=j) = \sum_{l\geq j} P(X=i)\frac{l!}{(l-j)!j!}(1-c(1))^j c(1)^{i-j}$$
$$\leq \sum_{l\geq j}\frac{e^{-k_2}k_2^l}{l!}\frac{l!}{(l-j)!j!}(1-c(1))^j c(1)^{l-j}$$
$$\leq \frac{[1-c(1)]^j k_2^j e^{-k_2}}{j!}\sum_{l\geq j}\frac{[c(1)k_2]^{l-j}}{(l-j)!}$$
$$= \frac{e^{-k_2+c(1)k_2}[k_2(1-c(1))]^j}{j!}$$

This is clearly dominated by a Poisson distribution with parameter $k_2(1-c(1))$. This also guarantees the remaining targets to be less than $1-e^{-\frac{4}{5k}}$ of the original targets.

C. Level Algorithm Expected Length

In each pass in the level algorithm, we are removing a constant factor of the t_j targets in the j^{th} level, and to do this, we are travelling:

$$\frac{(WH+Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}(k_1(j)t_j)^{2/3}$$

Therefore the total length of the Level algorithm (L_L)can be bounded by:

$$L_L = \frac{(WH+Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}\sum_{j\geq 1}E[(kj!t_j)^{2/3}]$$

By Jensen's inequality:

$$\leq \frac{(WH+Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}(k)^{2/3}\sum_{j\geq 1}(E[j!t_j])^{2/3}$$

Now, the expected number of points in a level is the number of *i*-rectangles that have stacks not lower than that level. Therefore,

$$E[j!t_j] = \frac{n}{k_2}P(X\geq j)$$
$$= \frac{j!n}{k_2}\sum_{i\geq j}\frac{e^{-k_2}k_2^i}{i!}$$

$$\leq \frac{j!nk_2^j}{k_2 j!} = nk_2^{j-1}$$

Therefore,

$$L_L \leq \frac{(WH + Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}(k)^{2/3}\sum_{j\geq 1}(\frac{nk_2^j}{k_2})^{2/3}$$

$$= \frac{(WH + Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}(kn)^{2/3}\sum_{j\geq 1}k_2^{\frac{2}{3}(j-1)}$$

$$= \frac{(WH + Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}(kn)^{2/3}S(k_2)$$

Where

$$S(k_2) = \sum_{j\geq 1}k_2^{\frac{2}{3}(j-1)}$$

We have the constraint $k_2 < 1$ to guarantee that $S(k_2)$ is finite.

D. Iterating the Level Algorithm

To iterate the level algorithm, it is necessary to bound the distribution of the remaining targets by a uniform distribution of αn targets where $\alpha < 1$.

The detailed proof of this bound is long, and will not be presented here. The fact that the distribution of the targets is uniform after the Level Algorithm is applied can be seen because of the following facts:

1) The distribution of remaining targets in level $i+1$ is ”better” than the distribution of a factor of the targets in level i.
2) If $k \leq \frac{1}{4k_2(1-e^{-k_2})}$, then the remaining targets in level 1 can be bounded by $[1-e^{-\frac{4}{5k}}]t_1$ independently distributed targets.

The first fact follows because the number of strips in level i is decreasing at a much slower rate than the number of targets in level i as a function of i. This is also what causes the Level Algorithm to remove $e^{-\frac{4}{5ki!}}$ of the targets in level i. This means that the remaining targets in level i will be more sparse than in level $i-1$.

The second fact follows from the geometry of strips in level 1 and the *i*-rectangles.

Therefore, with the assumption that $k \leq \frac{1}{4k_2(1-e^{-k_2})}$, the remaining targets after the level algorithm is applied can be bounded by uniformly distributed $[1-e^{-\frac{4}{5k}}]n$ targets.

E. Total Length:

If we define M by:

$$M = \frac{(WH + Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}$$

The total length of iterating the Level Algorithm will be bounded by:

$$L_t \leq MS(k_2)(\frac{k}{k_2})^{2/3}n^{2/3}\sum_{i=0}^{\infty}[1-e^{-\frac{4}{5k}}]^{\frac{2i}{3}}$$

$$= M\frac{S(k_2)}{k_2^{2/3}}\frac{k_1^{2/3}}{1-[1-e^{-\frac{4}{5k}}]^{\frac{2}{3}}}n^{2/3}$$

Now, k and k_2 can be chosen to minimize the constant factor. If $k = 2.13$ and $k_2 = 0.37$, we will have :

$$L_t \leq 4.9\frac{(WH + Wc_1)(\rho^{1/3})}{\sqrt[3]{(WH)^2}}n^{2/3}$$

V. Conclusions

In this paper, we studied the Stochastic DTSP, proved that the expected length of the optimal tour is $\Theta(n^{2/3})$. The algorithm we provided is novel and achieves a new bound. Another important property is that it can be generalized to when the area is not rectangular or the distribution is not uniform.

VI. Future Work

We have already generalized the results to the case of non-uniform distributions (with a finite number of discontinuities), and it will be reported elsewhere. For the non-uniform distribution scenario, we provided a lower bound that is $\Theta(n^{2/3})$ (it was different from the uniform case in that it depended on the probability distribution function of the targets), and we generalized the Iterated Level Algorithm so that it provides a tour that has an expected length that is $O(n^{2/3})$. These results will also help in the study of DTRP, where the DTRP is the travelling repairman problem introduced by Bertsimas and Van Ryzin in 1991 [7]. The DTSP and DTRP in the non-uniform case were never studied due to their complexity compared to the uniform case. The Generalized Iterated Level Algorithm will allow us to study that problem, and we hope to provide a lower bound and a constant factor algorithm for it.

References

[1] K. Savla, F. Bullo, and E. Frazzoli. ”On Traveling Salesperson Problems for Dubins’ vehicle: stochastic and dynamic environments.” *In Proc. IEEE Conf. on Decision and Control*, December 2005.

[2] K. Savla, E. Frazzoli, and F. Bullo. “On the point-to-point and traveling salesperson problems for Dubins’ vehicles.” *In Proc. of the American Control Conference*, 2005.

[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,* vol. 79, pp. 497-516,1957.

[4] D. Applegate, R. Bixby, V. Chv4atal, and W. Cook. “On the solution of traveling salesman problems.” *Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung*, (Berlin, Germany), pp. 645-656, Aug 1998. Proceedings of the International Congress of Mathematicians, Extra Volume ICM III.

[5] Erzberger, H., and Lee, H.Q.”Optimum Horizontal GuidanceTechniques for Aircraft.” *Journal of Aircraft,* Vol. 8 (No. 2), pp. 95-101, February, 1971.

[6] J. Bearwood, J. Halton, and J. Hammersly. “The shortest path through many points.” *Proceedings of the Cambridge Philosophy Society,* vol. 55, pp. 299-327, 1959.

[7] D.J. Bertsimas and G.J. Van Ryzin. “A stochastic and dynamic vehicle routing problem in the Euclidean plane.” *Operations Research*, vol. 39, pp. 601-615, 1991.

[8] S. Arora. “Nearly linear time approxation scheme for Euclidean TSP and other geometric problems.” *Proceedings of 38th IEEE Annual Symposium on Foundations of Computer Science,* (Miami Beach, FL,) pp. 554-563, Oct. 1997.

[9] Bui X. et. Al, “Shortest Path Synthesis for Dubins Non-holonomic Robot,” *IEEE* 1994