2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

The Stochastic Traveling Salesman Problem and Orienteering for Kinodynamic Vehicles

Aviv Adler* and Sertac Karaman†

***Abstract*—In the classic *Traveling Salesman Problem* (TSP), the objective is to find the shortest path that visits a set of target locations. This problem is embedded and essential in many planning problems that arise in robotics, particularly in the domains of exploration, monitoring, surveillance, and reconnaissance. In this paper we consider the *Stochastic TSP for Dynamical Systems*, where a vehicle with complex dynamics is tasked with visiting n random target locations. By borrowing techniques from the applied probability literature, which were used to study the related *stochastic Orienteering* problem (where the vehicle has to visit as many of the n points as possible with a path of fixed length), we simplify and extend the existing results for both the TSP and the stochastic Orienteering problems to cases where the target points can be picked up only when the vehicle is in a certain configuration (i.e. it is not enough simply to be on the target point). Specifically, we show that there is a special parameter γ of the dynamics of the vehicle, which governs the length of the TSP tour. The length of the shortest path will then be $\Theta(n^{(\gamma-1)/\gamma})$ with very high probability. For stochastic Orienteering, if the path must have length at most λ, the vehicle can pick up $\Theta(\lambda n^{1/\gamma})$ with very high probability. We also provide simple and efficient path planning algorithms which achieve these bounds, and are therefore within a constant factor of the length of the optimal path with very high probability.**

I. Introduction

The Traveling Salesman Problem (TSP) asks for the shortest path that visits a given set of locations. This problem attracted tremendous attention in the 1950s, as computers were utilized to solve transportation and logistics problems for the first time. Since then, the TSP has been considered a foundational problem in operations research, management science, mathematics and computer science.

It is known that solving this exactly is computationally challenging – specifically, it is NP-complete [10]. However, there are a number of successful algorithmic methods for this problem, including approximation algorithms and heuristics.

Although the TSP originated in operations research, it has found numerous applications in the context of robotics as well. Most notably, many robot motion planning and routing algorithms employ TSP algorithms at their core [13]. The applications of TSP in the robotics domain are far reaching, including persistent monitoring, surveillance, reconnaissance, exploration, among other important problems.

More recently, extensive research has been devoted to stochastic versions of the TSP, where the locations are distributed randomly according to some known statistics. Many problems in practice can be modeled this way, and it often allows for the derivation of better algorithms and tighter bounds than would be possible otherwise.

*Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT, Cambridge, MA, USA. adlera@mit.edu

† Laboratory of Information and Decision Systems (LIDS), MIT, Cambridge, MA, USA. sertac@mit.edu

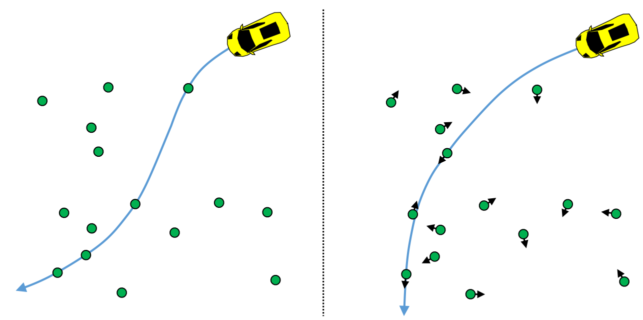

Fig. 1. An example of a dynamically-constrained Traveling Salesman Problem, with a Dubins car. **Left:** the standard variant in $\mathbb{R}^2$ (as solved by Savla et al. [15]). **Right:** the variant with each target given a direction, in which it must be visited; the solution to this problem was previously unknown, but is now covered as an instance of our main theorem.

However, in most cases in robotics, the vehicles are subject to non-trivial differential constraints which have a substantial impact on the length of the shortest tour. This means that new algorithms must be used to deal with the TSP in these instances [14]. Furthermore, understanding the impact of the dynamics on the length of the tour would allow a system designer to pick the best vehicle for the task at hand.

The importance of this problem has not gone unnoticed. The stochastic TSP for unconstrained Euclidean paths was solved by Beardwood et al. in 1959 [3], and in 1991, its application to vehicle routing was recognized [6]. In the past decade, tight bounds on the stochastic TSP were found for a number of vehicles, notably the Dubins car, Reeds-Shepp car, differential-drive vehicles, and double integrators (which are used to model quadcopters) [9, 15, 16, 17, 18]. The stochastic TSP for more general dynamical systems with points in 2 dimensions was considered by Itani et al. [12], and a very general model of vehicle dynamics was studied by Itani in his Ph.D. dissertation [11], extending these results to most cases where the target points are distributed in Euclidean space.

In addition to the above, there was progress being made on a very similar stochastic version of the *Orienteering* problem (itself a very important problem in operations research, with applications to robotics [2, 7, 19]), notably the 2005 paper of Arias-Castro et al. [2] which showed a general blueprint for solving the problem with various dynamical systems.

In this paper, our main contribution is to show how the methods developed in these two separate strands of research can be unified to produce a more general set of results for both the TSP and Orienteering. Specifically, we show analogous results even when the target points are not distributed in Euclidean space, but have to be picked up in a particular way by the vehicle (see Figure 1 for an example). We also show how the blueprint developed by Arias-Castro et al. can be automatically applied to all vehicles (rather than needing custom tweaking for each vehicle), producing greatly simplified proofs and algorithms for these results (for both the TSP and Orienteering).

II. Preliminaries

A. The Target and Configuration Spaces

We first discuss the points collected by the vehicle. We assume that they are distributed in a d-dimensional manifold; this means that we have a well-defined notion of volume in this space. By not limiting ourselves to the flat Euclidean space, we can model situations where the vehicle must collect points in a particular way. We refer to the space in which the points are distributed as the *workspace* and denote it $\mathcal{X}$. The volume of a set $Z \subseteq \mathcal{X}$ is denoted by $\mathrm{vol}_{\mathcal{X}}(Z)$, and if $\mathcal{X} = \mathbb{R}^d$ we denote the volume by $\mathrm{vol}_d(Z)$.

We model the constraints on our vehicle by considering the notion of the *configuration space* $\mathcal{C}$. This includes the vehicle's location in the workspace, but includes all the information needed to determine its possible future trajectories; for example, in the case of the Dubins car in $\mathbb{R}^2$, the vehicle's location in $\mathcal{C}$ is its location in $\mathbb{R}^2$ and its orientation. We note that by definition, each configuration is located at a point in the workspace, so we can project subsets of $\mathcal{C}$ onto $\mathcal{X}$. The projection of a set $S \subseteq \mathcal{C}$ onto $\mathcal{X}$ is denoted $[S]_{\mathcal{X}}$. Similarly, for a set $Z \subseteq \mathcal{X}$, we denote by $[Z]^{\mathcal{C}}$ the set of all configurations $x \in \mathcal{C}$ whose projection onto $\mathcal{X}$ is in Z, i.e.

$$[Z]^{\mathcal{C}} \triangleq \{x \in \mathcal{C} : [x]_{\mathcal{X}} \in Z\}.$$

The length of the path is most generally defined as being dependent on the trajectory in $\mathcal{C}$; this allows us to include in our analysis differential-drive vehicles and other systems which might take time to change their configuration while remaining in the same target location. In particular, we will be making assumptions which assure that $\mathcal{C}$ is an algebraic subset of a manifold (of possibly higher dimension than $\mathcal{X}$); the length of a path is defined as its length in $\mathcal{C}$.

B. The Dynamical System Π and the Target Points

We present our notation for trajectories through $\mathcal{C}$ and $\mathcal{X}$:

- Π denotes both the set of constraints on the vehicle, and the set of trajectories through $\mathcal{C}$ which satisfy those constraints (and which we call *valid*);
- $\hat{\Pi}$ denotes the set of trajectories through $\mathcal{X}$ which correspond to a valid trajectory, i.e. $\hat{\pi} \in \hat{\Pi}$ iff there is some $\pi \in \Pi$ such that $\hat{\pi} = [\pi]_{\mathcal{X}}$;
- for any trajectory $\pi \in \Pi$, we denote its length by $\ell[\pi]$, meaning π is a function from $[0, \ell[\pi]]$ to $\mathcal{C}$;
- for any $\lambda > 0$, we denote the set of valid trajectories with length $\leq \lambda$ by Π_λ, and the set of trajectories in $\mathcal{X}$ corresponding to this by $\hat{\Pi}_\lambda$;
- we say π *goes from x to y* if $\pi(0) = x$ and $\pi(\ell[\pi]) = y$.

We now define the properties of *memorylessness* and *translation-invariance* for constraints on trajectories.

Definition 1 (Memoryless Constraints)**.** *A constraint set* Π *is* memoryless *if for any trajectories $\pi_1, \pi_2 \in \Pi$ such that the endpoint of π_1 is the starting point of π_2, concatenating them produces another trajectory in* Π.

This means that the possible future trajectories for the vehicle only depends on its present configuration.

Definition 2 (Translation Invariance over $\mathcal{X}$)**.** Π *is* translation-invariant over $\mathcal{X}$ *if any valid trajectory can be translated anywhere in $\mathcal{X}$ and remain valid (since $\mathcal{X}$ is a manifold, this is well-defined).*

We now discuss the target points. We denote the n target points as $X_1, X_2, \ldots, X_n$, which are independently and identically distributed over $\mathcal{X}$ according to distribution f.

Definition 3 (Support and Configuration Support of f)**.** *We denote the support of f by $\mathcal{X}_f \subseteq \mathcal{X}$. Then, the* configuration support of f *is defined as $\mathcal{C}_f \triangleq [\mathcal{X}_f]^{\mathcal{C}}$, i.e. the configurations which project onto the support of f.*

C. Assumptions

We make the following assumptions:

1) the set of constraints Π is memoryless, affine in control, real analytic, and translation-invariant over $\mathcal{X}$;
2) the set $\mathcal{X}_f$ is full-dimensional and bounded;
3) the set $\mathcal{C}_f$ is Π-*bounded*, i.e. there is some $M \in \mathbb{R}$ such that for any two configurations $x, y \in \mathcal{C}_f$, there is a path in Π of length at most M from x to y;
4) f has bounded density.

These assumptions are very general, and broadly define the kinds of systems which describe vehicles.

D. ϵ-Time Reachability and Deviation Sets

We now define the notion of the *ϵ-time reachable set*. Intuitively, for any configuration $x \in \mathcal{C}$ of the vehicle, it is the area the vehicle can reach from x with a path of at most ϵ length. We define two notions for this, one in $\mathcal{C}$ and the other in $\mathcal{X}$. See Figure 2 for an illustration. Formally:

Definition 4 (ϵ-Time Reachability)**.** *Let $x \in \mathcal{C}$ and $\epsilon > 0$; then the* ϵ-time reachable set $R^{\Pi}_{\epsilon}(x)$ *of x at length ϵ is*

$$R^{\Pi}_{\epsilon}(x) \triangleq \{y \in \mathcal{C} : \exists \pi \in \Pi_\epsilon \text{ which goes from } x \text{ to } y\}.$$

Note that this is a subset of $\mathcal{C}$.

We define the projected ϵ-time reachable set $\hat{R}^{\Pi}_{\epsilon}(x)$ *as*

$$\hat{R}^{\Pi}_{\epsilon}(x) \triangleq \left[R^{\Pi}_{\epsilon}(x)\right]_{\mathcal{X}}$$

i.e. the projection of $R^{\Pi}_{\epsilon}(x)$ onto the workspace.

We also define the notion of the *ϵ-time deviation set*:

Definition 5 (ϵ-Time Deviation)**.** *Let $\pi \in \Pi$, $t \in \big[0, \ell[\pi]\big]$, and $\epsilon > 0$ (such that $t+\epsilon \in \big[0, \ell[\pi]\big]$ as well). We then define the ϵ-time deviation set of π at t as*

$$D^{\Pi}_{\epsilon}(t, \pi) \triangleq \{x \in \mathcal{C} : x \in R^{\Pi}_{\epsilon}(\pi(t)) \text{ and } \pi(t+\epsilon) \in R^{\Pi}_{\epsilon}(x)\}.$$

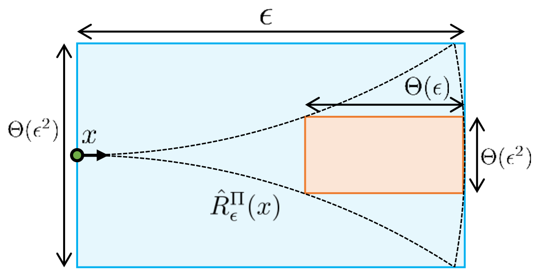

Fig. 2. An example of a (projected) reachability set for a Dubins car from configuration x (represented by the point and the heading arrow); the two boxes (the blue box containing the reachability set and the orange box contained by it) illustrate the Ball-Box Theorem (see Lemma 1) as applied to Dubins cars.

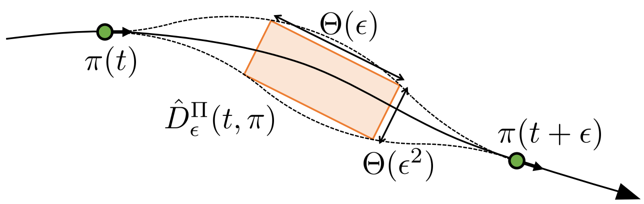

Fig. 3. An example of a (projected) deviation set for a Dubins car for configuration; the box contained by the deviation set indicates that the path is *flexible* at $\pi(t)$, as per the Ball-Box Theorem (see Lemma 2).

Similar to ϵ-time reachability, we define the projected ϵ-time deviation set *as*

$$\hat{D}^{\Pi}_{\epsilon}(t,\pi) \triangleq \left[D^{\Pi}_{\epsilon}(t,\pi)\right]_{\mathcal{X}}.$$

Intuitively, this is the set of configurations in $\mathcal{C}$ (or locations in $\mathcal{X}$) we can visit during a brief deviation from π, starting at $\pi(t)$ and returning at $\pi(t+\epsilon)$; see Figure 3.

E. The Ball-Box Theorem and Small-Time Constraint Factor

We present some lemmas, derived from the Ball-Box Theorem [4, 5]. Due to space constraints, we omit the proofs.

Lemma 1 (Volume of ϵ-Time Reachable Sets)**.** *For any dynamic system satisfying our assumptions, for any $x \in \mathcal{C}$ there is a parameter $\gamma(x) \in \mathbb{N}$ such that,*

$$\lim_{\epsilon\to 0} \mathrm{vol}_{\mathcal{X}}\left(\hat{R}^{\Pi}_{\epsilon}(x)\right) = \Theta(\epsilon^{\gamma(x)})$$

i.e. the volume of $\hat{R}^{\Pi}_{\epsilon}(x)$ in $\mathcal{X}$ is proportional to $\epsilon^{\gamma(x)}$. In particular, it both contains a box of volume $\Theta(\epsilon^{\gamma(x)})$ and is contained by a box of volume $\Theta(\epsilon^{\gamma(x)})$ (though of course these boxes' volumes have different constant parameters), and in fact the boxes have length $\Theta(\epsilon)$ and $(d-1)$-dimensional base volume $\Theta(\epsilon^{\gamma(x)-1})$.

Since we want the vehicle to collect as many points as possible, and the points get arbitrarily dense as $n \to \infty$, we are interested in the configuration x at each $z \in \mathcal{X}$ for which $\gamma(x)$ is *minimized* (as this allows the vehicle to explore the largest possible set in ϵ time starting at $z \in \mathcal{X}$).

Definition 6 (Small-Time Constraint Factor)**.** *We define the* small-time constraint factor *of Π as*

$$\gamma \triangleq \min_{x\in[z]^{\mathcal{C}}} \gamma(x)$$

for any $z \in \mathcal{X}$; since Π is translation-invariant over $\mathcal{X}$, this value is the same no matter what $z \in \mathcal{X}$ is chosen.

We refer to this as the *small-time constraint factor* because it describes the degree to which the motion of the vehicle through $\mathcal{X}$ is constrained over very small scales. $\gamma \geq d$ since trajectories must be continuous (and hence can't travel outside a ϵ-radius ball in $\mathcal{X}$ within $\leq \epsilon$ length). $\gamma(x)$ can be computed efficiently for any x via the Ball-Box Theorem [4, 5]; we can then search $[z]^{\mathcal{C}}$ for any $z \in \mathcal{X}$ to compute γ.

Lemma 2 (Volume of ϵ-Time Deviation Sets)**.** *For any point $x \in \mathcal{C}$, there exists some differentiable path π which passes through $x = \pi(t)$ (for some $t \in [0, \ell[\pi])$) and for which*

$$\lim_{\epsilon\to 0} \mathrm{vol}_{\mathcal{X}}\left(\hat{D}^{\Pi}_{\epsilon}(t,\pi)\right) = \Theta(\epsilon^{\gamma}).$$

Additionally, the set contains a box of the proportions described in the Ball-Box Theorem.

We call such a path π *flexible at* $x = \pi(t)$; we are interested in paths which are flexible at all times:

Definition 7 (c-Flexible Paths)**.** *A path π is c-flexible for constant $c > 0$ if, for every $t \in [0, \ell(\pi))$,*

$$\lim_{\epsilon\to 0} \frac{\mathrm{vol}_{\mathcal{X}}\left(\hat{D}^{\Pi}_{\epsilon}(t,\pi)\right)}{\epsilon^{\gamma}} \geq c.$$

Intuitively, a c-flexible path is one from which the vehicle may take small deviations from any point and then return. A path which is c-flexible for some nonzero constant c is referred to as *flexible*.

Definition 8 (Uniform c-Flexibility)**.** *A collection of paths Φ is* uniformly c-flexible *if every path $\phi \in \Phi$ is c-flexible.*

A collection of paths which is uniformly c-flexible for some nonzero c is referred to as *uniformly flexible*.

III. Main Results

We now formally define our problems and state our results.

A. The Traveling Salesman Problem and Orienteering

We define the following function, which takes the constraints Π and a set of points $X_1, X_2, \ldots, X_n \in \mathcal{X}$ and returns the length of the shortest path satisfying Π which collects all the points. Our goal for the remainder of the paper will be to determine (with high probability) the behavior of this function when the inputs $X_1, X_2, \ldots, X_n$ are randomly distributed (independently and identically).

Definition 9 (The Traveling Salesman Problem)**.**

$$\mathrm{TSP}_{\Pi}(X_1, X_2, \ldots, X_n) \triangleq$$
$$\inf(\ell[\pi] : \pi \in \Pi \textit{ such that } X_i \in \hat{\pi} \textit{ for all } i).$$

To analyze this, we introduce the objective ORNT_{Π} of the Orienteering problem. This takes the targets $X_1, X_2, \ldots, X_n$, as well as a length parameter $\lambda > 0$, and returns the maximum number of targets in a valid length-λ path.

Definition 10 (The Orienteering Problem)**.**

$$\mathrm{ORNT}_{\Pi}(X_1, X_2, \ldots, X_n; \lambda) \triangleq$$
$$\max(|\{X_i\}_{i=1}^{n} \cap \hat{\pi}| : \pi \in \Pi \textit{ and } \ell[\pi] \leq \lambda).$$

Our main results concern the properties of these functions.

B. Our Results

In our terminology, an event occurs "with very high probability" means that the probability that it does not happen goes to 0 exponentially fast as $n \to \infty$. We then present the following two results:

Theorem 1 (Bounds on TSP_Π). *Let Π be a set of constraints, and f be a probability distribution over $\mathcal{X}$, satisfying the assumptions given in Section II-C. Then, if $X_1, X_2, \ldots, X_n \in \mathcal{X}$ are identically and independently distributed according to f,*

$$\mathrm{TSP}_\Pi(X_1, X_2, \ldots, X_n) = \Theta\big(n^{(\gamma-1)/\gamma}\big)$$

with very high probability.

Theorem 2 (Bounds on ORNT_Π). *Let Π be a set of constraints, and f be a probability distribution over $\mathcal{X}$, satisfying the assumptions given in Section II-C, and let $\lambda > 0$. Then, if $X_1, X_2, \ldots, X_n \in \mathcal{X}$ are identically and independently distributed according to f,*

$$\mathrm{ORNT}_\Pi(X_1, X_2, \ldots, X_n; \lambda) = \Theta\big(\lambda n^{1/\gamma}\big)$$

with very high probability.

We will prove these theorems by showing upper bounds for both of them and that the upper bound for Theorem 2 implies the lower bound for Theorem 1 (and vice versa).

C. Simplifications

We make the following simplifications:

1) $\mathcal{X}$ is $\mathbb{R}^d$ (rather than a d-dimensional manifold);
2) f is uniform over $[0, 1]^d$;
3) there exists some direction v in $\mathbb{R}^d$ such that:
    a) every path spanning $[0, 1]^d$ parallel to v is valid;
    b) the collection of such paths is uniformly flexible.

At first glance, we have just made the problem vastly less general. However, if we prove Theorems 1 and 2 with these simplifications, we can generalize the results to the original version. Due to space constraints, we sketch this proof.

We generalize from these simplifications by breaking the support of f into hypercubic pieces. Since $\mathcal{X}$ is a manifold, once we have made the pieces small enough, each piece behaves as if it is in $\mathbb{R}^d$, making simplification 1 approximately true (note that this allows generalization to arbitrary f, not just piece-wise uniform f). Furthermore, at sufficient granularity, f is roughly uniform in each piece, making simplification 2 approximately true as well. The slight deviation from perfect uniformity on each piece will increase the length of the tour by only a constant factor. Finally, any constraints satisfying the assumptions in Section II-C must permit uniformly flexible smooth paths, which on small scales are approximately straight, making 3(a) approximately true. Finally, we note that by translation invariance every point in $z \in \mathcal{X}$ has a configuration $x \in [z]^{\mathcal{C}}$ such that $\gamma(x) = \gamma$; there then exists some $\delta > 0$ such that a length-δ flexible path passes through z. By ensuring that our pieces here have width at most δ, we can ensure that simplification 3(b) is true.

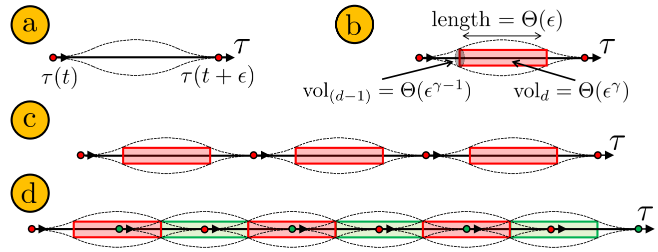

Fig. 4. A visual representation of the tiling process of a 'row' (of which there are $\Theta(n^{(\gamma-1)/\gamma})$) at resolution ϵ; arrows included to indicate that the waypoints represent *configurations*. (a) Two points along straight flexible trajectory τ at distance ϵ, and their deviation set. (b) The Ball-Box Theorem implies that the deviation set contains a box of the dimensions shown. (c) Deviation sets are strung together along trajectory τ at distance ϵ. (d) With a constant number of passes (one depicted in red, the other in green) through trajectory τ, these boxes cover a whole 'row' around τ.

We note that the scale of the pieces does not depend on n, only on f; thus, there are a fixed number of pieces. For the upper bound, we find a path of the appropriate length through all the points in each piece, and by the boundedness assumption in Section II-C, we can string together the paths for each piece for a fixed (with respect to n) cost.

IV. Upper Bound of TSP_Π

Proposition 1. *Let Π be a set of constraints satisfying the assumptions given in Section II-C over $\mathbb{R}^d$. Then, if $X_1, X_2, \ldots, X_n$ are identically and independently distributed uniformly on $[0, 1]^d$, there is a constant α_1 s.t.*

$$\mathrm{TSP}_\Pi(X_1, X_2, \ldots, X_n) \leq \alpha_1 n^{(\gamma-1)/\gamma}$$

with very high probability.

We show this by using an algorithm similar to the Recursive Bead-Tiling Algorithm introduced by Savla et al for Dubins cars [16]. The strategy is centered around two steps:

1) divide $[0, 1]^d$ into cells of size $1/n$ such that there is a short path visiting a target at each (nonempty) cell;
2) run through the collection path repeatedly, periodically merging the cells as they start to empty out in order to avoid wasting time traversing through empty cells (so that future collection paths collect one point from each of the new, larger cells).

A. Tiling the Workspace

Due to space constraints, we will sketch the tiling process, which is based on the concept of using a fixed set of flexible paths as a track; the vehicle follows the track, making small deviations from certain configurations in order to collect points. We refer to the points from which it makes its devations as *waypoints*. Consider a flexible trajectory τ through $[0, 1]^d$, and consider configurations $\tau(t), \tau(t+\epsilon) \in \mathcal{C}$ on this path.

The tiling process works in 'rows' which are each centered around a straight flexible trajectory τ; the process for a given row τ is depicted in Figure 4. Consider waypoints $\tau(t), \tau(t+\epsilon), \tau(t+2\epsilon), \ldots,$ and consider the ϵ-deviation sets at each of these configurations. Note that the vehicle can start at $\tau(t)$, visit any chosen point in $\hat{D}^{\Pi}_{\epsilon}(\tau, t)$ and make it to configuration $\tau(t+\epsilon)$, visit any chosen point in $\hat{D}^{\Pi}_{\epsilon}(\tau, t+\epsilon)$, and so forth - this means that in time at most twice that it takes to traverse τ, the vehicle can visit any set of target points consisting of at most one point in each ϵ-deviation set along τ. By the Ball-Box Theorem, these sets contain boxes of length $\Theta(\epsilon)$ and total volume $\Theta(\epsilon^{\gamma})$; thus, the $(d-1)$-dimensional volume of its base is $\Theta(\epsilon^{\gamma-1})$. Since we want each of these cells to have volume $1/n$, we set $\epsilon = \Theta(n^{-1/\gamma})$, so the base has $(d-1)$-dimensional volume $\Theta(n^{-(\gamma-1)/\gamma})$. Although there are places around τ which these boxes do not cover, since their length is $\Theta(\epsilon)$, we can just repeat this a constant number of times with the waypoints shifted each time to cover a whole 'row' around τ. This row has base volume $\Theta(n^{-(\gamma-1)/\gamma})$ – meaning that to cover all of $[0,1]^d$ we need $\Theta(n^{(\gamma-1)/\gamma})$ rows.

B. Collecting a Point from Each Cell

The vehicle can then traverse each row a constant number of times – which takes a constant amount of time per since each row spans $[0,1]^d$, and by assumption (3) from Section II-C traversing from the end of one row to the beginning of another takes a path of length at most M – and collect one point from each cell. Since there are $\Theta(n^{(\gamma-1)/\gamma})$ rows in total, this means that we can visit one point in each cell with a path of length $\Theta(n^{(\gamma-1)/\gamma})$. This effectively removes a point from each cell; we call this a *collection* step.

It would then be convenient if we could simply repeat this process (each reset – i.e. going from the end of some row to the beginning of another row – takes a path of length at most M, and so increases the length by at most a constant factor) until all the cells are empty (i.e. all the points are visited). However, this would mean that we would have traverse the track once for each target point in the most populated cell. Since the points are distributed uniformly and independently at random into the n cells of volume $1/n$, the most populated cell contains (with high probability) $\Theta(\frac{\log n}{\log\log n})$ points, giving us only a $(\frac{\log n}{\log\log n})$-approximation of the claimed lower bound.

C. Merging Cells

Instead, we make the following observation: as we perform collections, the cells with fewer points empty out. Any time spent visiting an empty cell is wasted. However, if we *merge* many cells together into a larger cell (with a larger resolution), even if most of the merged cells are empty, the larger one may not be empty (see Figure 5), preventing waste.

In particular, consider a cell generated by the method depicted in Figure 4, but with the scale 2ϵ (again, as in Figure 5). In this case, cell will have length approximately double the length of a cell at scale 2ϵ, but its base will be $2^{\gamma-1}$ times the size. Thus, there will be only one row at this scale per $2^{\gamma-1}$ rows at the original ϵ scale. Since the cost of removing a point from each cell is proportional to the number of rows, the cost of the collection process on cells at scale 2ϵ is $2^{-(\gamma-1)}$ the cost of the collection process at scale ϵ. Of course, there are fewer cells so the collection is coarser in addition to being cheaper; in particular, for every 2^{γ} cells originally, there is only one new cell.

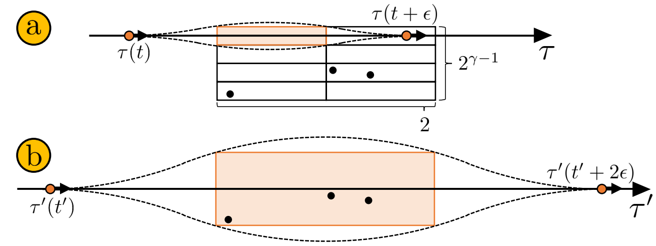

Fig. 5. An illustration of the cell-merging process, as applied to Dubins cars. (a) 2^{γ} cells (since this is a Dubins car, $\gamma = 3$); the top-left cell is shown as being created by an ϵ-deviation set from a path τ. The smaller dots are target points which still need to be visited. (b) The new, merged cell is a 2ϵ-deviation set from a path τ'.

We then note that we don't have to stop at one merge step – we can keep merging until we have one 'cell' consisting of the whole region $[0,1]^d$. At this point, the cost of a collection is a constant, but collects only one (arbitrary) point.

Now that we have described the process of cell merging, it remains to be shown how we can use it effectively to reduce the length of the tour created by the algorithm. We describe the merging strategy we use, but due to lack of space we omit the proof of efficiency. In brief, we can use the effects of merging – in particular, how the length of a pass is affected – to describe an abstract discrete problem whose solution gives our algorithm..

We use the following strategy:

1) collect once with the original cells (*initialization*), then merge;
2) collect 2^{γ} times before each subsequent merge, until we have only one 'cell' over $[0,1]^d$ (*main steps*);
3) visit all remaining points in arbitrary order (*cleanup*).

We now claim that this process requires a trajectory of length at most $\Theta(n^{(\gamma-1)/\gamma})$ with very high probability. To prove this, however, requires a fairly lengthy (though beautiful) proof, which we don't have the space here to give. However, interested readers should the read proof given in the appendix of our full paper [1].

Remark: The recipe for collecting and merging does *not* depend on the distribution of points, relying instead on the predictability of distributions of many i.i.d. points. This allows our algorithm to be computationally efficient, since it needs no computations to determine when to merge. See [1] for a discussion of the complexity of the algorithm.

D. Synthesizing the TSP Algorithm

From this we can produce the full TSP algorithm. It follows the steps outlined above:

1) Tile $[0,1]^d$ with cells of volume $1/n$ as described in Section IV-A.
2) Follow the merging strategy given in Section IV-C:
    a) each 'collect' travels through all the cells, visiting a previously unvisited point in each cell (provided the cell contains such a point);
    b) each 'merge' in the balls-and-buckets problem corresponds to merging cells in groups of 2^{γ}, to

produce a tiling as described in Section IV-A but at twice the scale.

Thus, by the above, we have proved Proposition 1 and given an algorithm which, with very high probability, produces a path of length $\Theta(n^{\frac{\gamma-1}{\gamma}})$ that visits all n targets.

V. Upper Bound of ORNT_Π

Proposition 2. *Let Π be a set of constraints, and f be a probability distribution over $\mathcal{X}$, satisfying the assumptions given in Section II-C, and let $\lambda > 0$. Then, if $X_1, X_2, \ldots, X_n \in \mathcal{X}$ are identically and independently distributed according to f, there exists a constant α_2 such that*

$$\mathrm{ORNT}_\Pi(X_1, X_2, \ldots, X_n; \lambda) \leq \alpha_2 \lambda n^{1/\gamma}$$

with very high probability.

We prove this by using a generalized version of the powerful probabilistic argument introduced by Arias-Castro et al [2] (who refer to it as *Connect-the-Dots*). It begins with a discretization of $\mathcal{C}_f$ into cells with the following properties:

- the projection of each cell onto $\mathcal{X}$ has volume $\Theta(1/n)$;
- there are a polynomially-bounded (in n) number of 'starter' cells, which cover the whole space;
- each cell has (at most) a bounded number b of 'successor' cells, such that any valid path of length λ can be covered by a sequence of $\Theta(\lambda n^{(\gamma-1)/\gamma})$ cells, beginning with a 'starter' cell and choosing a successor of the current cell at each subsequent step.

Once this discretization has been given, a probability argument from the Chernoff and Union bounds gives the result.

A. Implicitly Discretizing the Workspace

Our discretization will be *implicit*, in the sense that we do not directly define the cells. Instead, we start with a set of cells, and define new cells as successors to previously-generated cells. We define $\mathcal{C}_{[0,1]^d} \triangleq \left[[0,1]^d\right]^{\mathcal{C}}$; this corresponds to the notion of $\mathcal{C}_f$ as defined in Section II-A. We now give some crucial lemmas; due to space constraints, we sketch the proofs.

Lemma 3 (Polynomial Cover of $\mathcal{C}_{[0,1]^d}$)**.** *There exists a polynomial p such that for any $\epsilon > 0$, there exists a collection of $p(1/\epsilon)$ configurations $y_1^*, y_2^*, \ldots, y_{p(1/\epsilon)}^* \in \mathcal{C}$ such that*

$$\mathcal{C}_{[0,1]^d} \subseteq \bigcup_{i=1}^{p(1/\epsilon)} R_\epsilon^\Pi(y_i^*).$$

We refer to this set as Y_ϵ^*.

In short, as $\epsilon \to 0$, the set $\mathcal{C}_{[0,1]^d}$ can be covered by a collection of polynomially (in $1/\epsilon$) many ϵ-reachability sets.

Proof. Consider the collection of straight, parallel flexible trajectories whose existence we assumed in Section III-C; let τ be any of these trajectories.

Then, by the Ball-Box Theorem, $R_\epsilon^\Pi(\tau(t))$ contains a rectangular prism of volume $\Theta(\epsilon^\gamma)$; we note that in each direction, the length of the prism is $\Omega(\epsilon^\gamma)$ (it can't be shorter or else the volume is too small; in fact, this is a very loose bound). Thus, the reachability set contains a sphere of radius $\Omega(\epsilon^\gamma)$, which itself contains a hypercube of edge length $\Omega(\epsilon^\gamma)$. This hypercube can be oriented in any direction, so we orient it so that its sides are parallel to those of $[0,1]^d$. □

Lemma 4 (Fixed-Size Cover of $R_{2\epsilon}^\Pi(x)$)**.** *There exists a positive integer b such that, for any $\epsilon > 0$ and $x \in \mathcal{C}$, there exists a set of b configurations $y_1, y_2, \ldots, y_b \in \mathcal{C}$ such that*

$$R_{2\epsilon}^\Pi(x) \subseteq \bigcup_{i=1}^{b} R_\epsilon^\Pi(y_i).$$

In other words, any (2ϵ)-reachable set can be covered by a fixed number of ϵ-reachable sets. For a point $x \in \mathcal{C}$ and $\epsilon > 0$, we refer to $Y_\epsilon(x) \triangleq \{y_1, \ldots, y_b\}$ as the *menu* of x.

Proof. We prove this with Lemma 1; we note that the Ball-Box Theorem still applies when $\mathcal{X} = \mathcal{C}$ (i.e. the vehicle must be in a particular configuration to collect a target), though it produces a different exponent which we denote as ξ. We note that by Lemma 1, $R_{2\epsilon}^\Pi(x)$ then is contained by a box of volume $\Theta(\epsilon^\xi)$, while for any configuration y, $R_\epsilon^\Pi(y)$ contains a box of volume $\Theta(\epsilon^\xi)$. Thus, only a constant number of the latter boxes is needed to cover the former; but the former contains $R_{2\epsilon}^\Pi(x)$ and the latter are all contained by $R_\epsilon^\Pi(y)$ for some y, so $R_{2\epsilon}^\Pi(x)$ (for any $x \in [0,1]^d$) can be covered by the union of $R_\epsilon^\Pi(y)$ for boundedly many $y \in \mathcal{C}$. □

B. A Discrete Representation Scheme for Valid Paths

We now want a way to discretely represent any $\pi \in \Pi_\lambda$. First, we note that we only need to consider paths π which begin at a configuration $\pi(0)$ in $\mathcal{C}_{[0,1]^d}$. This is because if $\pi(0) \notin \mathcal{C}_{[0,1]^d}$, we can simply remove the beginning of π and get a shorter path collecting the same targets.

The basic idea is to represent π by a sequence of configurations $\psi = \{\psi_0, \psi_1, \ldots, \psi_{\lfloor \lambda/\epsilon \rfloor}\}$ such that:

- $\pi(i\epsilon) \in R_\epsilon^\Pi(\psi_i)$ for each $i = 0, 1, \ldots, \lfloor \lambda/\epsilon \rfloor$;
- $\psi_0 \in Y_\epsilon^*$;
- $\psi_{i+1} \in Y_\epsilon(\psi_i)$ for each $i = 0, 1, \ldots, \lfloor \lambda/\epsilon \rfloor - 1$.

In other words, we first select a configuration ψ_0 from the covering set Y_ϵ^* of $\mathcal{C}_{[0,1]^d}$ such that the starting point of π is ϵ-reachable from ψ_0. We then iteratively select ψ_{i+1} from the menu $Y_\epsilon(\psi_i)$, keeping the invariant that $\pi(i\epsilon)$ is ϵ-reachable from ψ_i. We now need to show that this representation exists.

By Lemma 3, since $\pi(0) \in \mathcal{C}_{[0,1]^d}$, there is some $\psi_0 \in Y_\epsilon^*$ such that $\pi(0) \in R_\epsilon^\Pi(\psi_0)$. We now inductively show that if $\pi(i\epsilon) \in R_\epsilon^\Pi(\psi_i)$, we can select some $\psi_{i+1} \in Y_\epsilon(\psi_i)$ such that $\pi((i+1)\epsilon) \in R_\epsilon^\Pi(\psi_{i+1})$.

Suppose (by the inductive assumption) that $\pi(i\epsilon) \in R_\epsilon^\Pi(\psi_i)$. Then, since $\pi((i+1)\epsilon)$ is by definition ϵ-reachable from $\pi(i\epsilon)$, we know that $\pi((i+1)\epsilon) \in R_{2\epsilon}^\Pi(\psi_i)$. By Lemma 4, any configuration in $R_{2\epsilon}^\Pi(\psi_i)$ is covered by an ϵ-reachability set from some point in $Y_\epsilon(\psi_i)$; we can then set ψ_{i+1} to be this point. See Figure 6 for illustration.

Thus, we can represent any $\pi \in \Pi_\lambda$ by such a sequence of $\lfloor \lambda/\epsilon \rfloor + 1 = \Theta(\lambda n^{1/\gamma})$ configurations. Furthermore:

$$\pi \subset S(\psi) \triangleq \bigcup_{i=0}^{\lfloor \lambda/\epsilon \rfloor} R_{2\epsilon}^\Pi(\psi_i), \text{ so}$$

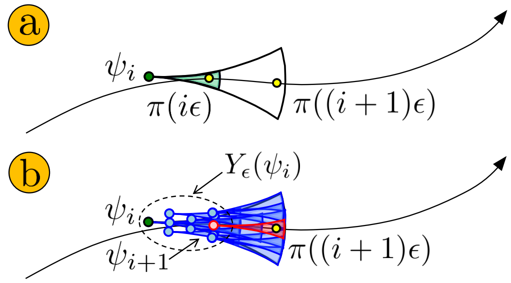

Fig. 6. An illustration of the iterative generation of the discrete representation ψ of path π. (a) The rule that $\pi(i\epsilon) \in R^{\Pi}_{\epsilon}(\psi_i)$ (shown in green) applies, so we know that $\pi((i+1)\epsilon) \in R^{\Pi}_{2\epsilon}(\psi_i)$. (b) By Lemma 4, $R^{\Pi}_{2\epsilon}(\psi_i)$ can be covered by a set of ϵ-reachability sets (depicted in blue); thus, one of these sets (depicted in red) contains $\pi((i+1)\epsilon)$. Its starting point is chosen as ψ_{i+1}, so $\pi((i+1)\epsilon) \in R^{\Pi}_{\epsilon}(\psi_{i+1})$, allowing the process to continue.

$$\hat{\pi} \subset \hat{S}(\psi) \triangleq [S(\psi)]_{\mathcal{X}} = \bigcup_{i=0}^{\lfloor \lambda/\epsilon \rfloor} \hat{R}^{\Pi}_{2\epsilon}(\psi_i).$$

This is because any point on π is at most ϵ away from some $\pi(i\epsilon)$, which is itself ϵ-reachable from ψ_i by definition.

However, there exist only finitely many representations. ψ_0 is chosen from Y^*_ϵ, i.e. there are $p(1/\epsilon)$ possible initial configurations of the sequence; from then on, each configuration is chosen from the menu of the previous one, which has a cardinality of b. Since $\epsilon = \Theta(n^{-1/\gamma})$, there is some polynommial q such that $p(1/\epsilon) \leq q(n)$. Thus, with $\leq q(n)$ initial choices, and b choices at each of the subsequent $\lfloor \lambda/\epsilon \rfloor = \Theta(\lambda n^{1/\gamma})$ steps, we have at most $q(n) \cdot b^{\Theta(\lambda n^{1/\gamma})}$ representations in total – so *every* valid path of length λ is covered by one of these representations. We refer to the set of possible representations ψ at resolution ϵ as $\Psi(\lambda, \epsilon)$.

We can then find a constant A_1 such that

$$|\Psi(\lambda, \epsilon)| \leq q(n) \cdot b^{A_1 \lambda n^{1/\gamma}} \leq (b+1)^{A_1 \lambda n^{1/\gamma}} \qquad (1)$$

(the second inequality holds for all sufficiently large n).

C. The Set Collection Problem

We now define a related problem, which we refer to as the *Set Collection Problem* at resolution ϵ with length λ. Instead of a path $\pi \in \Pi_\lambda$, we want a $\psi \in \Psi(\epsilon, \lambda)$ such that $\hat{S}(\psi)$ contains as many targets as possible.

Definition 11 (The Set Collection Problem)**.**

$$\mathrm{SCP}(X_1, \ldots, X_n; \lambda, \epsilon) \triangleq \max_{\psi \in \Psi(\lambda,\epsilon)} |\hat{S}(\psi) \cap \{X_1, \ldots, X_n\}|$$

Since each $\hat{\pi} \in \hat{\Pi}_\lambda$ is in $\hat{S}(\psi)$ for some $\psi \in \Psi(\lambda, \epsilon)$ (for any ϵ), we can upper-bound ORNT with SCP:

Lemma 5 (SCP $\geq$ ORNT)**.** *For any $\epsilon > 0$,*

$$\mathrm{SCP}(X_1, \ldots, X_n; \lambda, \epsilon) \geq \mathrm{ORNT}_\Pi(X_1, \ldots, X_n; \lambda).$$

D. The Probability Argument for SCP

We can now prove Proposition 2. Note that for any $\psi \in \Psi$,

$$\mathrm{vol}_d[\hat{S}(\psi)] = O(\lambda n^{-(\gamma-1)/\gamma})$$

as $\hat{S}(\psi)$ is a union of $\Theta(\lambda n^{1/\gamma})$ (possibly overlapping) sets of the form $\hat{R}^{\Pi}_{2\epsilon}(\psi_i)$, each with volume $\Theta(1/n)$.

Thus, the probability that an individual point falls within $\hat{S}(\psi)$ (which, since the targets are uniformly distributed at random, is equal to its volume) is at most $O(\lambda n^{-(\gamma-1)/\gamma})$; this means that for some constant A_2, for every $\psi \in \Psi(\lambda, \epsilon)$ and any X_i, $\Pr[X_i \in \hat{S}(\psi)] \leq A_2 \lambda n^{-(\gamma-1)/\gamma}$. Thus,

$$\mathbb{E}[|\hat{S}(\psi) \cap \{X_1, \ldots, X_n\}|] \leq A_2 \lambda n^{1/\gamma}.$$

$|\hat{S}(\psi) \cap \{X_1, \ldots, X_n\}|$ is a sum of independent indicators (one for each X_i), so by the Chernoff Bound, for any $B > 1$:

$$\Pr[|\hat{S}(\psi) \cap \{X_1, \ldots, X_n\}| > B \cdot A_2 \lambda n^{1/\gamma}] \leq e^{-\frac{B^2}{2} \cdot A_2 \lambda n^{1/\gamma}}.$$

This applies for *any* $\psi \in \Psi(\lambda, \epsilon)$; we can then use the Union Bound on this and expression (1) to get

$$\Pr[\exists \psi \in \Psi(\lambda, \epsilon) : |\hat{S}(\psi) \cap \{X_1, \ldots, X_n\}| > B \cdot A_2 \lambda n^{1/\gamma}]$$
$$\leq e^{-\frac{(B-1)^2}{2} \cdot A_2 \lambda n^{1/\gamma}} \cdot (b+1)^{A_1 \lambda n^{1/\gamma}}.$$

But we note that the log of this is

$$\Big(-\frac{(B-1)^2}{2} \cdot A_2 + \log(b+1) \cdot A_1 \Big) \cdot \lambda n^{1/\gamma}, \qquad (2)$$

which is negative if $B > \sqrt{(2A_1 \log(b+1))/A_2} + 1$. Since these inequalities hold for *any* $B > 1$, we set B to be such a constant, and as $n \to \infty$ expression (2) goes to $-\infty$. Thus, the probability that $\mathrm{SCP}(\lambda, \epsilon) > BA_2 \lambda n^{1/\gamma}$ goes to 0 as $n \to \infty$, which by Lemma 5, implies Proposition 2 (with $\alpha_2 = BA_2 > \sqrt{2A_1A_2 \log(b+1)} + A_2$), thus giving us our upper bound to the Orienteering problem.

VI. Lower Bounds for TSP_Π and ORNT_Π

We now complete Theorems 1 and 2.

Proof of the lower bound of Theorem 1. Suppose that the shortest valid path collecting all n points has length L. Then there must be a length-1 subpath of this path which collects at least $n/(2L)$ points (the $1/2$ factor is simply there to correct for the fact that the path is not necessarily a closed loop, and so the natural n/L bound can be escaped by a little bit). By definition, the maximum number of points which can be collected by *any* valid length-1 path is $\mathrm{ORNT}_\Pi(X_1, X_2, \ldots, X_n; 1)$, which by Proposition 2 is with high probability at most $\alpha_2 n^{1/\gamma}$.

But this means, with very high probability,

$$\frac{n}{2L} \leq \alpha_2 n^{1/\gamma} \implies L \geq \frac{2}{\alpha_2} n^{(\gamma-1)/\gamma}$$

thus proving the lower bound to Theorem 1. □

The lower bound to Theorem 2 is similarly shown:

Proof of the lower bound of Theorem 2. By Proposition 1, with very high probability there is a valid path of length at most $\alpha_1 n^{(\gamma-1)/\gamma}$ which collects all the points. We then consider the subpath of length λ that collects the most points; since the average length-λ subpath collects at least

$$\lambda \cdot \frac{n}{2\alpha_1 n^{(\gamma-1)/\gamma}} = \frac{\lambda}{2\alpha_1} \cdot n^{1/\gamma}$$

points, the 'densest' subpath collects at least $\frac{\lambda}{2\alpha_1} n^{1/\gamma}$ points as well (once again, we divide by 2 to avoid technical issues involving the fact that the path is not a closed loop, so points located near the beginning and end are underrepresented in the 'average') - showing our lower bound. □

We note that this proof automatically gives an algorithm for Orienteering as well: use our TSP algorithm to find a path through all n points, and then find the length λ subpath which collects the most points. This achieves the bound given in Theorem 2 with very high probability.

VII. Conclusion

In this paper, we solved the longstanding open problem of determining (up to a constant factor, with very high probability) the length of the shortest trajectory for a kinodynamic vehicle through a set of n randomly-distributed points in a manifold. In particular, we extend and simplify previous results on the dynamic TSP by using techniques from the applied probability literature, which had been used to study the stochastic Orienteering problem (which we solve as well). We additionally provide simplified algorithms for finding approximations of the optimal trajectory (with very high probability) for these problems.

There are many possibilities for future work. An important direction is to broaden the scope of our results to dynamical systems which are not translation-invariant (and hence do not describe vehicles in the common sense), which can be used to represent systems such as multi-jointed robotic limbs. Finally, a variety of related problems – such as the *Traveling Repairman Problem* and the *Dynamic Vehicle Routing Problem* [8, 11, 17] – might be solved by applying the techniques used here.

Another avenue for future work is to more closely examine the relationship between the distribution generating the target points and the length of the TSP tour; in particular, the Beardwood-Halton-Hammersley Theorem [3] gives a model for how such a result should look like. In particular, results for a special class of vehicles, known as *symmetric* or *driftless* systems (whose basic feature is the ability to go into 'reverse' and backtrack over their paths) [4, 5, 13], look very promising for this extension.

Finally, we might consider situations where the points are not distributed randomly, or by a stochastic process which is not i.i.d. A case of particular interest is *adversarial* point generation – where the points are distributed (within some bounded set) by an adversary whose goal is to make the length of the shortest tour as long as possible. Symmetric vehicles also look particularly promising for this extension.

Acknowledgement

We wish to thank Michael Cohen, of the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT, for an extremely useful and insightful discussion about the abstract problem arising from the merging process described in Section IV-C.

References

[1] Adler, A., and Karaman, S. "The Stochastic Traveling Salesman Problem and Orienteering for Kinodynamic Vehicles." *http://peris.mit.edu/Adler.Karaman.ICRA2016.pdf*

[2] Arias-Castro, E., Donoho, D., Huo, X., and Tovey, C., "Connect the Dots: How many random points can a regular curve pass through?", *APT Adv. in Appl. Prob.*, vol.37(3), pp.571-603, 2005.

[3] J. Beardwood, J. Halton, and J. Hammersly, The Shortest Path through Many Points, *Proc. Cambridge Philos. Soc.*, vol.55, pp.299-327, 1959.

[4] Bellaiche, A., and Risler, J., eds. "Sub-Riemannian Geometry," *Birkhauser Verlag*, Basel, Switzerland, 1996.

[5] A Bellaiche, F Jean, and J J Risler. "Geometry of Nonholonomic Systems". In J P Laumond, editor, *Robot Motion Planning and Control*, pp.5591. Springer, 1998.

[6] Bertsimas, D., Chervi, P., and Peterson, M. "Computational Approaches to Stochastic Vehicle Routing Problems", Adv. Appl. Prob., vol.25, pp.947-978, 1991.

[7] Blum, A., Chawla, S., Karger, D., Lane, T., and Meyerson, A., "Approximation Algorithms for Orienteering and Discounted-Reward TSP", *SIAM Journal on Computing*, vol.37(2), pp.653-670, 2007.

[8] Bullo, F., Frazzoli, E., Pavone, M., Savla, K., and Smith, S.L. "Dynamic Vehicle Routing for Robotic Systems," *Proc. of the IEEE*, vol.99(9): pp.1482-1504, 2011.

[9] Enright, J. and Frazzoli, E., "The Traveling Salesman Problem for the Reeds-Shepp Car and the Differential Drive Robot," *Proc. of the 45th IEEE Conf. on Decision and Control*, pp.3058-3064, San Diego, CA, Dec. 2006.

[10] Karp, Richard M. "Reducibility among combinatorial problems". Springer US, 1972.

[11] Itani, Sleiman. "Dynamic Systems and Subadditive Functionals" (Doctoral Thesis). Department of Electrical Engineering and Computer Science, MIT, 2009. Retrieved from http://dspace.mit.edu/handle/1721.1/53282.

[12] Itani, S., Frazzoli, E., and Dahleh, M., "Traveling Salesperson Problems for dynamic systems," *Proc. of the 17th World Congress, Int. Fed. of Auto. Control*, pp.13318-13323, Seoul, Korea, July 2008.

[13] Lavalle, S. M., "Planning Algorithms," *Cambridge University Press*, Cambridge, U.K., 2006.

[14] Le Ny, J., Feron, E., and Frazzoli, E., "On the Dubins Traveling Salesman Problem" *IEEE Trans. on Automatic Control*, vol.57(1), pp.265-270, 2012.

[15] Savla, K., Bullo, F., and Frazzoli, E., "On Traveling Salesperson Problems for a double integrator," *Proc. of the 45th IEEE Conf. on Decision and Control*, pp.3058-3064, San Diego, CA, Dec. 2006.

[16] Savla, K., Frazzoli, E., and Bullo, F., "On the point-to-point and traveling salesperson problems for Dubins' vehicle," *Proc. of the Amer. Control Conf.*, pp.786-791, Portland, OR, June 2005.

[17] Savla, K., Frazzoli, E., and Bullo, F., "Traveling Salesperson Problems for the Dubins Vehicle," *IEEE Trans. on Autom. Control*, vol.53(6), pp.1378-1391, July 2008.

[18] Savla, K., Frazzoli, E., and Bullo, F., "On the Dubins Traveling Salesperson Problems: Novel approximation algorithms," *Robotics: Science and Systems II*, 2006.

[19] Vansteenwegen, P., Souffriau, W., and van Oudheusden, D. "The Orienteering Problem: a Survey," *Euro. Journal of Operations Research*, vol.209, pp.1-10, 2011.