The Traveling Salesman Problem and Orienteering for Kinodynamic Vehicles

by

Aviv Adler

A.B., Mathematics, Princeton University, 2014

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering and Computer Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY September 2017

© 2017 Massachusetts Institute of Technology. All rights reserved.

Signature of Author: ................................

Department of Electrical Engineering and Computer Science August 31, 2017

Signature redacted

Certified by: ..................................................

Sertac Karaman
Professor of Aeronautics and Astronautics Thesis Supervisor

Signature redacted

Accepted by: .............................................

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science Chair of the EECS Committee on Graduate Students
The Traveling Salesman Problem and Orienteering for Kinodynamic Vehicles
by
Aviv Adler

Submitted to the Department of Electrical Engineering and Computer Science on August 31, 2017, in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering and Computer Science

ABSTRACT

The Traveling Salesman Problem is a major foundational problem in the fields of Computer Science, Operations Research, and Applied Mathematics, in which an agent wants to visit a set of target points with the shortest path possible. This problem is of the highest interest both theoretically and in practice. When the agent is a vehicle whose trajectory must satisfy a set of dynamic constraints and the target points are distributed over a continuous space, this problem is especially relevant to robotics. Although this problem is considered computationally intractable to solve precisely, in many settings a good approximate path can be computed efficiently. We study the case where the target points are distributed independently at random and ask how the length of the optimal tour grows as the number of such target points increases, a question which has attracted interest from both the robotics and motion planning community and the applied probability community; however, there has been little interaction between the two communities on this problem.

By combining the approaches developed independently by these two communities, we re-derive the most general and powerful results with a simplified method. We then demonstrate the power of our method by extending it to show novel stronger results for an important sub-class of vehicles, as well as novel results for an alternative setting in which the target points are distributed by an adversary rather than at random.

Thesis Supervisor: Sertac Karaman
Title: Professor of Aeronautics and Astronautics
Acknowledgements

This work would not have been possible without the following people, to whom I owe my deepest gratitude:

- Professor Sertac Karaman, who provided an immensely interesting problem for me to work on and who has always treated me with great generosity during my time at MIT. Without his ideas, support and guidance, none of this work would have seen the light of day.

- My friends and professors, who have made my time at MIT a great pleasure, have been source of immense inspiration, and have been invaluable in teaching me – both in computer science and about life in general. I would especially like to thank Michael Cohen, of the MIT EECS Department, for the insights he provided on the critical result developed in Appendix A, and Professor Erik Demaine, with whom I worked on a number of other projects and whose classes were a special joy.

- My parents – without their tremendous love, support, and encouragement, nothing at all would have been possible in my life.
## Contents

1 Introduction and Preliminaries 9
  1.1 Motivation .................................................. 9
    1.1.1 The Traveling Salesman Problem and Robotics ............ 9
  1.2 Previous Work ............................................. 9
    1.2.1 The Stochastic TSP for Specific Vehicles ................ 9
    1.2.2 Sub-Additivity with Dynamics ............................ 10
    1.2.3 Stochastic Orienteering (Connect-the-Dots) ............... 10
    1.2.4 Stochastic Euclidean TSP ................................ 11
  1.3 Contribution ............................................... 11
    1.3.1 For General Vehicles ................................... 12
    1.3.2 For Symmetric Vehicles ................................. 12
  1.4 Basic Notions, Notation, and Assumptions ..................... 12
    1.4.1 The Work and Configuration Spaces ....................... 12
    1.4.2 Constraints and Trajectories ........................... 13
    1.4.3 Basic Notation ......................................... 14
    1.4.4 Reachable and Deviation Sets ............................ 15
    1.4.5 The Small-Time Constraint Factor and Agility Function .. 16
    1.4.6 The TSP and ORNT Functions ............................. 17
  1.5 Assumptions and How to Deal with Them ........................ 17
    1.5.1 The Spaces $\mathcal{X}$ and $\mathcal{Z}$ (a.k.a. Traveling to Flatland) . 17
    1.5.2 Assumptions about the Target Points ..................... 18
    1.5.3 Vehicular Constraints and Assumptions .................. 18
  1.6 Summary of Main Results .................................... 19
    1.6.1 Results for General Vehicles ............................ 20
    1.6.2 Results for Symmetric Vehicles ........................ 20

2 Dynamic Constraints and Vehicular Motion 23
  2.1 Notation and Space ......................................... 23
    2.1.1 The Workspace and Configuration Space .................... 23
    2.1.2 Notation ................................................. 24
  2.2 Vehicles and Control ....................................... 24
    2.2.1 Controllability ......................................... 24
    2.2.2 Local Controllability ................................. 25
  2.3 Simplification Tactics ..................................... 26
    2.3.1 Projected Dynamics ..................................... 26
  2.4 Main Lemmas ............................................... 27
    2.4.1 The Ball-Box Theorem ................................... 27
    2.4.2 Parallel Trajectories .................................. 28
  2.5 Space-Covering Tilings ..................................... 29
    2.5.1 Space-Covering Trajectories .............................. 29
    2.5.2 Adjacent Tiling for Symmetric Vehicles .................. 32
3 Results for General Vehicles
  3.1 Upper Bound of TSP\(_\Pi\) .......................... 33
    3.1.1 Tiling the Workspace .......................... 33
    3.1.2 Collecting a Point from Each Cell .............. 34
    3.1.3 Merging Cells ................................. 35
    3.1.4 A Combinatorial Representation of Collect-and-Merge 35
    3.1.5 Synthesizing the TSP Algorithm ................. 36
  3.2 Upper Bound of ORNT\(_\Pi\) .......................... 36
    3.2.1 Implicitly Discretizing the Workspace .......... 37
    3.2.2 A Discrete Representation Scheme for Valid Paths 38
    3.2.3 The Set Collection Problem ..................... 39
    3.2.4 The Probability Argument for SCP .............. 39
  3.3 Lower Bounds for TSP\(_\Pi\) and ORNT\(_\Pi\) ............ 39

4 Results for Symmetric Vehicles
  4.1 Vehicles with Symmetric Constraints .................. 41
  4.2 Lower-Bound for TSP\(_\Pi\) in the Stochastic Setting .... 42
    4.2.1 The Cost-Balanced Orienteering Problem ........ 42
    4.2.2 Upper-Bounding CBO\(_\Pi\) ...................... 44
    4.2.3 Lower-Bounding TSP\(_\Pi\) from CBO\(_\Pi\) .......... 45
  4.3 Upper Bound of TSP\(_\Pi\) in the Stochastic Setting .... 46
  4.4 Orienteering for Symmetric Vehicles in the Stochastic Setting 48
  4.5 Symmetric Vehicles and Adversarially-Distributed Targets 49
    4.5.1 The Lower Bound on the Worst Case .............. 49
    4.5.2 The Upper Bound on the Worst Case ............. 51
    4.5.3 Orienteering with Adversarial Targets .......... 51
  4.6 Targets on Sets of Lesser Dimension .................. 52

5 Future Work and Conclusion
  5.1 Future Work ......................................... 53
    5.1.1 Multidimensional Surfaces ...................... 53
    5.1.2 Non-Full-Dimensional \(F\) for Symmetric Vehicles 53
    5.1.3 Better Constant Bounds ........................ 53
    5.1.4 A Full Extension of Beardwood-Halton-Hammersley 53
    5.1.5 Guaranteed Approximation Algorithms .......... 54
  5.2 Conclusion .......................................... 54

A Analyzing the Abstraction
  A.1 The Ball Collection Problem .......................... 57
    A.1.1 A Representation of the TSP Algorithm .......... 57
    A.1.2 The Problem ................................... 57
    A.1.3 An Algorithm for Low-Cost Ball Removal (ouch!) 58
    A.1.4 Analysis ...................................... 59

B Bounding Binomials
  B.1 Quick and Dirty .................................... 65
    B.1.1 Upper Bound ................................... 65
    B.1.2 Lower Bound .................................. 65
  B.2 More Satisfying Technique ........................... 66

C Curious Counterexamples
  C.1 Lesser-Dimensional \(F\) .............................. 67
    C.1.1 Exploding Tour Length for General Vehicles .... 67
    C.2 Unbounded \(S\) .................................. 67
Chapter 1

Introduction and Preliminaries

1.1 Motivation

1.1.1 The Traveling Salesman Problem and Robotics

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. Although the TSP is often given on a graph, an important variant involves visiting locations in some continuous space. We will be considering this variant, as it corresponds naturally to motion-planning problems.

The TSP is well-known to be computationally challenging – specifically, finding the shortest tour through a given set of points is NP-complete [8], even for Euclidean paths through points in $\mathbb{R}^2$ [14]. 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, a number of robot motion planning and routing algorithms employ TSP algorithms at their core [12]. The applications of TSP in the robotics domain are far reaching, including persistent monitoring, surveillance, reconnaissance, exploration, among other important problems.

An important variant of the TSP is the case in which the locations are independently and identically distributed (i.i.d.); we refer to this as the Stochastic TSP.

However, in most cases in robotics, the vehicles are subject to non-trivial differential constraints which have a substantial impact on the TSP tour. This means that new algorithms must be used to deal with the TSP in these instances [13]. Furthermore, understanding the impact of the dynamics on the length of the tour would allow a system designer to pick the best robot 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 [2]; In 1991, its application to vehicle routing was recognized [4]. More recently, 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) [7, 15, 16, 17, 18]. The stochastic TSP for a symmetric dynamical system in 2 dimensions was considered by Itani et al. [9], and generalized to all vehicles with affine control in $d$-dimensional Euclidean space.

1.2 Previous Work

1.2.1 The Stochastic TSP for Specific Vehicles

The Stochastic TSP for vehicles is a well-studied problem. In particular, the rate of growth of the length of the shortest path through $n$ i.i.d. random points is known for a handful of specific vehicles:
for the unconstrained vehicle (which can follow any continuous path) in \( \mathbb{R}^d \), the length of the optimal path is \( \Theta(n^{\frac{d-1}{2}}) \) [2] (with high probability). This is actually only a part of a well-known result concerning the length of the shortest Euclidean path through \( n \) random points in \( \mathbb{R}^d \), known as the Beardwood-Halton-Hammersley Theorem, which was proven in 1959.

- for a Dubins Car in \( \mathbb{R}^2 \), the length of the path is \( \Theta(n^{\frac{3}{2}}) \) [15, 16] (with high probability)
- for a Reeds-Shepp Car in \( \mathbb{R}^2 \), the length of the path is \( \Theta(n^{\frac{3}{2}}) \) [7] (with high probability)
- for a Differential Drive vehicle in \( \mathbb{R}^2 \), the length of the path is \( \Theta(n^{\frac{3}{2}}) \) [7] (with high probability)
- for a double-integrator (i.e. a simple model of a quadcopter) the length of the path is \( \Theta(n^{\frac{3}{2}}) \) in \( \mathbb{R}^2 \) and \( \Theta(n^{\frac{5}{4}}) \) in \( \mathbb{R}^3 \) [17, 18] respectively.

We note that all of these results have a particular commonality, that the resulting length of the path is always of the form \( \Theta(n^{\frac{d-1}{2}}) \). As it turns out, this is the result of a general rule: there is a particular parameter \( \gamma \) (related to the space in which the target points are distributed and the constraints on the vehicle's motion) such that as the number of points \( n \) grows, the length of the shortest path grows proportionally to \( n^{\frac{\gamma}{d-1}} \).

In particular, the work of Savla, Frazzoli, and Bullo [16] shows the upper-bound on the length of the tour through \( n \) points by a Dubins Car by use of an algorithm called the Recursive Bead-Tiling Algorithm. Our algorithm, which holds for all vehicles, is in effect a generalization of this algorithm (note that their later algorithm for Dubins Cars – [17] – is not generalizable in the same manner).

1.2.2 Sub-Additivity with Dynamics

The Stochastic TSP in Euclidean space (without dynamic constraints) can also be viewed as a special case about a class of sub-additive functions called sub-additive Euclidean functionals, an angle explored by Steele [19]. In his Ph.D. work, Sleiman Itani applied a geometric understanding of dynamic constraints to generalize this idea to a class of functions he called sub-additive Quasi-Euclidean functionals, of which the Stochastic TSP with dynamics is the most prominent example – first in a relatively constrained case [9] and then in the more general setting of affine-control vehicles for his Ph.D. thesis (supervised by Professors Dahleh and Frazzoli) [10].

This work resolved the problem for all vehicles which are affine in control and move through Euclidean space, and provided algorithms for both symmetric and non-symmetric vehicles.

1.2.3 Stochastic Orienteering (Connect-the-Dots)

In the Stochastic Orienteering problem, one is still given a constrained vehicle and a set of \( n \) randomly-distributed points, but the objective is to visit as many of these points as possible with a trajectory of length at most \( \lambda \). In addition to being an interesting problem in its own right, it is deeply connected to the Stochastic TSP, and in fact, an upper bound on the Stochastic Orienteering problem can be translated directly into a lower bound on the Stochastic TSP. While we will go deeper into this argument later, the basic intuition is that if one can only collect a certain number \( k \) of points with a length-\( \lambda \) trajectory, to collect all \( n \) points will take at least a path of length \( \frac{n^2}{k} \lambda \).

In addition to the work on the Stochastic TSP, there has been an interesting development in solving the Stochastic Orienteering problem, courtesy of Arias-Castro, Donovo, Huo, and Tovey in their 2005 paper Connect the Dots [1]. This work represented a breakthrough in that it gives a general recipe for producing an upper bound to the Stochastic Orienteering problem (given the set of constraints on the vehicle). The general outline is as follows (given the number \( n \) of points and the constraints and maximum length \( \lambda \) of the path with which we wish to solve the Stochastic Orienteering problem):

- define a finite set of paths through the space (which we call the discretization of the space), such that any path of length at most \( \lambda \) which obeys the constraints is sufficiently ‘close’ to a path in the finite set;
1.3. CONTRIBUTION

- define a path in the finite set to collect a random point if it passes sufficiently close to the point, and use the fact that the points are distributed randomly to mathematically characterize the random variable describing the number of points collected by an arbitrary path from the finite set;
- use a concentration bound (specifically, the Chernoff Bound) to show that it is exceedingly unlikely for any given path from the finite set to collect more than a certain number of the random points;
- use the Union Bound to show that this means there is very little probability that there exists a path from the finite set collecting more than this number of points;
- finally, use the fact that any path obeying the constraints is approximated by a path from the finite set to show that no path obeying the constraints can collect more than this number of points.

They outline this method and demonstrate it in action on the (unconstrained) Euclidean paths and Dubins Cars. Although this method is exceedingly powerful and general, there is a serious drawback: one must apply it separately for any set of constraints — in particular, defining the finite set of paths described above — and even for the case of the Dubins Car, this is highly nontrivial.

In this work, we will use a modified version of their method, in which the discretization step is done automatically (and provably achieves its intended goals), thus eliminating the requirement of a separate proof for every new case.

1.2.4 Stochastic Euclidean TSP

Finally, a great deal of work has been done for the special case of the Stochastic Euclidean TSP, in which the goal is to connect \( n \) i.i.d. points in \( \mathbb{R}^d \) with a continuous path of minimum Euclidean length; such a path can be shown to consist of a sequence of straight line segments, each joining two points which need to be visited. In 1959, the following theorem was shown [2]:

The Beardwood-Halton-Hammersley Theorem. Let \( F \) be a probability distribution over \( \mathbb{R}^d \) with bounded support and density function \( f \), and let \( X_1, X_2, \ldots \sim F \) independently. Let \( TSP(X_1, X_2, \ldots, X_n) \) denote the length of the shortest tour through \( X_1, X_2, \ldots, X_n \) by a continuous path (with no dynamic constraints). Then there exists a constant \( \beta_d \) (depending only on the dimension \( d \) of the space) such that

\[
\lim_{n \to \infty} \frac{TSP(X_1, X_2, \ldots, X_n)}{n^{(d-1)/d}} = \beta_d \int_{\mathbb{R}^d} f(x)^{(d-1)/d} dx
\]

with probability 1.

We note that this can be considered as a problem involving a ‘vehicle’ with no dynamic constraints (aside from the constraint of following a continuous path), and that this ‘vehicle’ satisfies our intuitive notion of symmetry (it satisfies the formal definition as well, but we have yet to define this). Thus, it’s natural to try to generalize this theorem to the class of symmetric vehicles, i.e. those vehicles for which a valid trajectory remains valid even when time-reversed.

1.3 Contribution

In this paper, we introduce a set of properties on dynamic constraints which intuitively correspond to the behavior of vehicles; this set of properties is very general and applies to the constraints of virtually all commonly-studied vehicles, including all the vehicles discussed in the previous section. We refer to any set of constraints satisfying these properties as vehicular, and is associated with an integer constant \( \gamma \) (referred to in Itani's Ph.D. thesis as \( \|r\|_1 \) [10]) which we refer to as the small-time constraint factor). Intuitively, \( \gamma \) denotes the degree to which the vehicle's dynamics constrain the region of space that it can explore in a short time. This factor tends to grow with the dimension of the space itself (because this decreases the fraction of the space that is reachable in a short span of time) and with the degree of separation between the control input of the vehicle and the effect on the vehicle's motion.

Broadly speaking, we show that the length of the shortest path a vehicle (obeying vehicular constraints) can make through \( n \) points grows at a rate proportional to \( n^{2-\gamma} \). We also show the equivalent result for
Orienteering, that the maximum number of points (out of n) which a vehicle can visit with length-λ path is proportional to \( \lambda n^{\frac{1}{\lambda}} \). Additionally, we provide efficient algorithms for both TSP and Orienteering which achieve these bounds with very high probability (and hence are constant-factor approximations of the optimal solutions).

Although our basic results replicate those found in Itani’s Ph.D. thesis, we achieve them with a much different (and, for us, more intuitive and visual) method founded on discrete probability, rather than geometry. This provides additional insight into the nature of this problem, and allows us to obtain novel, more thorough results when working with the sub-class of symmetric vehicles. We also formulate and solve a novel lemma concerning discrete probability [see Appendix A], which we consider to be an interesting and worthwhile problem in its own right.

1.3.1 For General Vehicles

For general vehicles, our results concern stochastic versions TSP and Orienteering. We show that for any vehicle:

- the length of the TSP path through \( n \) i.i.d. points is, with very high probability, proportional to \( (n^{\frac{1}{\gamma}}) \);

- the maximum number of points which can be visited by a path of length at most \( \lambda \) (out of \( n \) i.i.d. points) is, with very high probability, proportional to \( \lambda n^{\frac{1}{\lambda}} \).

We therefore settle the long-standing open problem of how long a path a vehicle needs to take to visit \( n \) i.i.d. points. We thus unify the aforementioned existing results on the TSP – that is, they are a direct corollary of our results – and extend the reach of the existing literature to cover the dynamics of virtually all vehicles.

1.3.2 For Symmetric Vehicles

We then give stronger results for symmetric vehicles. In particular, for symmetric vehicles, we can provide worst-case bounds on the length of the TSP path and on the number of points which can be visited by a length-λ path – and in fact, we show the length (up to a constant factor) of the shortest path through \( n \) adversarially-distributed points. We can also strengthen the bounds obtained for the Stochastic TSP for general vehicles, by showing how the constant factor multiplying the growth term \( n^{\frac{1}{\gamma}} \) depends on the distribution which is used to generate the points, generalizing the Beardwood-Halton-Hammersley Theorem. As with the results for general vehicles, we also provide tight bounds (up to a constant factor) for the Orienteering problem under the same conditions.

Furthermore, in order to demonstrate these theorems, we define and solve a slight generalization of the TSP and Orienteering in these settings for symmetric vehicles; specifically, we replace the notion of ‘length’ of a path by ‘cost’. Intuitively, we impose a cost function over the space in which the target points are distributed, and a path incurs cost proportional to the value of this function in the regions it passes through.

1.4 Basic Notions, Notation, and Assumptions

1.4.1 The Work and Configuration Spaces

We first discuss the target points which the vehicle is supposed to visit. 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, e.g. where every target point comes with a heading vector and the vehicle must be traveling in the direction of the heading vector in order to collect the point. We refer to the space in which the points are distributed as the workspace and denote it \( \mathcal{X} \). The volume of a Borel set \( W \subseteq \mathcal{X} \) is denoted by \( \text{vol}_{\mathcal{X}}(W) \). If \( \mathcal{X} = \mathbb{R}^d \) we denote the volume by \( \text{vol}_{d}(W) \).

We model the constraints on our vehicle by considering the notion of the configuration space \( Z \). This includes the vehicle’s location in the workspace, but also contains 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 \( Z \) 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 \( Z \) onto \( X \); more specifically, we will model the configuration space as being expressible as \( Z = X \times S \), where \( S \) is another manifold.

The projection of a set \( Z \subseteq Z \) onto \( X \) is denoted \([Z]_X\). Similarly, for a set \( W \subseteq X \), we denote by \([W]^Z\) the set of all configurations \( z \in Z \) whose projection onto \( X \) is in \( W \), i.e.

\[
[W]^Z \triangleq \{ z \in Z : [z]_X \in W \}.
\]

The length of the path is most generally defined as being dependent on the trajectory in \( Z \); 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 location in \( X \). In particular, we will be making assumptions which assure that \( Z \) is an algebraic subset of a manifold (of dimension at least \( d \)); the length of a path is defined as its length in \( Z \).

**Assumption 1 (Workspace and Configuration Space as Manifolds).** We will assume that \( X, Z, S \) (where \( Z = X \times S \)) are manifolds with bounded curvature, embedded in Euclidean space (\( X \) in \( \mathbb{R}^{dx} \), \( S \) in \( \mathbb{R}^{ds} \), and \( Z \) in \( \mathbb{R}^{dx} \times \mathbb{R}^{ds} \)). We also assume that \( S \) is itself bounded.

Note that a vehicle is defined by the configuration space; depending on the problem, the workspace can differ even for the same vehicle, because the target points can be distributed in different sub-spaces of the configuration space. We now give the example of a Dubins Car, which is a vehicle that can only travel forward at a fixed speed, and can only change its heading at or below a fixed speed. Its trajectories are simply those with bounded curvature.

**Example (Dubins Car).** For the Dubins Car in 2 dimensions, we have \( Z = \mathbb{R}^2 \times S^1 \) (where \( S^1 \) is the one-dimensional sphere, and denotes the heading of the vehicle). More generally, for the Dubins Car in \( d \) dimensions, \( Z = \mathbb{R}^d \times S^{d-1} \). If the problem is to collect points in \( \mathbb{R}^2 \), as studied by Savla and Frazzoli and by Arias-Castro et al. (which they refer to as "connect the dots"), then \( X = \mathbb{R}^2 \) and \( S = S^1 \). However, we might instead have the case where the points also come with a heading and have to be visited while traveling in this direction, which Arias-Castro et al. refer to as "connect the darts"; in this case \( X = \mathbb{R}^2 \times S^1 \) and \( S \) is just the zero-dimensional space.

1.4.2 Constraints and Trajectories

Although we will explore the details of dynamic constraints in the next chapter, in order to clearly state our results we must address a few basic points about the kind of trajectories we are studying.

In particular, we define the notions of memorylessness and symmetry for sets of trajectories. All the sets of trajectories we describe are memoryless, which intuitively just means that the present configuration of the vehicle always fully defines the possible trajectories it can take in the future (which is the reason we need the notion of a configuration space which contains the workspace). Symmetric constraint sets constitute and important subset, for which we can give stronger results; intuitively, a set of constraints is symmetric if a valid trajectory played backwards is also valid.

**Definition 1 (Trajectory and Length).** A trajectory is a continuous function \( \pi : [0, T] \to Z \), where \( T \) is the length of the trajectory; given a trajectory \( \pi \), we also denote its length by \( \ell(\pi) \).

Later, we will use the word "trajectory" to mean valid trajectory, i.e. a trajectory that satisfies our constraint set \( \Pi \).

We now define the set of constraints \( \Pi \); by another small abuse of notation, we use it to describe both the dynamic constraints themselves, and the set of trajectories through \( Z \) which satisfy those constraints. At any given instant, the vehicle takes a control \( u \in [-1, 1]^m \) (where \( m \) is some integer), which, along with its current configuration \( z \in Z \), determines its velocity and direction. We will assume that the vehicle is affine in control.
Definition 2 (Affine Vehicles). A vehicle defined by constraint set \( \Pi \) is affine in control if there exist functions \( f_0, f_1, \ldots, f_m \) such that for every \( \pi \in \Pi \), there exists a control function \( u \) such that for every \( t \),

\[
\pi(t) = f_0(\pi(t)) + \sum_{i=1}^{m} f_i(\pi(t))u_i
\]

where \( f_0, \ldots, f_m : \mathbb{Z} \to \mathbb{R}^{dz} \) and are always tangent to their input points (so as to keep the vehicle in \( \mathbb{Z} \)).

Example (Dubins Car). A Dubins Car can be modeled in these terms by the following: \( \mathbb{Z} = \mathbb{R}^2 \times S^1 \) (where \( S^1 \) is the one-dimensional spherical set), which is embedded in \( \mathbb{R}^4 \). For any control \( u \in [-1, 1] \) (it takes in only one control input) and at any location \( z \in \mathbb{Z} \) (which we denote as \( z = (z_1, z_2, z_3, z_4) \) where \( z_3^2 + z_4^2 = 1 \)), we have

\[
z = (z_3, z_4, 0, 0) + u \cdot (0, 0, -z_4, z_3)
\]

Remark: The notation and concepts above will be explored more thoroughly in the next chapter, and will be dropped thereafter in favor of a more abstract view (after outlining certain important theorems, most notably the Ball-Box Theorem).

Definition 3 (Length-limited trajectory set). For any \( \lambda \in \mathbb{R}_{>0} \), we define \( \Pi_\lambda = \{ \pi \in \Pi : \ell(\pi) \leq \lambda \} \).

This definition will become important in the definition of the Orienteering problem.

Definition 4 (Memoryless Constraints). A constraint set \( \Pi \) is memoryless if for any trajectories \( \pi_1, \pi_2 \in \Pi \) such that the endpoint of \( \pi_1 \) is the starting point of \( \pi_2 \), concatenating them gives another trajectory in \( \Pi \).

The idea of memorylessness is essentially implied in the concept of the configuration space, since the total configuration of the vehicle is supposed to give us full information about its possible futures. We now define symmetric constraints. In order to do so, it is useful for us to first define reverse trajectories:

Definition 5 (Reverse Trajectories). If \( \pi \) is a trajectory, we define its reverse trajectory \( \pi^R : [0, \ell(\pi)] \to \mathbb{Z} \) (which by definition has the same length) by \( \pi^R(t) = \pi(\ell(\pi) - t) \). For the set of trajectories \( \Pi \), we denote the set of reverse trajectories as \( \Gamma \), i.e \( \pi \in \Pi \iff \pi^R \in \Gamma \).

Note that this definition is for trajectories in \( \mathbb{Z} \), not their projection into \( \mathcal{X} \).

Definition 6 (Symmetric Constraints). A set of dynamic constraints \( \Pi \) is symmetric if \( \Pi = \Gamma \).

Note that this definition is crucially about trajectories in \( \mathbb{Z} \); this means, for example, that a Dubins car is not symmetric because to go in reverse, it would have to instantaneously flip its heading around. A symmetric and memoryless vehicle can switch back and forth at will, since it can concatenate ‘reverse’ trajectories with ‘forward’ trajectories.

1.4.3 Basic Notation

For clarity, we present here in one place our notation for trajectories through \( \mathbb{Z} \) and \( \mathcal{X} \):

- \( x \) (and variations like \( x' \)) will always signify a point in \( \mathcal{X} \); \( z \) (and variations) will always signify a point in \( \mathbb{Z} \);
- the projection of \( z \) onto \( \mathcal{X} \) is denoted \( [z]_\mathcal{X} \);
- the pre-image of projection onto \( \mathcal{X} \) of \( x \) is denoted \( [x]^Z \), i.e. \( [x]^Z \triangleq \{ z \in \mathbb{Z} : [z]_\mathcal{X} = x \} \);
- \( \mathcal{X}, \mathcal{S}, \mathcal{Z} \) are manifolds of dimension \( d_\mathcal{X}, d_\mathcal{S}, d_\mathcal{Z} \), respectively (and by definition \( d_\mathcal{Z} = d_\mathcal{X} + d_\mathcal{S} \));
- \( \mathcal{X}, \mathcal{S}, \mathcal{Z} \) are represented as being embedded in Euclidean spaces of dimension \( d'_\mathcal{X}, d'_\mathcal{S}, d'_\mathcal{Z} \), respectively (and by definition \( d'_\mathcal{Z} = d'_\mathcal{X} + d'_\mathcal{S} \));
- \( z = (z_X, z_S) \) where \( z \in \mathbb{Z} \subseteq \mathbb{R}^{dz} \), \( z \in \mathcal{X} \subseteq \mathbb{R}^{d_\mathcal{X}} \), and \( z \in \mathcal{S} \subseteq \mathbb{R}^{d_\mathcal{S}} \);
1.4. BASIC NOTIONS, NOTATION, AND ASSUMPTIONS

- \( \Pi \) denotes both the set of constraints on the vehicle, and the set of trajectories through \( Z \) which satisfy those constraints (and which we call valid);
- for any trajectory \( \pi \in \Pi \), we denote its length by \( \ell(\pi) \), meaning \( \pi \) is a function from \( [0, \ell(\pi)] \) to \( Z \);
- for \( \pi \in \Pi \), we let \( \hat{\pi} : [0, \ell(\pi)] \to \mathcal{X} \) so that \( \hat{\pi}(t) = [\pi(t)]_X \) for all \( t \), and we let \( \hat{\pi} \subseteq \mathcal{X} \) be the set of points which lie on \( \hat{\pi} \);
- \( \hat{\Pi} \) denotes the set of trajectories through \( \mathcal{X} \) corresponding to a valid trajectory, i.e. \( \hat{\Pi} \triangleq \{ \hat{\pi} : \pi \in \Pi \} \);
- for any \( \lambda > 0 \), we denote the set of valid trajectories with length at most \( \lambda \) by \( \Pi_\lambda \), and the set of trajectories in \( \mathcal{X} \) corresponding to this by \( \hat{\Pi}_\lambda \triangleq \{ \hat{\pi} : \pi \in \Pi_\lambda \} \);
- we say \( \pi \) goes from \( z \) to \( z' \) (where \( z, z' \in Z \)) if \( \pi(0) = z \) and \( \pi(\ell(\pi)) = z' \), or from \( x \) to \( x' \) (where \( x, x' \in \mathcal{X} \)) if \( \hat{\pi}(0) = x \) and \( \hat{\pi}(\ell(\pi)) = x' \).

Remark: In general we use the hat notation \( \hat{\cdot} \) to denote workspace versions of things which are also defined in the configuration space.

1.4.4 Reachable and Deviation Sets

The small-time reachable set (terminology we share with Itani’s thesis) is a key notion in the analysis of these problems, and in fact the size of the small-time reachable sets is the factor which determines the length of the optimal tour as the number of target points increases to infinity.

**Definition 7** (\( \varepsilon \)-Time Reachability). Let \( z \in Z \) and \( \varepsilon > 0 \); then the \( \varepsilon \)-time reachable set \( R_\varepsilon(z) \) of \( z \) at length \( \varepsilon \) is

\[
R_\varepsilon(z) \triangleq \{ z' \in Z : \exists \pi \in \Pi_\varepsilon \text{ which goes from } z \text{ to } z' \}.
\]

We define the projected \( \varepsilon \)-time reachable set \( \hat{R}_\varepsilon(z) \) as \( \hat{R}_\varepsilon(z) \triangleq [R_\varepsilon(z)]_X \). For either of these sets, the configurations \( z \) is referred as its anchor point, and \( \varepsilon \) is referred to as its radius.

Both \( R_\varepsilon(z) \) and \( \hat{R}_\varepsilon(z) \) play a major role in our method because it is important to know both what configurations can be reached (which are in \( Z \)) and what target points can be reached (which are in \( \mathcal{X} \)).

We now introduce an extension of the small-time reachable set, which will make it significantly easier to describe our algorithm: the small-time deviation set. Intuitively, our algorithm sends the vehicle along a fixed trajectory, and makes slight detours off of this trajectory in order to visit target points (returning to the trajectory afterwards), and the small-time deviation set is the set of points reachable with a short detour.

**Definition 8** (\( \varepsilon \)-Time Deviation). Let \( \pi \in \Pi \), \( t \in [0, \ell(\pi)] \), and \( \varepsilon > 0 \) (such that \( t + \varepsilon \in [0, \ell(\pi)] \) as well). We then define the \( \varepsilon \)-time deviation set of \( \pi \) at \( t \) as

\[
D_\varepsilon(t, \pi) \triangleq \{ z \in Z : z \in R_\varepsilon(\pi(t)) \text{ and } \pi(t + \varepsilon) \in R_\varepsilon(q) \}.
\]

Similar to \( \varepsilon \)-time reachability, we define the projected \( \varepsilon \)-time deviation set as

\[
\hat{D}_\varepsilon(t, \pi) \triangleq [D_\varepsilon(t, \pi)]_X
\]

We also introduce the notion of a flexible trajectory. This is a trajectory where the small-time deviation sets are only smaller than the small-time reachable sets by a constant factor, which is important because we want to be able to cover a large area when we deviate from our fixed trajectory.

**Definition 9** (\( \varepsilon \)-Flexible Paths). A path \( \pi \) is \( \varepsilon \)-flexible for constant \( \varepsilon > 0 \) if, for every \( t \in [0, \ell(\pi)] \),

\[
\lim_{\varepsilon \to 0} \frac{\text{vol}_X(\hat{D}_\varepsilon(t, \pi))}{\varepsilon^r} \geq c.
\]

Intuitively, a \( \varepsilon \)-flexible path is one from which the vehicle may take small deviations from any point and then return. A path which is \( \varepsilon \)-flexible for some nonzero constant \( c \) is referred to as flexible.
CHAPTER 1. INTRODUCTION AND PRELIMINARIES

Figure 1.1: An example of (projected) reachability and deviation sets for a Dubins car. (a) The $\epsilon$-reachable set from configuration $q$ (represented by the point and 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 3) as applied to Dubins cars. (b) The $\epsilon$-deviation set of path $\pi$ at $\pi(t)$; the dimensions of the contained orange box indicate that $\pi$ is flexible at $\pi(t)$. This diagram uses $q$ because it was created before we switched notation to $z$.

Definition 10 (Uniform c-Flexibility). A collection of paths $\Phi$ 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.

Example (Flexible and Non-Flexible Trajectories). For a Dubins Car with maximum curvature 1, a straight-line trajectory is flexible (the deviation sets correspond to the ‘beads’ in the algorithm given by Savla and Frazzoli [15]), while the trajectory given by $\pi(t) = (\sin(t), \cos(t))$ (moving counterclockwise on the unit circle) is not flexible because any deviation makes it impossible to return to the circle in a short time.

1.4.5 The Small-Time Constraint Factor and Agility Function

We mentioned in the introduction that the most significant factor was the volume of the reachable sets (in $\mathcal{X}$) on short time-scales. Based on the size of the reachable sets and deviation sets as $\epsilon \to 0$, we formally define the small-time constraint factor and agility function. Intuitively, the small-time constraint factor $\gamma$ represents the rate at which the volume of $\hat{R}_\epsilon(z)$ shrinks to 0 as $\epsilon \to 0$, and should be constant across all $z \in Z$; the agility function $g$ takes in $z \in Z$ and returns the constant factor for this volume. Thus, what we really want is that for sufficiently small $\epsilon$ and all $z \in Z$,

$$\text{vol}_\mathcal{X}(\hat{R}_\epsilon(z)) \approx g(z)\epsilon^\gamma$$

Among our assumptions will be that $\gamma$ and $g : Z \to R_{>0}$ are well-defined and consistent across configurations.

Formally, we give the following definitions:

Definition 11 (The Small-Time Constraint Factor). The small-time constraint factor $\gamma$ is:

$$\gamma \triangleq \lim_{\epsilon \to 0} \log \left( \frac{\text{vol}_\mathcal{X}(\hat{R}_\epsilon(z))}{\log(\epsilon)} \right)$$

Note that we assume this is the same for all $z$.

We now introduce the agility functions— one $g^Z : Z \to R_{>0}$ and the other $g : \mathcal{X} \to R_{>0}$. We refer to these as the configuration space and workspace agility functions, respectively; for simplicity, since the configuration space agility function is not used except as a way to explain and define the workspace agility function (which is the one which appears in our theorems), we will generally refer to the latter as just the “agility function”. As stated before, the agility function represents the constant factor on the volume $\text{vol}_\mathcal{X}\hat{R}_\epsilon(z)$.

Definition 12 (The Agility Functions). The configuration-space agility function $g^Z : Z \to R_{>0}$ is:

$$g^Z(z) \triangleq \lim_{\epsilon \to 0} \frac{\text{vol}_\mathcal{X}(\hat{R}_\epsilon(z))}{\epsilon^\gamma}$$

The workspace agility function $g : \mathcal{X} \to R_{>0}$ is then defined as $g(x) \triangleq \max_{s \in S} g^Z(x, s)$. 


1.5. Assumptions and How to Deal with Them

The reason we need these definitions is that the vehicle will have an easier time collecting target points in places where the agility is larger; and we are interested in the maximum value of the configuration-space agility function because the vehicle can position itself in such a way as to maximize this value.

These definitions form the basis for our theorems; between them (and the distribution $F$ of the target points over $\mathcal{X}$) they provide the bounds for the length of the optimal tour.

1.4.6 The TSP and ORNT Functions

We now formally define functions based on the TSP and Orienteering; our results will then be stated as bounding these functions in various scenarios. For these definitions, $X_1, X_2, \ldots, X_n$ are $n$ points in $\mathcal{X}$, and $\lambda \in \mathbb{R}_+$. 

Definition 13 (The Traveling Salesman Problem).

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

Definition 14 (The Orienteering Problem).

\[
\text{ORNTr}(X_1, X_2, \ldots, X_n; \lambda) \triangleq \max_{\pi \in \Pi_\lambda}(\# \cap \{|X_i\}_{i=1}^n).
\]

In short, $\text{TSP}_\Pi(X_1, X_2, \ldots, X_n)$ is the length of the shortest path satisfying $\Pi$ which passes through $X_1, X_2, \ldots, X_n$, and $\text{ORNTr}(X_1, X_2, \ldots, X_n; \lambda)$ is the maximum number of points from $X_1, X_2, \ldots, X_n$ which lie on the same length-$\lambda$ path satisfying $\Pi$.

1.5 Assumptions and How to Deal with Them

In this section, we give a high-level view of the assumptions which are needed for our results to hold. Some of these we simply assume to be true; others can be bent a little bit or approximated (in particular, the requirement of the workspace to be “flat”, and the requirement of the target point distribution to be uniform). We will later show that a very general definition of vehicle dynamics satisfies these assumptions.

First, we would like to make a few very basic assumptions regarding the portion of the space in which the targets are distributed and imposing a speed-limit for the vehicle (as measured in $\mathbb{R}^d$).

Assumption 2 (Speed Limits). Any $\pi \in \Pi$ is Lipschitz continuous with parameter 1.

Assuming parameter 1 was merely for convenience; any (positive) constant would work, because we can simply re-scale the time dimension to make it 1-Lipschitz.

Assumption 3 ("Teleportation"). There exists a constant $M \in \mathbb{R}_{>0}$ such that for any $z_1, z_2 \in \mathcal{Z}$, there exists some $\pi \in \Pi$ where $\ell(\pi) \leq M$ and $\pi$ goes from $z_1$ to $z_2$.

We refer to this as the "teleportation assumption" because we will primarily use it to ‘teleport’ the vehicle from one configuration to another for a fixed constant cost $M$ (the cost is actually $\leq M$, we are just simplifying here). Note that this, along with the speed limit, also implies that $\mathcal{Z}$ itself is bounded (as otherwise we could choose points in $\mathcal{Z}$ which are arbitrarily far apart from each other), which by definition means that $\mathcal{S}$ is bounded and $\mathcal{X}$ is bounded.

1.5.1 The Spaces $\mathcal{X}$ and $\mathcal{Z}$ (a.k.a. Traveling to Flatland)

Because our method involves partitioning the space into rectangular “cells” and “translating” trajectories, dealing with curved manifolds is unpleasant. Therefore, we would specifically like the work- and configuration spaces to be flat, i.e. $\mathcal{X} = \mathbb{R}^{d_x}$ and $\mathcal{Z} = \mathbb{R}^{d_z}$.

In particular, we make the following rather strong assumptions:

- $\mathcal{X} = \mathbb{R}^{d_x}$ (i.e. it is flat);
• $\Pi$ is invariant under translation in $\mathcal{X}$, i.e. for any $v \in \mathbb{R}^{d\mathcal{X}}$ if $\pi$ is a valid trajectory, then $\pi'$ is too where $\pi'(t) = \pi(t) + v$.

Of course, these assumptions sometimes simply don’t hold. For example, the “connect-the-darts” case of the Dubins Car has a curved workspace (specifically, $\mathcal{X} = \mathbb{Z} = \mathbb{R}^2 \times S^1$). To fix this problem, we break the configuration space up into small ‘regions’; then, using the fact that it is a manifold of bounded curvature, we project these ‘regions’ onto a tangent plane and consider the dynamics on that plane; we refer to this as the “flattening” process, which will be discussed fully in Section 2.3. Given the proper assumptions (which, unlike the strict flatness requirement, cover all commonly-studied vehicles), the flattening process is a close enough approximation to allow us to still derive our results.

The reason we can employ this trick is that the length of the tour depends overwhelmingly on how efficiently the vehicle can visit closely-packed target points (the same reason that the volume of the reachable sets over small times is the most important feature). Therefore, we can consider very small regions - which are approximately flat - in isolation.

1.5.2 Assumptions about the Target Points

We denote the $n$ points which we wish to visit as $X_1, X_2, \ldots, X_n$; each point is independently and identically distributed over $\mathcal{X}$ according to some distribution $F$.

Definition 15 (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 $\mathcal{Z}_F \triangleq [\mathcal{X}_F]^2$, i.e. the configurations which project onto the support of $F$.

To make this work simpler, when dealing with general vehicles we will be assuming that the target points are uniformly distributed in a full-dimensional unit hypercube $\mathcal{X}_F \in \mathcal{X}$ (having assumed that $\mathcal{X}$ is flat, this is well-defined). We can make this assumption hold good for general vehicles by dividing the space into finitely many regions in which the distribution is approximately uniform, and then working on each region separately. If we are being very careful, we note that this means the number of target points distributed into any particular region is no longer a fixed $n$ but a random variable; this is not terribly difficult to deal with, and we give the formal reasoning in Appendix B.

However, for symmetric vehicles our results are more precise and depend on the distribution of the target points. Instead we assume that $F$ satisfies the following:

• $F$ has bounded support;

• $F$ is a full-dimensional continuous distribution, with piecewise-lipschitz-continuous density function $f$ (where the curvature of the boundary of those pieces is also bounded);

• $f$ is bounded above (i.e. $F$ has a bounded density).

Our results for general vehicles hold under these assumptions as well, for the reasons given above.

1.5.3 Vehicular Constraints and Assumptions

We now describe a set of conditions which are used directly in the derivation of our results. A very broad and basic category of dynamic constraints satisfy these conditions, or are sufficiently good approximations that our methods will still work, and we will discuss this in the next chapter. If $\Pi$ satisfies these constraints, we refer to it as vehicular.

Assumption 4 (The Small-Time Reachability Condition). There exists $\gamma \in \mathbb{Z}_{>0}$, $0 < c^\text{less}_i < c^\text{more}_i$ for $i \in [d\mathcal{X}]$, an integer vector $\bar{\gamma} \in \mathbb{Z}^{d\mathcal{X}}_{>0}$ such that, for every $z \in \mathcal{Z}$ and $\epsilon > 0$:

• $Q^\text{less} \subseteq \mathcal{R}(z) \subseteq Q^\text{more}$ where $Q^\text{less}$ and $Q^\text{more}$ are full-dimensional rectilinear sets with edge-lengths $c^\text{less}_i e^\bar{\gamma}_i$ and $c^\text{more}_i e^\bar{\gamma}_i$ respectively (where $i$ goes from 1 to $d\mathcal{X}$).

• $\min_{i \in [d\mathcal{X}]} \bar{\gamma}_i = 1$. 
1.6. SUMMARY OF MAIN RESULTS

- \[ \sum_{i=1}^{d_x} \gamma_i = \gamma. \]

Additionally, we assume that at every \( z \in \mathcal{Z} \), the quantity

\[ g(z) = \lim_{\epsilon \to 0} \frac{\text{vol}(\tilde{R}_\epsilon(z))}{\epsilon^\gamma} \]

is well-defined. Furthermore, \( g(z) \) is assumed to be Lipschitz-continuous and bounded both above and below (away from 0).

Note that this assumption relies on the previous assumption that \( \mathcal{X} \) is flat in order to have a coherent definition of what a rectilinear set is. Note also that this implies that the volume of \( \tilde{R}_\epsilon(z) \) is proportional to \( \epsilon^\gamma \) as \( \epsilon \to 0 \). We also note that the second bullet point is not really necessary as if it is not true, then we can derive a contradiction (in that it makes it impossible for a motion of length \( \alpha \) from a configuration to the edge of its inscribed box to be built out of \( \alpha/\epsilon \) motions of length \( \epsilon \) if \( \epsilon < \ll \alpha \)); however, we include it here to keep things simple.

**Assumption 5** (Straight Flexible Trajectories). There exists some trajectory \( \pi \) such that:

- \( \pi \) is flexible;
- its projection \( \tilde{\pi} \) onto the workspace is straight;
- it satisfies a minimum-speed requirement, i.e. \( \| \tilde{\pi}(t) - \tilde{\pi}(t + \delta) \|_2 \geq c \delta \) for any \( \delta, t \) and some \( c \geq 0 \)

We now give some assumptions regarding how these reachable sets cover the space and one another.

**Assumption 6** (The Reachable-Set Coverage Condition). There exists \( b \in \mathbb{Z}_{>0} \) such that for all \( z \in \mathcal{Z} \) and (sufficiently small) \( \epsilon > 0 \), there exists \( z_1, z_2, \ldots, z_b \) (dependent on \( z, \epsilon \)) such that

\[ R_{2\epsilon}(z) \subseteq \bigcup_{i=1}^{b} R_{\epsilon}(z) \]

Note that this assumption is for reachability sets in \( \mathcal{Z} \), as opposed to Assumption 4, which is about the reachability sets in \( \mathcal{X} \).

We also need a similar condition regarding the support of the target points' distribution in the workspace. This is not an assumption, but a lemma.

**Lemma 1** (The Starting-Point Coverage Condition). There is a polynomial \( q \) such that, for any \( \epsilon > 0 \), there are configurations \( z_1, \ldots, z_{q(1/\epsilon)} \) such that

\[ \mathcal{Z}_F \subseteq \bigcup_{i=1}^{q(1/\epsilon)} R_{\epsilon}(z_i) \]

This is immediately implied by Assumption 4, the boundedness of \( \mathcal{Z}_F \), and the flatness assumption. We call this the starting-point coverage condition because what is really important is that the set of viable starting points for the vehicle is bounded and can be covered by a polynomially-growing (in \( 1/\epsilon \)) number of \( \epsilon \)-reachable sets (the vehicle can start anywhere but obviously there is no reason to start outside of \( \mathcal{Z}_F \)); in fact, our methods also cover the case where \( \mathcal{S} \) is unbounded but the vehicle is only allowed to start within a given bounded region.

### 1.6 Summary of Main Results

We now formally state our main results, whose proofs will be developed in the rest of this work. Each result on the TSP is accompanied by a 'companion' result for Orienteering; all the results consist of both a lower- and upper-bound. In general, we prove for each result the upper bound for both the TSP and Orienteering variants, and then prove the lower bound for each from the upper bound of the other.

In our terminology, an event occurs with very high probability if the probability that it doesn’t happen goes to 0 exponentially fast as \( n \to \infty \). Formally, we consider random events parameterized by a value \( n \) (for example, “out of \( n \) random coin flips, at least \( n/2 \) of them land on Heads”):
Definition 16 (Very High Probability). Let \( A(n) \) be a random event parametrized by the value \( n \). Then \( A(n) \) occurs with very high probability if there is some \( \xi > 0 \) such that for all sufficiently large \( n \),
\[
\Pr[A(n) \text{ doesn't happen}] \leq e^{-(n^\xi)}
\]

Note that the right-hand side shrinks faster than any inverse polynomial, which is why we say it occurs with 'very' high probability (in order to distinguish it from the case where the probability shrinks according to an inverse polynomial).

We also define big-\( \Theta \) notation as:

Definition 17 (Big-\( \Theta \)). For any functions \( a(n) \) and \( b(n) \), we say that \( a(n) = \Theta(b(n)) \) if there exist positive constants \( c_1, c_2 \) such that for sufficiently large \( n \),
\[
c_1 b(n) \leq a(n) \leq c_2 b(n).
\]

In certain situations, we will make additional conditions on the constants, specifically that they don’t depend on other parameters of the problem; whenever this happens we will explicitly state it.

1.6.1 Results for General Vehicles

As mentioned in Section 1.3, our results for general vehicles concern the asymptotic properties of the Stochastic TSP and Stochastic Orienteering as \( n \to \infty \).

Theorem 1 (Bounds on TSP\( \Pi \)). Let \( \Pi \) be a vehicular set of constraints, and \( F \) be a probability distribution over \( \mathcal{X} \) satisfying the assumptions given in Section 1.5. Then, if \( X_1, X_2, \ldots, X_n \in \mathcal{X} \) are identically and independently distributed according to \( F \),
\[
\text{TSP}_{\Pi}(X_1, X_2, \ldots, X_n) = \Theta(n^{\frac{2\xi-1}{\gamma}})
\]
with very high probability.

Theorem 2 (Bounds on ORNT\( \Pi \)). Let \( \Pi \) be a vehicular set of constraints, and \( F \) be a probability distribution over \( \mathcal{X} \) satisfying the assumptions given in Section 1.5, and let \( \lambda > 0 \). Then, if \( X_1, X_2, \ldots, X_n \in \mathcal{X} \) are identically and independently distributed according to \( F \),
\[
\text{ORNT}_{\Pi}(X_1, X_2, \ldots, X_n; \lambda) = \Theta(\lambda n^{\frac{2}{3}})
\]
with very high probability.

1.6.2 Results for Symmetric Vehicles

We now present the strengthened results for symmetric vehicles. These strengthened results come in two forms: (1) more precise articulation for how the constant factor depends on the target point distribution and the agility function, and (2) an entirely new set of theorems bounding the worst-case point placement. For our standard results, we have something that looks suspiciously similar to the Beardwood-Halton-Hammersley Theorem, and is in fact a generalization of it:

Theorem 3 (Stronger Bounds on TSP\( \Pi \) for Symmetric Vehicles). Let \( \Pi \) be a symmetric set of dynamical constraints and \( F \) be a distribution on \( \mathcal{X} \) (with density function \( f \)) satisfying the assumptions in Section 1.5. Let \( X_1, X_2, \ldots, X_n \sim F \) independently. Then:
\[
\text{TSP}_{\Pi}(X_1, X_2, \ldots, X_n) = \Theta\left(n^{\frac{2\xi-1}{\gamma}} \int_{\mathcal{X}} f(x)^{\frac{2\xi-1}{\gamma}} g(x)^{-\frac{1}{\gamma}} dx \right)
\]
with very high probability.

In order to describe worst-case results, it no longer makes sense to discuss the points \( X_1, \ldots, X_n \) as being randomly-distributed; instead, we assume that an adversary is distributing them in order to maximize \( \text{TSP}_{\Pi}(X_1, \ldots, X_n) \) and/or minimize \( \text{TSP}_{\Pi}(X_1, \ldots, X_n; \lambda) \). However, in order to avoid a situation in which the adversary simply places the target points arbitrarily far away, we specify a (full-dimensional and bounded) region \( \mathcal{X}^* \subseteq \mathcal{X} \) in which they are allowed to place the points.
1.6. SUMMARY OF MAIN RESULTS

**Theorem 4 (Worst Case TSP₁ for Symmetric Vehicles).** Let \( \Pi \) be a symmetric vehicular constraint set. Then for any full-dimensional \( X^* \subseteq \mathcal{X} \),

\[
\sup_{x_1, \ldots, x_n \in X^*} \text{TSP}_\Pi(X_1, X_2, \ldots, X_n) = \Theta\left( n^{2/3} \left( \int_{X^*} g(x)^{-1}dx \right)^{1/3} \right).
\]

The constant factors implied by the big-\( \Theta \) notation in this case are not only not dependent on \( n \), they aren't dependent on \( F \) (for the first result) or on \( X^* \) (for the second result). We also have the equivalent results for the Orienteering problem:

**Theorem 5 (Stronger Bounds on ORNT₁ for Symmetric Vehicles).** Let \( \Pi \) be a symmetric set of dynamical constraints satisfying the assumptions in Section 1.5, and let \( F \) be a full-dimensional probability distribution over \( \mathcal{X} \) which is bounded and has a bounded density function \( f \). Let \( X_1, X_2, \ldots, X_n \sim F \) independently. Then:

\[
\text{ORNT}_\Pi(X_1, X_2, \ldots, X_n; \lambda) = \Theta(\lambda n^{1/3} \sup_{x \in \mathcal{X}} (f(x)g(x))^{1/3})
\]

with very high probability.

**Theorem 6 (Worst Case ORNT₁ for Symmetric Vehicles).** Let \( \Pi \) be a symmetric vehicular constraint set. Then for any full-dimensional \( X^* \subseteq \mathcal{X} \),

\[
\min_{x_1, \ldots, x_n \in X^*} \text{ORNT}_\Pi(X_1, X_2, \ldots, X_n; \lambda) = \Theta\left( \lambda n^{1/3} \left( \int_{X^*} g(x)^{-1}dx \right)^{-1/3} \right)
\]

(this is guaranteed).
CHAPTER 1. INTRODUCTION AND PRELIMINARIES
Chapter 2

Dynamic Constraints and Vehicular Motion

In this chapter, we define and explore some of the notions underlying our model and analysis. Many of the results and assumptions described here are taken from Chapters 2 and 3 of Itani's thesis [10] (and some of it re-cast in our own terminology). We note that our results can be generalized from the assumptions given here if these assumptions still hold piece-wise over the space (since the vehicle can deal with each piece of the space separately).

Though we will later draw heavily on the results and insights provided in this section, we will not revisit the elementary notions discussed here. As a consequence, the notation used in this section clashes somewhat with the notation in other sections in that some symbols may mean different things here than in other sections. This is intended to allow us to adhere somewhat to the notation used by Itani. However, we make an effort to clearly identify when the notation differs, and also to keep the notation for any objects which are used both here and in the rest of this work consistent (for example, the workspace is denoted by $\mathcal{X}$ both here and elsewhere).

There are also additional topics we do not address, such as how $\gamma$ comes out of the dynamic system; we use it as a black-box. This topic is addressed in Chapter 3 of Itani's thesis [10].

2.1 Notation and Space

Because this chapter is notation-heavy, we give a comprehensive list of our notation. However, we first need to establish the basic spaces we are working with.

2.1.1 The Workspace and Configuration Space

Recall that $\mathcal{X}$ (the workspace) is the space through which our vehicle moves (and in which the points it is supposed to visit are distributed). As mentioned in the introduction, we define the state of the vehicle as a point in what we call the configuration space $\mathcal{Z}$, which denotes not only its position in $\mathcal{X}$ but all information which determines where it can go in the future. In particular, we will assume that the configuration space can be represented as the location of the vehicle in the workspace and additional information from a manifold $\mathcal{S}$, i.e. $\mathcal{Z} \equiv \mathcal{X} \times \mathcal{S}$. We note briefly that this is actually slightly less general than the set-up used by Itani, who defines a general “output function” mapping $\mathcal{Z}$ to $\mathcal{X}$, because our ‘output function’ is by definition simply the projection function from $\mathcal{Z}$ to $\mathcal{X}$.

Example (Dubins Car). For the Dubins Car in 2 dimensions, we have $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{S} = S^1$, where $S$ encodes the current heading of the vehicle. More generally, for the Dubins Car in $d$ dimensions, $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{S} = S^{d-1}$ (where $S^k$ denotes the $k$-dimensional sphere, i.e. $S^k = \{v \in \mathbb{R}^{k+1} : \|v\|_2 = 1\}$).
2.1.2 Notation

We now give some notation which will be employed for the remainder of this chapter:

- $\mathcal{Z} = \mathcal{X} \times \mathcal{S}$ and all three are manifolds with bounded curvature, and $\mathcal{S}$ is bounded;
- $x$ (and variations like $x'$) will always signify a point in $\mathcal{X}$; $z$ (and variations) will always signify a point in $\mathcal{Z}$;
- $\mathcal{X}, \mathcal{S}, \mathcal{Z}$ are manifolds of dimension $d_X, d_S, d_Z$, respectively (and by definition $d_Z = d_X + d_S$);
- $\mathcal{X}, \mathcal{S}, \mathcal{Z}$ are represented as being embedded in Euclidean spaces of dimension $d_X^*, d_S^*, d_Z^*$, respectively (and by definition $d_Z^* = d_X^* + d_S^*$);
- $z = (z_X, z_S)$ where $z \in \mathcal{Z} \subseteq \mathbb{R}^{d_Z}$, $z \in \mathcal{X} \subseteq \mathbb{R}^{d_X}$, and $z \in \mathcal{S} \subseteq \mathbb{R}^{d_S}$;
- for $x \in \mathcal{X}$, $H_x \subseteq \mathbb{R}^{d_X}$ is the hyperplane tangent to $\mathcal{X}$ at $x$;
- for $x, y \in \mathcal{X}$, $[y]_{H_x}$ denotes the projection of $y$ onto $H_x$;
- for $x \in \mathcal{X}$ and $z \in \mathcal{Z}$, $[z]_{H_x} = ([z_X]_{H_x}, z_S) \in H_x \times \mathcal{S}$ (projecting only the $\mathcal{X}$-component);
- for $x \in \mathcal{X}$ and $\tau > 0$, $\hat{B}_x(\tau)$ is the ball of radius $\tau$ around $x$ in $\mathcal{X}$, and $B_{H_x}(\tau)$ is the projection of $\hat{B}_x(\tau)$ onto $H_x$. Note that because $\mathcal{X}$ has a bounded curvature, as $\tau$ becomes small, $B_{H_x}(\tau)$ approximates the ball of radius $\tau$ about $x$ on $H_x$;
- for $z \in \mathcal{Z}$ and $\tau > 0$, $B_z(\tau)$ is the ball of radius $\tau$ around $z$ in $\mathcal{Z}$.

When defining a ball of radius $\tau$ about a point $y$ in a curved manifold, we simply consider all points which can be reached from $y$ by a path of length $\tau$ or less within the manifold with the Euclidean length metric.

2.2 Vehicles and Control

We will be using a model of vehicle control which is similar to Itani's, but slightly simplified. The most important feature of this model is that it is affine in control with bounded control inputs; this category covers almost all commonly-studied vehicles but is simple enough that strong results about its behavior are known.

2.2.1 Controllability

In this model, the velocity of the vehicle is determined by its configuration (which for vehicles such as cars includes information like heading and speed) and bounded control inputs, represented as values from $[-1, 1]$. These control inputs affect the rate of change of the state of the vehicle in an additive manner.

**Definition 18 (Affine in Control).** A vehicle is affine in control if there exist functions $g_0, g_1, \ldots, g_m : \mathcal{Z} \to \mathbb{R}^d$ such that any valid trajectory $\pi : [0, T] \to \mathcal{Z}$ corresponds (not necessarily uniquely) to some input $u : [0, T] \to [-1, 1]^m$ such that for all $t \in [0, T]$,

$$\frac{d\pi(t)}{dt} = g_0(\pi(t)) + \sum_{i=1}^m g_i(\pi(t)) \cdot u_i(t)$$

Note that for any $z \in \mathcal{Z}$, $g_i(z)$ is in the tangent space of $\mathcal{Z}$ at $z$ (otherwise, we could set all inputs except $u_i$ to 0 and cause the vehicle to leave $\mathcal{Z}$, which is not allowed). Note also that we are abusing notation by using $g_i$ (as we already used $g$ for the agility function).

Note also that if $g_0(z) = 0$ for all $z \in \mathcal{Z}$, the vehicle is symmetric, as reverse trajectories become easy to construct. Formally, a trajectory $\pi$ (with control inputs $u$) going from time 0 to $T$ can be reversed by using the trajectory $\pi'$ (and control inputs $u'$) going from 0 to $T$ where $\pi'(T) = 0$ and $u'(t) = -u(T-t)$. The
2.2. VEHICLES AND CONTROL

g_0 term is called the drift, and symmetric dynamics are also often referred to as driftless. We note that by symmetry, if the dynamic system is driftless then \( z \) is at the center of \( R_z(z) \).

We now state some basic assumptions (taken from Chapter 3 of Itani's thesis):

**Assumption 7.** The functions \( g_0, \ldots, g_m \) satisfy the following:

- they are analytic;
- the integral curves of the functions are well-defined;
- (for now) \( g_0 = 0 \), i.e. the system is symmetric.

The last assumption is temporary. The idea is simply to allow an understanding of how the control inputs can move the vehicle in various directions - in particular, allowing us to approximate the region reachable with a short trajectory by a box with certain dimensions. We then add in the drift later, which mostly has the effect of translating the box (so that it is no longer centered around the current configuration, but around the point which drift would carry the vehicle to).

We also add the following two new assumptions:

**Assumption 8** (Lipschitz Continuity). The functions \( g_0, \ldots, g_m \) are Lipschitz-continuous (for simplicity we will assume a Lipschitz constant of 1, but by re-scaling we can use any constant).

**Assumption 9** (Lower Speed Limit in \( \mathcal{X} \)). For any \( z \in Z \): 1 \( \leq \max_{u \in [-1,1]^m} \left\| [g_0(z) + \sum_{i=1}^{m} g_i(z) u_i] \right\|_2 \)

Effectively, this assumption states that the vehicle can always move at a speed of at least 1 through \( \mathcal{X} \), given the proper input \( u \). We use 1 instead of a generic constant \( c \) for simplicity.

Finally, we have one last general assumption: an upper speed limit on how fast the vehicle can change its configuration.

**Assumption 10** (Upper Speed Limit in \( \mathcal{S} \)). There is some constant \( c \) such that for any \( \pi \in \Pi \) and any \( t \),
\[
\left[ \frac{d\pi(t)}{dt} \right]_S < c.
\]

2.2.2 Local Controllability

An important concept is the local controllability of a vehicle. Intuitively, a vehicle is small-time locally controllable at \( z \in Z \) if it can move in any direction (in \( Z \)) from \( z \). Formally, it is defined as:

**Definition 19** (Local Controllability). A system is small-time locally controllable if for all \( z \in Z \),
\[
z \in \text{Interior}(R_z(z)) \text{ for all } \varepsilon > 0
\]

Itani also gives the notion of output small-time locally-controllable, which is this but projected into \( \mathcal{X} \). This is necessary because Itani's work refers to a generic function which maps points from the configuration space into the workspace; since we are in a more specific setting in which this function is just projection, we don't require this notion.

**Definition 20** (Driftless Extension). For any vehicle with control dynamics governed by \( g_0, g_1, \ldots, g_m \) (as above), we define its driftless extension to be the same vehicle only with an extra control input \( u_0 \in [-1,1] \).

It is then defined by
\[
d\pi(t) = \sum_{i=0}^{m} g_i(\pi(t)) u_i.
\]

**Assumption 11** (Driftless Local Controllability). We assume the driftless extension of any \( \Pi \) we consider is small-time locally controllable.

This is a standard assumption which applies to all normally considered vehicles.
2.3 Simplification Tactics

Because the space of possible dynamic systems is so large and varied we would like to reduce to a simpler set of dynamics to study. In particular, we want the following simplifications:

1. A flat Euclidean $\mathcal{X}$ rather than a manifold, i.e. $\mathcal{X} = \mathbb{R}^{d_x}$.
2. A uniform target-point distribution over a hyper-cubic region of volume $1$.
3. $\Pi$ is translation-invariant in $\mathcal{X}$, i.e. if $\pi \in \Pi$, then $\pi' \in \Pi$ where $\pi'(t) = \pi(t) + (v, 0)$ for some fixed $v \in \mathcal{X}$ (and the $0$ is just the $0$-vector in $\mathcal{S}$).
4. There exists a straight, full-speed valid trajectory which is flexible.

We will also want to simplify the distribution of target points; in particular we will assume they are uniformly distributed over a hyper-rectangular region of area $1$ in $\mathcal{X}$.

2.3.1 Projected Dynamics

Working on manifolds - particularly, in a case where $\mathcal{X}$ is a manifold - causes various problems, and we would like to simplify to the case where $\mathcal{X}$ is Euclidean. We do this by working with only small pieces of the manifold at a time, where it is very close to being flat.

Since $\mathcal{X}, \mathcal{S}$ have bounded curvature, we can define a fixed $\tau^* > 0$ such that for any $\tau \leq \tau^*$, projecting from $B_2(\tau)$ onto $B_{H_x}(\tau)$ is a bijective function. We define an alternative set of ‘flat’ dynamics based around $x \in \mathcal{X}$ for such $\tau$ as follows:

**Definition 21 (Projected Dynamics).** A projected dynamic system of $\Pi$ at a point $x \in \mathcal{X}$ with radius $\tau$ is denoted $\Pi'_{x,\tau}$, and is defined so that every $\pi' \in \Pi'_{x,\tau}$ contained within $B_{H_x}(\tau)$ is associated with a path $\pi \in \Pi$ contained in $B_z(\tau)$ such that $\pi'(t) = [\pi(t)]_{H_x}$. We can define this as an affine-in-control system by the following. Let $g'_i : B_{H_x}(\tau) \times \mathcal{S}$ be the functions defining the control of $\Pi'_{x,\tau}$ as in Definition 18, and let $[\cdot]_{H_x}$ be the inverse of the projection from $B_{z}(\tau)$ onto $B_{H_x}(\tau)$ (since the projection is a bijection, the inverse exists). Then $g'_i(y) = [g_i([y]_{H_x})]_{H_x}$.

We had to be careful with this definition because of the inherent ambiguity created if we simply defined the projected dynamics as the regular dynamics projected onto $H_x$; specifically we had to restrict it so that only one point on $\mathcal{X}$ would project onto a given point in $H_x$ (that we were interested in).

Furthermore, for sufficiently small $\tau$, we get that the projected vectors $g'_i$ are basically the same as the vectors $g_i$ and so the reachable sets remain (asymptotically) the same size. We note that as long as we stay within the ball of radius $\tau$, we can execute the motion planning task in the flat region $B_{H_x}(\tau)$ rather than the curved one $B_z(\tau)$, which is one of the simplifications we had above.

Thus, our method breaks up $\mathcal{X}$ into radius-$\tau$ regions which can be very small but are fixed (so they don’t shrink or change as the number $n$ of target points grows to $\infty$). Each of these regions is dealt with separately; and traversing between regions might require trajectory of some length, but the total length devoted to this is constant with respect to $n$ - which means that as a fraction of the total length of the trajectory, traversing from region to region has negligible cost as $n$ goes to $\infty$.

We now need to show that we can work in the projected dynamics (locally) with the same assumptions:

**Lemma 2.** For a sufficiently small (fixed) $\tau$ and any $x \in \mathcal{X}$, the projected dynamics $\Pi'_{x,\tau}$:

- $\Pi'_{x,\tau}$ is affine in control and analytic, and any $\pi' \in \Pi'_{x,\tau}$ satisfies
  \[
  \frac{d\pi'(t)}{dt} = g_0(\pi(t)) + \sum_{i=1}^{m} g'_i(\pi(t)) \cdot u_i(t)
  \]
  for some control function $u$ (with output in $[-1, 1]$).
- the functions $g'_i$ are Lipschitz (not necessarily 1-Lipschitz)
2.4. MAIN LEMMAS

The point is that whatever controls we use in the projected dynamics will produce Of course, we are fudging the definitions a little, because we have not defined how to extend \( \Pi_{z, v} \) to the whole space.

Proof. The first condition follows simply from the fact that \( \pi' \) is a projection of \( \pi \) from \( Z = \mathcal{X} \times \mathcal{S} \) to \( H \times H \), and therefore we can take the same projection of \( g_1 \) to get \( g'_1 \). The second condition follows trivially from the fact that \( \mathcal{X} \) has bounded curvature. \( \square \)

To simplify things, we will assume that it remains 1-Lipschitz, even though projecting can change the Lipschitz constant. We can do this without affecting our theorems because for any \( \delta > 0 \), choosing a sufficiently small \( \tau \) will ensure that they will be \((1 + \delta)-\text{Lipschitz}\), and re-scaling things by a \( 1 + \delta \) factor does not affect any of our results.

2.4 Main Lemmas

Here we briefly discuss some main results which allow us to apply our method to a wide variety of different vehicles.

2.4.1 The Ball-Box Theorem

One of the main results we will rely on is the Ball-Box Theorem. This theorem bounds the reachable set \( \overline{R}_e(z) \) both inside and outside by boxes whose size is proportional to \( \epsilon^\gamma \). For this section, we are only considering driftless vehicles; however we will extend the most important notions to vehicles with drift as well.

Assumption 12 (Chow's Condition and Regularity). Our dynamic system II satisfies Chow's Condition and has no singular points (i.e. every \( z \in Z \) is regular).

These terms are taken from [3] and [11]. We will only describe these in general terms: Chow's Condition means that the vehicle can move freely over the whole configuration space, and does not somehow find itself unable to move in certain directions, and the lack of singular points means that the small-time behavior of the vehicle is consistent over the whole space, and \( \gamma \) is constant rather than dependent on the configuration. In short, this assumption is just that the dynamics governing the vehicle are well-behaved in a concrete manner.

Lemma 3 (Volume of \( \epsilon \)-Time Reachable Sets). For any dynamic system satisfying our assumptions, for any \( z \in Z \) there are parameters \( \gamma \leq \xi \in \mathbb{N} \) such that,

\[
\lim_{\epsilon \to 0} \text{vol}_Z(\overline{R}_e(z)) = \Theta(\epsilon^{\xi}) \quad \text{and} \quad \lim_{\epsilon \to 0} \text{vol}_X(\overline{R}_e(z)) = \Theta(\epsilon^\gamma)
\]

i.e. the volume of \( \overline{R}_e(z) \) in \( \mathcal{X} \) is proportional to \( \epsilon^\gamma \). In particular, its projection onto the tangent space \( H_\mathcal{X}(z) \) both contains a box of volume \( \Theta(\epsilon^\gamma) \) and is contained by a box of volume \( \Theta(\epsilon^\gamma) \) (though these boxes' volumes might have different constant parameters), and in fact the boxes have length \( \Theta(\epsilon) \) and \( (d_\mathcal{X} - 1) \)-dimensional base volume \( \Theta(\epsilon^{\gamma-1}) \).

Of course, the above implies "privileged" coordinates in which to orient the boxes. We will also make the assumption that the vehicle in some sense separates the workspace \( \mathcal{X} \) from \( \mathcal{S} \):

Assumption 13 (Workspace Separation). There exists a set of "privileged" coordinates in which the boxes described in Lemma 3 inscribed and circumscribed around \( R_e(z) \) in \( Z \) can be oriented, such that \( d_\mathcal{X} \) of them are tangent to \( \mathcal{X} \) and the remaining \( d_\mathcal{S} \) are tangent to \( \mathcal{S} \).

Note that this assumption means that we can inscribe and circumscribe the boxes in \( Z \) in such a way that every dimension will be either parallel or perpendicular to \( H_\mathcal{X} \).
2.4.2 Parallel Trajectories

We now want to show that restriction to small, flat regions yields another benefit - the dynamics are approximately translation-invariant, i.e. a valid path can be translated and (with minimal tweaks) remains a valid path. We do this by considering parallel trajectories which are those that can be generated by using the same control inputs from different starting points.

Recall that a trajectory \( \pi \in \Pi \) with length \( \ell(\pi) \) can be defined by specifying a starting point \( z = \pi(0) \in Z \) and a control function \( u : [0, \ell(\pi)] \rightarrow U \).

**Definition 22 (Parallel Trajectories).** Two trajectories \( \pi_1, \pi_2 \) are parallel if they are the same length and can both be associated with the same control function \( u \). We call them fully parallel if their starting points \( z_1, z_2 \) satisfy \( z_1 = (x_1, s) \) and \( z_2 = (x_2, s) \) where \( x_1, x_2 \in \mathcal{X} \) and \( s \in S \).

Note that this means the difference between \( \pi_1 \) and \( \pi_2 \) is caused by the difference in their starting points. This will be a useful notion because we will want to show that we can “cover” a certain region of space with a relatively uniform set of trajectories, and parallel trajectories can be shown to vary only slightly if their starting points are sufficiently close together.

**Lemma 4 (Staying on Track).** If \( \pi_1, \pi_2 \) are parallel with length \( \ell(\pi_1) = \ell(\pi_2) \) and \( \| \pi_1(0) - \pi_2(0) \| \leq \delta \), then for any \( L \leq \ell(\pi_1) \),

\[
\| (\pi_1(L) - \pi_2(L)) - (\pi_1(0) - \pi_2(0)) \| \leq (e^{(m+1)L} - 1)\delta
\]

This lemma establishes that short parallel trajectories that start out close to one another stay close to one another.

**Proof.** Let us define \( h : [0, L] \rightarrow \mathbb{R}^d \) and \( h_i : [0, L] \rightarrow \mathbb{R}^d \) for \( i = 0, 1, \ldots, m \) as

\[
h(t) = \pi_1(t) - \pi_2(t) \quad \text{and} \quad h_i(t) = g_i(\pi_1(t)) - g_i(\pi_2(t))
\]

Hence, our goal is to bound \( \| h(L) - h(0) \| \). From these definitions, we get

\[
\frac{d}{dt} h(t) = h_0'(t) + \sum_{i=1}^{m} u_i(t)h_i'(t)
\]

We then note that by our Lipschitz condition,

\[
\left\| \frac{d}{dt} h(t) \right\| \leq \| h_0'(t) \| + \sum_{i=1}^{m} \| u_i(t)h_i'(t) \| \leq (m + 1)\| h(t) \|
\]

Therefore, we can apply this to get (with an abuse of notation, for convenience):

\[
\| h(L) - h(0) \| \leq \int_{0}^{L} \left\| \frac{d}{dt} h(t) \right\| dt \leq \int_{0}^{L} (m + 1)\| h(t) \| dt
\]

We then note that this is upper-bounded by the solution to the dynamic equation

\[
\frac{d}{dt} y = (m + 1)y \quad \text{where} \quad y(0) = \| h(0) \|
\]

But that solution is simply \( y = \| h(0) \| e^{(m+1)t} \). Therefore, we can conclude that

\[
\| h(L) - h(0) \| \leq \| h(0) \| (e^{(m+1)L} - 1) \leq (e^{(m+1)L} - 1)\delta
\]

(by definition of \( h(0) \)) which is what we wanted. \( \square \)
2.5 Space-Covering Tilings

We now get to the main notion needed for our algorithms: space-covering tilings. The idea is to use the cells described in the ball-box theorem to cover the space. For general vehicles, the cells are those inscribed in the small-time deviation sets, while for symmetric vehicles it is those inscribed in the small-time reachable sets.

The radius $\epsilon$ of these cells (the radius of a cell is just the radius of the reachable or deviation set it is inscribed in, analogous to Definition 7) is set so that each cell covers $n^{-1}$ of the probability mass of $F$ (so that in expectation each cell contains 1 of the $n$ target points) -- which means in general that $\epsilon = \Theta(n^{-\frac{1}{2}})$. The target points are in collected by first going to the anchor of each cell (which again follows the terminology from Definition 7) and then proceeding from there to the target point.

The cells do not have to be rectilinear, but it is easier to think about and visually represent them if they are. Since we know by Lemma 3 that for any reachable set we can inscribe and circumscribe rectilinear sets whose dimensions differ only by constant factors, we will do so for convenience.

2.5.1 Space-Covering Trajectories

For the case of general vehicles, our space-covering tiling can really be described as “space-covering trajectories”, because the tilings are produced by deviation sets, which are necessarily defined by a trajectory. The idea is that a flexible trajectory travels with a “coverage tube” surrounding it -- the set of points which the vehicle can visit with a short deviation from the trajectory; we then choose a trajectory so that the whole region $\mathcal{X}_T$ in which the target points are distributed is covered by this “tube”. We also want such a trajectory to move at a rate lower-bounded by a positive constant, and to have a bounded curvature. Because the width of the “tube” changes with $\epsilon$, which changes with $n$, we need a method for constructing such a thing on any scale.

We do this by first finding a suitable flexible trajectory, and then making sufficient parallel copies to cover the space. We therefore need to show that (1) such a suitable trajectory exists; (2) trajectories fully parallel to it have the same properties (within the small ball that we are considering); (3) $\Theta(n^{\frac{2}{n-1}})$ of these parallel trajectories can cover the space. These parallel trajectories can then be joined end-to-end (using “teleportation” as defined by Assumption 3) to create a space covering trajectory of length $\Theta(n^{\frac{2}{n-1}})$.

Lemma 5 (Flexible Parallel Trajectories). For any dynamic system satisfying our assumptions, for sufficiently small $\tau > 0$ and constants $c_1, c_2, c_3 > 0$, $s \in S$, and fixed controls $u_1, \ldots, u_m \in [-1, 1]$ such that for all $x \in \mathcal{X}$ the following holds: for every $y \in B_x(\tau)$, let $\pi_y(t)$ be the trajectory starting from $z_y := (y, s) \in \mathcal{Z}$ with controls held at $u_1, \ldots, u_m$; then for all $y \in B_x(\tau)$ and all $t$ such that $\pi_y(t) \in B_x(\tau)$,

1. $\left\| \frac{d}{dt} \pi_y(t) \right\|_2 \geq c_1$ for all $t$;

2. $\pi_y(t)$ is $c_2$-flexible;

3. $\pi_y(t)$ has curvature at most $c_3$ at every point.

Remark: We will be playing loose with the constant factors in this proof -- everything is still correct, but we won’t make the constant factors as tight as they could be. Hence, there will be a lot of constants like $10^{-4}$ and $10^6$, which are used only so that things can be kept as simple as possible.

Proof. We first set $s \in S$ arbitrarily. We note by Assumption 10 that the vehicle cannot move through $S$ faster than some constant which we will refer to as $c^*$. We then note that for any fixed-control trajectory, the Lipschitz continuity condition (Assumption 8) guarantees that the curvature is bounded above (and in fact this bound is uniform over the whole space). Therefore, the third condition in the lemma is satisfied, with a fixed constant.

We also fix $c_1 = 0.0001$ (the exact quantity is not important, just that it is constant and sufficiently smaller than 1), and ensure that $\tau$ is sufficiently small so that any trajectory satisfying the curvature constraint and starting inside $B_x(\tau)$ must leave the ball before $t = 3\tau/c_1$. This ensures that we are only concerned with configurations in $B_{(x, s)}(c^*(3\tau/c_1) + \tau)$. This is because we are only interested when the position in $\mathcal{X}$ is at
most $\tau$ away from $x$, and we leave after at most $3\tau/c_1$ time, so we can only have traveled $c^*(3\tau/c_1)$ distance in $S$.

Therefore, for any $\delta > 0$, by setting a sufficiently small $\tau$, we can ensure by the Lipschitz property that over any two configurations in $B(x, \delta)(c^*(3\tau/c_1) + \tau)$ none have a difference in any $g_i$ of more than $\delta$.

We now break into two cases:

- $\|g_0(x, s)\|_2 \geq 0.01$
- $\|g_0(x, s)\|_2 \leq 0.01$

(again, the 0.01 isn’t important, we’re just using a constant that is much smaller than 1 and much larger than $c_1$).

In the first case, we will use a fixed control of $u_i = 0$ for all $i$. For a sufficiently small $\tau$, by the Lipschitz continuity condition (Assumption 8) $\|g_0(z)\|_2 \geq 0.05$ for all $z \in B(x, \delta)(c^*(3\tau/c_1) + \tau)$. Therefore, the first condition is met, and we have only the second condition left.

To examine this, we consider some small $\varepsilon$ and look at the trajectory $\pi_y$ generated by this. In particular, we define the reverse reachable set $Q_\varepsilon(z)$ and its workspace version $\hat{Q}_\varepsilon(z)$ as

$$Q_\varepsilon(z) := \{z' \in Z : z \in R_\varepsilon(z')\} \text{ and } \hat{Q}_\varepsilon(z) := [Q_\varepsilon(z)]_x$$

We note that the deviation set $D_\varepsilon(\pi_y, 0)$ satisfies

$$R_\varepsilon(\pi_y(0)) \cap Q_\varepsilon(\pi_y(\varepsilon)) \subset D_\varepsilon(\pi_y, 0)$$

(that is, the set of configurations which can be reached in $\varepsilon$ time from $\pi_y(0)$ and from which $\pi_y(\varepsilon)$ can be reached in $\varepsilon$ time must be contained in the deviation set).

We now consider the driftless extension of $\Pi$ (see Definition 20), and denote the reachable set under it as $\bar{R}_\varepsilon(z)$. We now consider the set

$$\bar{R}_{\varepsilon/4}(\pi_y(\varepsilon/2))$$

which we claim falls within both $R_\varepsilon(\pi_y(0))$ and $Q_\varepsilon(\pi_y(\varepsilon))$. This is because if we pick any $z \in \bar{R}_{\varepsilon/4}(\pi_y(\varepsilon/2))$, there must be a set of controls $u_0, \ldots, u_m$ over the time-span $t \in [0, \varepsilon/4]$ (including a control for the drift component $g_0$ of $\Pi$) which from $\pi_y(\varepsilon/2)$ arrives at $z$.

We then define for all $i = 0, 1, \ldots, m$

$$v_\varepsilon^* = \int_0^{\varepsilon/4} u_i(t)dt \in [-\varepsilon/4, \varepsilon/4]$$

(that is, the total input on control $i$ during the period $[0, \varepsilon/4]$) we can define

$$\varepsilon^* = \varepsilon/2 + v_0^* \in [\varepsilon/4, 3\varepsilon/4], \text{ and } u_i^* = v_i^*/\varepsilon^* \text{ for all } i \neq 0.$$
Therefore, we have proved in this case that $\mathbf{R}_{1/4}(\pi_\eta(\epsilon/2)) \subset D_\epsilon(\pi_\eta,0)$ and therefore that $\text{vol}_X D_\epsilon(\pi_\eta,0) = \Theta(\epsilon^\nu)$, so the trajectory is flexible. Since we can do this starting at any $t$, and not just $t = 0$, with the same constant factors, it is uniformly flexible, as we wanted.

In the second case, we note that by Assumption 9 (the lower speed limit), there is some fixed control $u^* \in [-1,1]^m$ such that
\[
\|g_0(z_x) + \sum_{i=0}^{m} u^*_i g_i(z_x)\| \geq 1
\]
(\text{where } z_x = (x,s), \text{as defined in the lemma}). Then, by Assumption 8 (Lipschitz continuity), for sufficiently small $\tau$ this implies that for all $z \in B(z_\eta,\epsilon)(\epsilon^*(3\tau/c_1) + \tau)$,
\[
\|g_0(z) + \sum_{i=0}^{m} u^*_i g_i(z)\| \geq 1/2 \text{ and } \|g_0(z)\| \leq 0.05
\]

We then use as our fixed control $u_i = u^*_i/2$. By the above, this means that
\[
\|g_0(z) + \sum_{i=0}^{m} u_i g_i(z)\| = \|g_0(z) + \sum_{i=0}^{m} (u^*_i/2) g_i(z)\| \geq 1/4 - 0.05 > 1/8
\]

Therefore, we have satisfied the requirement for the speed of our fixed-control trajectory. Therefore, since the third condition (on the curvature) was already dealt with at the beginning of the proof, only the flexibility remains.

However, we can consider such a trajectory to be a "drifting" trajectory with 0 control by redefining $g_0, g_1, \ldots, g_m$ to
\[
g^*_i = g_i/2 \text{ for all } i \neq 0, \text{ and } g^*_0 = g_0 + \sum_{i=1}^{m} (u^*_i/2) g_i
\]

We then note that since $u_1, \ldots, u_m \in [-1/2,1/2]$, if we have some controls $v_1, v_2, \ldots, v_m \in [-1,1]$, then the control
\[
g_0^*(z) + \sum_{i=1}^{m} v_i g^*_i(z)
\]
on the new dynamics can be recreated by control on the original dynamics (by using $v^*_i = u_i + v_i/2$). Therefore the deviation sets for the original dynamics must be at least as large as those for the new dynamics.

But the new dynamics are identical to the original dynamics except for the drift term and the fact that all the other terms have been scaled by a factor of 1/2. Therefore, the argument given above for the first case can be applied to prove that the given "drifting" trajectory on the new dynamics is flexible – and therefore the trajectory given for the original dynamics (having deviation sets at least as large) is also flexible. Therefore we are done. \qed

Remark: The reason we needed the two cases was because the fixed controls needed to be at different settings in the two cases to ensure that the vehicle is moving at a speed which is bounded below, as required by condition (1) in the lemma.

Lemma 6 (Space-Covering Trajectories). For any $x \in \mathcal{X}$, there exists constants $c_1, c_2, \tau > 0$ such that for every $\epsilon > 0$ there is a set of trajectories $\Phi_\epsilon$ such that:

1. $|\Phi_\epsilon| = \Theta(\epsilon^{-\nu})$;
2. $\Phi_\epsilon$ is uniformly $c_1$-flexible;
3. all trajectories in $\Phi_\epsilon$ have curvature at most $c_2$ at every point (uniformly bounded curvature);
4. $B_\tau(x) \subseteq \bigcup_{\pi \in \Phi_\epsilon} \tilde{D}_\epsilon(\pi)$ where $\tilde{D}_\epsilon(\pi) = \bigcup_t D_\epsilon(t,\pi)$. 
Proof. This is a direct consequence of Lemmas 4 and 5. The deviation sets of the trajectories defined by Lemma 5 have their longest dimension (the length of the inscribed boxes as defined in Lemma 3) in the direction they travel (we can make \( \tau \) sufficiently small so that these trajectories are effectively straight, since they have uniformly bounded curvature, and parallel by Lemma 4). Therefore, since the deviation sets have total volume \( \Theta(\epsilon^\gamma) \), their cross-sections have area \( \Theta(\epsilon^{\gamma-1}) \) (with inscribed boxes of this order of size as well). Therefore, we can tile \( \Theta(\epsilon^{-(\gamma-1)}) \) parallel trajectories in such a way that their deviation sets cover the whole ball. 

These trajectories will then be joined together to form a single “track”, using Assumption 3. The algorithm will then involve having the vehicle follow the track and deviate from it to visit target points.

### 2.5.2 Adjacent Tiling for Symmetric Vehicles

For symmetric vehicles, we will have an additional constraint, which is that the space should be tiled in such a way that adjacent tiles should be reachable from one another in a short time. Formally, we want the following:

**Lemma 7** (Adjacent Tiling). For any symmetric dynamic system satisfying our assumptions, there exists some \( \tau > 0 \) and constant \( c > 0 \) such that for every (sufficiently small) \( \epsilon > 0 \) and every \( x \in \mathcal{X} \), there exists configurations \( z_1, \ldots, z_{\epsilon r^d x g(x)^{-1} \epsilon^{-\gamma}} \) such that:

- \( \hat{B}_r(x) \subseteq \bigcup_{i=1}^{\epsilon r^d x g(x)^{-1} \epsilon^{-\gamma}} \hat{R}_e(z_i) \)
- if we let \( z_i, z_j \) be adjacent if \( R_e(z_i) \cap R_e(z_j) \neq \emptyset \), and define the graph \( G \) on vertices such that vertex \( i \) and \( j \) are adjacent if and only if \( z_i, z_j \) are adjacent, then \( G \) is a connected graph.

Note that in the second bullet point, the reachability sets used are in the configuration space \( \mathcal{Z} \).

**Proof.** This follows from the fact that the reachable set \( R_e(z) \) is centered at \( z \) when \( \Pi \) is symmetric. Let \( s \in S \) be such that \( \text{vol}_x \hat{R}_e(x, s) \approx g(x) \epsilon^\gamma \) as \( \epsilon \to 0 \). Then we know that if we select a sufficiently small \( \tau \), for all \( y \in \hat{B}_x(\tau), \text{vol}_x \hat{R}_e(x, s) \approx \text{vol}_x \hat{R}_e(y, s) \) due to Lemma 4. Furthermore, by Lemma 3 and Assumption 13 that we can inscribe boxes in the configuration-space reachable sets (at points of the form \( (y, s) \) where \( y \in \hat{B}_x(\tau) \) and \( s \) is as defined above) and project them down to the workspace, and lose only a constant factor of area.

Each tile has volume \( \Theta(g(x)\epsilon^\gamma) \), and the ball has volume \( \Theta(\tau^d x) \), and therefore we can complete the tiling with \( c\tau^d x g(x)^{-1} \epsilon^{-\gamma} \) tiles for some constant \( c \), as we wanted.

We now also note that because we used the cells inscribed in the \( \mathcal{Z} \) reachable sets \( R_e(y, s) \), when two cells overlap in \( \mathcal{X} \) they overlap in \( \mathcal{Z} \) as well. But this means that the adjacency graph, as defined in the lemma, is connected, finishing the proof. \( \square \)
Chapter 3

Results for General Vehicles

In this chapter, we formally prove our results for general vehicles, for both the TSP and ORNT functions. While most of the proofs here are self-contained, in Section 3.1.4 we will describe an abstraction of our problem (given the framework we are using) and use the solution to this abstraction as a black box. This is because the abstraction requires its own intricate analysis and requires notions that are not used in the rest of the work – which would make it difficult to read if it was in the middle of this section. Instead, we put those proofs in Appendix A.

The chapter is organized as follows: in Section 3.1, we give the algorithm which upper-bounds the TSP function; in Section 3.2 we give the probabilistic argument which upper-bounds the ORNT function; and we then use each these two upper bounds to prove the lower bounds for the other function.

3.1 Upper Bound of $\text{TSP}_{\Pi}$

**Proposition 1.** Let $\Pi$ be a set of vehicular constraints over $\mathbb{R}^d$. Then, if $X_1, X_2, \ldots, X_n$ are identically and independently distributed uniformly on $[0, 1]^d$, there exists a constant $a_1$ such that

$$\text{TSP}_{\Pi}(X_1, X_2, \ldots, X_n) < a_1 n^{\frac{3}{2}}$$

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 an efficient 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).

3.1.1 Tiling the Workspace

We will use the space-covering trajectory construction from Section 2.5.1, and describe here how to construct the proper tiling using the lemmas given there. We refer to the space-covering trajectory as the *track* and the points from which it makes its deviations as *waypoints*. Consider a straight flexible trajectory $\tau$ through $[0, 1]^d$, which we will assume without loss of generality to be parallel to one dimension of the hypercube (we can really consider the hypercube as circumscribed around a sphere in which our targets are really distributed, and orient it accordingly), and consider configurations $\tau(t), \tau(t + \epsilon) \in Z$ on this path. By Lemma 6 we know that such a $\tau$ exists (or rather that a sufficiently good approximation of it exists) and can be approximately translated to produce our tiling.

The tiling process works in 'corridors' which are each centered around a straight flexible trajectory $\tau_i$; the process for a given corridor $\tau$ is depicted in Figure 3.1. Consider waypoints $\tau(t), \tau(t + \epsilon), \tau(t + 2\epsilon), \ldots, $ and
CHAPTER 3. RESULTS FOR GENERAL VEHICLES

3. RESULTS FOR GENERAL VEHICLES

consider the ε-deviation sets at each of these configurations. Note that the vehicle can start at \( \tau(t) \), visit any chosen point in \( \tilde{D}_\epsilon(\tau, t) \) and make it to configuration \( \tau(t + \epsilon) \), visit any chosen point in \( \tilde{D}_\epsilon(\tau, t + \epsilon) \), and so forth - this means that in time at most twice that it takes to traverse \( \tau \), the vehicle can visit any collection of targets consisting of a single point in each ε-deviation set. By Lemma 3 (the Ball-Box Theorem), these sets contain boxes of length \( \Theta(\epsilon) \) and total volume \( \Theta(\epsilon^d) \); thus, the \((d-1)\)-dimensional volume of its base is \( \Theta(\epsilon^{d-1}) \). Since we want each of these cells to have volume \( 1/n \), we set \( \epsilon = \Theta(n^{-\frac{1}{d-1}}) \), so the base has \((d-1)\)-dimensional volume \( \Theta(n^{-\frac{2d-1}{d-1}}) \). Although there are places around \( \tau \) 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 ‘corridor’ around \( \tau \) (we can adjust the length of the boxes so that they perfectly cover the corridor with no overlap, as per Figure 3.1). This corridor has base volume \( \Theta(n^{-\frac{2d-1}{d-1}}) \) - meaning that to cover all of \([0,1]^d\) we need \( \Theta(n^{-\frac{2d-1}{d-1}}) \) corridors.

3.1.2 Collecting a Point from Each Cell

The vehicle can then traverse each corridor a constant number of times - which takes a constant amount of time per corridor since each spans \([0,1]^d\), and by assumption (3) from Section 1.5 traversing from the end of one corridor to the beginning of another takes a path of length at most \( M \) - and collect one point from each cell. Since there are \( \Theta(n^{-\frac{2d-1}{d-1}}) \) corridors in total, this means that we can visit one point in each cell with a path of length \( \Theta(n^{-\frac{2d-1}{d-1}}) \). We can think of this as removing a point from each cell; we call this the collection process.

It would then be convenient if we could simply repeat this process (each reset 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, \( \Theta\left(\frac{\log n}{\log \log n}\right) \) points, giving us only a \( \left(\frac{\log n}{\log \log n}\right) \)-approximation of the claimed lower bound.

Figure 3.1: A visual representation of the tiling process of a ‘corridor’ (of which there are \( \Theta(n^{-\frac{2d-1}{d-1}}) \)) at resolution \( \epsilon \); arrows included to indicate that the waypoints represent configurations. (a) Two points along straight flexible trajectory \( \tau \) at distance \( \epsilon \), 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 \( \tau \) at distance \( \epsilon \). (d) With a constant number of passes (depicted in red and green) through trajectory \( \tau \), these boxes cover a whole ‘corridor’ around \( \tau \).
3.1.3 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 3.2), preventing waste.

In particular, consider a cell generated by the method depicted in Figure 3.1, but with the scale $2\epsilon$ (again, as in Figure 3.2). In this case, cell will have length approximately double the length of a cell at scale $2\epsilon$, but its base will be $2^{-1}$ times the size. Thus, there will be only one corridor at this scale per $2^{-1}$ corridors at the original $\epsilon$ scale. Since the cost of removing a target point from every cell is proportional to the number of corridors, the cost of the collection process on cells at scale $2\epsilon$ is $2^{-\gamma-1}$ the cost of the collection process at scale $\epsilon$. Of course, there are fewer cells so the collection is coarser in addition to being cheaper; in particular, for every $2^2$ cells originally, there is only one new cell.

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.

3.1.4 A Combinatorial Representation of Collect-and-Merge

We can now use this notion of merging to move to an abstract version of the problem. We view the points we wish to collect as $n$ balls, which are distributed uniformly and independently at random into $n$ buckets, which represent the cells. We then wish to remove all the balls from the buckets at minimal cost using the following two operations:

- **Collection**: removes one ball from each (nonempty) bucket; this operation has a cost of $c$ (initially, $c = 1$).

- **Merge**: merges buckets by groups of $2^\gamma$ and decreases $c$ by a factor of $2^{\gamma-1}$; this operation has no cost.

Note that these are simply the operations described above, with a cost of 1 representing a path of length $\Theta(n^{\frac{d-1}{d}})$. The groups by which buckets are merged into larger buckets are fixed before the balls are distributed, and are known in advance. We want to show that, with very high probability, we can remove all the balls for a fixed constant cost.

Since one-dimensional problems are trivial, we can assume that $d \geq 2$, so $\gamma \geq 2$; we can also assume that $n = 2^{k+1}$ for some positive integer $k$ (for $n$ not satisfying this, we have at most a constant factor extra cost). This means that we can merge the buckets exactly $k$ times. We define a strategy for this problem as an algorithm which determines how many times to collect between each pair of consecutive merges; a strategy is **blind** if this number does not depend on the actual distribution of balls into the buckets. Note that a blind strategy has a deterministic cost, but may not successfully remove all the balls. We consider the following blind strategy (which we apply without observing how the balls are distributed):

**Strategy 1** (Exponential Moment Suppression). *Given any distribution for the balls into the buckets:*

- **Initialization**: collect once before any merges are made;

- **Main Stages**: collect $2^\gamma$ times before each subsequent merge, and after the last ($k$th) merge;
CHAPTER 3. RESULTS FOR GENERAL VEHICLES

- Cleanup: collect an additional \(2^{k(\gamma - 1)}\) times after the \(k\)th merge.

We refer to these three parts of the algorithm as its phases.

Its name (Exponential Moment Suppression) refers to its central idea, which will be discussed in Appendix A.1.4. Since Strategy 1 is blind, it has a fixed cost, which is:

- 1 for the initialization;
- \(2^\gamma \cdot (2^{-(\gamma - 1)} + 2^{-2(\gamma - 1)} + \cdots + 2^{-k(\gamma - 1)}) < 4\) (since \(\gamma \geq 2\)) for the main steps;
- \(2^{k(\gamma - 1)} \cdot 2^{-k(\gamma - 1)} = 1\) for the cleanup.

Thus, the total cost is less than \(1 + 4 + 1 = 6\).

We then need to analyze its probability of successfully removing all the balls. In particular, it succeeds with very high probability:

Lemma 8 (Strategy 1 Succeeds). Strategy 1 successfully removes all the balls with very high probability.

While this lemma has an extremely elegant proof, it requires a lengthy explanation and analysis, as well as a significantly different setup to the rest of this work. Therefore, we prove this result in the Appendix, specifically in Appendix A.1.4.

Thus, with very high probability, we can remove all the balls for a total cost of less than 6.

### 3.1.5 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 3.1.1.
2. Follow the Balls-and-Buckets strategy given in Section 3.1.4 and the Appendix:
   - (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 \(2^\gamma\), to produce a tiling as described in Section 3.1.1 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 \(\Theta(n^{\frac{2}{3}})\) that visits all \(n\) targets. We note that this algorithm requires a local planner, which we do not discuss, to work. The local planner does the following: for any \(\pi \in \Pi\), \(t \in [0, \ell(\pi)]\), \(\epsilon > 0\), and \(x \in D_c(t, \pi)\), it returns a valid path of length at most \(2\epsilon\) from \(\pi(t)\) to \(\pi(t + \epsilon)\), which passes through \(x\).

Remark: There is a small wrinkle with the fact that the "merges" become less idealized (less like perfect parallel cells) as they become bigger. However, this is fairly simple to deal with: we define a fixed level of coarseness, below which the merging works as described above. Then we execute the merge algorithm up to this fixed level. By the logic given above, the number of target points remaining is bounded above by a constant (with high probability), and we can then simply visit these points in any order for a constant extra cost (which becomes negligible as \(n \to \infty\)).

### 3.2 Upper Bound of ORNT\(_\Pi\)

Proposition 2. Let \(\Pi\) be a set of constraints, and \(f\) be a probability distribution over \(\mathcal{X}\), satisfying the assumptions given in Section 1.5, 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 \(\alpha_2\) such that

\[
\text{ORNT}_\Pi(X_1, X_2, \ldots, X_n; \lambda) \leq \alpha_2 \lambda n^{\frac{3}{2}}
\]

with very high probability.
We prove this by using a generalized version of the powerful probabilistic argument introduced by Arias-Castro et al [1] (who refer to it as Connect-the-Dots). It begins with a discretization of \( Z_F \) into cells with the following properties:

- the projection of each cell onto \( X \) 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 \( \lambda \) can be covered by a sequence of \( \Theta(n^{\frac{2\lambda-1}{\lambda}}) \) cells, beginning with a ‘starter’ cell and choosing a successor of the current cell at each subsequent step.

Then, as we will see, a probability argument from the Chernoff and Union bounds finishes the proof.

### 3.2.1 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 \( Z_{[0,1]^d} \equiv [0,1]^d \); this corresponds to the notion of \( Z_F \) as defined in Section 1.4.1. We now give some crucial lemmas.

**Lemma 9** (Polynomial Cover of \( Z_{[0,1]^d} \)). There exists a polynomial \( p \) such that for any \( \epsilon > 0 \), there exists a collection of \( p(1/\epsilon) \) configurations \( y_1^*, y_2^*, \ldots, y_{p(1/\epsilon)}^* \in \mathcal{Z} \) such that

\[
Z_{[0,1]^d} \subseteq \bigcup_{i=1}^{p(1/\epsilon)} R_{\epsilon}(y_i^*). 
\]

We refer to this set as \( Y_{\epsilon}^* \).

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

**Proof.** Consider the collection of straight, parallel flexible trajectories whose existence we assumed in Section 2.3; let \( \tau \) be any of these trajectories.

Then, by the Ball-Box Theorem, \( R_{\epsilon}(\tau(t)) \) contains a rectangular prism of volume \( \Theta(\epsilon^q) \); we note that in each direction, the length of the prism is \( \Omega(\epsilon^q) \) (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 \( \Omega(\epsilon^q) \), which itself contains a hypercube of edge length \( \Omega(\epsilon^q) \). 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 10** (Fixed-Size Cover of \( R_{2\epsilon}(z) \)). There exists a positive integer \( b \) such that, for any \( \epsilon > 0 \) and \( z \in \mathcal{Z} \), there exists a set of \( b \) configurations \( y_1, y_2, \ldots, y_b \in \mathcal{Z} \) such that

\[
R_{2\epsilon}(z) \subseteq \bigcup_{i=1}^{b} R_{\epsilon}(y_i). 
\]

In other words, any \( (2\epsilon) \)-reachable set can be covered by a fixed number of \( \epsilon \)-reachable sets. For any \( z \in \mathcal{Z} \) and \( \epsilon > 0 \), we refer to \( Y_{\epsilon}(z) := \{y_1, \ldots, y_b\} \) as the menu of \( z \).

**Proof.** We prove this with Lemma 3; we note that the Ball-Box Theorem still applies when \( X = \mathcal{Z} \) (i.e. the vehicle must be in a particular configuration to collect a target), though it produces a different exponent which we denote as \( \xi \). We note that by Lemma 3, \( R_{2\epsilon}(z) \) then is contained by a box of volume \( \Theta(\epsilon^\xi) \), while for any configuration \( y \), \( R_{\epsilon}(y) \) 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 \( R_{2\epsilon}(z) \) and the latter are all contained by \( R_{\epsilon}(y) \) for some \( y \), so \( R_{2\epsilon}(z) \) (for any \( z \in [0,1]^d \)) can be covered by the union of \( R_{\epsilon}(y) \) for boundedly many \( y \in \mathcal{Z} \).
3. RESULTS

FOR GENERAL VEHICLES

Figure 3.3: An illustration of the iterative generation of the discrete representation \( \psi \) of path \( \pi \). (a) The rule that \( \pi(\epsilon) \in R_\epsilon(\psi_i) \) (shown in green) applies, so it is clear that \( \pi((i+1)\epsilon) \in R_\epsilon(\psi_i) \). (b) By Lemma 10, \( R_\epsilon(\psi_i) \) can be covered by a set of \( \epsilon \)-reachability sets (depicted in blue); thus, one of these sets (depicted in red) contains \( \pi((i+1)\epsilon) \). Its starting point is chosen as \( \psi_{i+1} \), so \( \pi((i+1)\epsilon) \in R_\epsilon(\psi_{i+1}) \), allowing the process to continue. Note that this diagram is not 100% accurate because the represented \( \epsilon \)-reachable sets are supposed to be in \( Z \) and not \( X \); however, due to the limitations of 2-dimensional figures, we cannot show them properly in \( Z \).

3.2.2 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 \( \pi \) which begin at a configuration \( \pi(0) \in Z_{[0,1]^d} \). This is because if \( \pi(0) \not\in Z_{[0,1]^d} \), we can simply remove the beginning of \( \pi \) and get a shorter path collecting the same targets.

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

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

In other words, we first select a configuration \( \psi_0 \) from the covering set \( Y_\epsilon^* \) of \( Z_{[0,1]^d} \) such that the starting point of \( \pi \) is \( \epsilon \)-reachable from \( \psi_0 \). We then iteratively select \( \psi_{i+1} \) from the menu \( Y_\epsilon(\psi_i) \), keeping the invariant that \( \pi(\epsilon) \) is \( \epsilon \)-reachable from \( \psi_i \). We now need to show that this representation exists.

By Lemma 9, since \( \pi(0) \in Z_{[0,1]^d} \), there is some \( \psi_0 \in Y_\epsilon^* \) such that \( \pi(0) \in R_\epsilon(\psi_0) \). We now inductively show that if \( \pi(\epsilon) \in R_\epsilon(\psi_i) \), we can select some \( \psi_{i+1} \in Y_\epsilon(\psi_i) \) such that \( \pi((i+1)\epsilon) \in R_\epsilon(\psi_{i+1}) \).

Suppose (by the inductive assumption) that \( \pi(\epsilon) \in R_\epsilon(\psi_i) \). Then, since \( \pi((i+1)\epsilon) \) is by definition \( \epsilon \)-reachable from \( \pi(\epsilon) \), we know that \( \pi((i+1)\epsilon) \in R_\epsilon(\psi_i) \). By Lemma 10, any configuration in \( R_\epsilon(\psi_i) \) is covered by an \( \epsilon \)-reachability set from some point in \( Y_\epsilon(\psi_i) \); we can then set \( \psi_{i+1} \) to be this point. See Figure 3.3 for illustration.

Thus, we can represent any \( \pi \in \Pi_\lambda \) by such a sequence of \( \lfloor \lambda/\epsilon \rfloor + 1 = \Theta(\lambda n^{1/3}) \) configurations. Furthermore:

\[
\pi \subset S(\psi) := \bigcup_{i=0}^{\lfloor \lambda/\epsilon \rfloor} R_\epsilon(\psi_i), \quad \text{so} \quad \hat{\pi} \subset \hat{S}(\psi) := [S(\psi)]_X = \bigcup_{i=0}^{\lfloor \lambda/\epsilon \rfloor} \hat{R}_\epsilon(\psi_i).
\]

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

However, there exist only finitely many representations. \( \psi_0 \) is chosen from \( Y_\epsilon^* \), 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 \( \epsilon = \Theta(n^{-2/3}) \), there is some polynomial \( p^* \) such that \( p(1/\epsilon) \leq p^*(n) \). Thus, with at most \( p^*(n) \) initial choices for \( \psi_0 \), and \( b \) choices at each of the subsequent \( \lfloor \lambda/\epsilon \rfloor = \Theta(\lambda n^{1/3}) \) steps, we have at most \( p^*(n) \cdot b^{\Theta(\lambda n^{1/3})} \) representations in total – so every valid path of length \( \lambda \) is covered by one of these representations. We refer to the set of possible representations \( \psi \) at resolution \( \epsilon \) as \( \Psi(\lambda, \epsilon) \).

We can then find a constant \( A_1 \) such that

\[
|\Psi(\lambda, \epsilon)| \leq p^*(n) \cdot b^{A_1 \lambda n^{1/3}} \leq (b + 1)^{A_1 \lambda n^{1/3}} \tag{3.1}
\]

(the second inequality holds for all sufficiently large \( n \)).
3.2.3 The Set Collection Problem

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

**Definition 23** (The Set Collection Problem).

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

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

**Lemma 11** (SCP \( \geq \) ORNT). For any \( \epsilon > 0 \),
\[
SCP(X_1, \ldots, X_n; \lambda, \epsilon) \geq ORNT(\Pi(X_1, \ldots, X_n; \lambda)).
\]

3.2.4 The Probability Argument for SCP

We can now prove Proposition 2. Note that for any \( \psi \in \Psi \),
\[
\text{vol}_{\hat{S}(\psi)} = O(\lambda n^{-\frac{3}{2}})
\]
as \( \hat{S}(\psi) \) is a union of \( \Theta(\lambda n^{-\frac{1}{2}}) \) (possibly overlapping) sets of the form \( \hat{R}_{2\epsilon}(\psi, i) \), each with volume \( \Theta(1/n) \).

Thus, the probability that a given point falls within \( \hat{S}(\psi) \) (which, since the targets are uniformly distributed at random, is equal to its volume) is at most \( O(\lambda n^{-\frac{3}{2}}) \); this means that for some constant \( A_2 \), for every \( \psi \in \Psi(\lambda, \epsilon) \) and any \( X_i \),
\[
Pr[X_i \in \hat{S}(\psi)] \leq A_2 \lambda n^{-\frac{3}{2}}.
\]

Thus,
\[
E[|\hat{S}(\psi) \cap \{X_1, \ldots, X_n\}|] \leq A_2 \lambda n^{-\frac{1}{2}}.
\]

This applies for any \( \psi \in \Psi(\lambda, \epsilon) \); we can then use the Union Bound on this and expression (3.1) to get
\[
Pr[\exists \psi \in \Psi(\lambda, \epsilon) : |\hat{S}(\psi) \cap \{X_1, \ldots, X_n\}| > B \cdot A_2 \lambda n^{-\frac{1}{2}}] \leq e^{-\frac{(B - 1)^2}{2} \lambda^{-\frac{1}{2}}}. A_2 \lambda n^{-\frac{1}{2}}. (b + 1) A_1 \lambda^{-\frac{1}{2}}.
\]

But we note that the log of this is
\[
-\frac{(B - 1)^2}{2} \cdot A_2 \lambda n^{-\frac{1}{2}} + \log(b + 1) \cdot A_1 \lambda^{-\frac{1}{2}} = \left( -\frac{(B - 1)^2}{2} \cdot A_2 + \log(b + 1) \cdot A_1 \right) \cdot \lambda^{-\frac{1}{2}},
\]

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 (3.2) goes to \( -\infty \). Thus, the probability that \( SCP(\lambda, \epsilon) > B A_2 \lambda n^{-\frac{1}{2}} \) goes to 0 as \( n \to \infty \), which by Lemma 11, implies Proposition 2 (with \( \alpha_2 = B A_2 \geq \sqrt{2A_1 A_2 \log(b + 1)} + A_2 \)), thus giving us our upper bound to the Orienteering problem.

### 3.3 Lower Bounds for TSP\(_\Pi\) and ORNT\(_\Pi\)

We now complete Theorems 1 and 2; we do this by showing the lower bound in Theorem 1 from the upper bound in Theorem 2 (i.e. Proposition 2) and vice versa.

**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 \( ORNT(\Pi(X_1, \ldots, X_n; 1)) \), which by Proposition 2 is with high probability at most \( \alpha_2 n^{\frac{1}{4}} \).
But this means, with very high probability,

\[ \frac{n}{2L} \leq \alpha_2 n^{\frac{1}{\gamma}} \implies L \geq \frac{2}{\alpha_2} n^{\frac{2\gamma-1}{\gamma}}, \]

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 \( \alpha_1 n^{\frac{\gamma-1}{\gamma}} \) which collects all the points. We then consider the subpath of length \( \lambda \) that collects the most points; since the average length-\( \lambda \) subpath collects at least

\[ \lambda \cdot \frac{n}{2\alpha_1 n^{\frac{\gamma-1}{\gamma}}} = \frac{\lambda}{2\alpha_1} \cdot n^{\frac{1}{\gamma}} \]

points, the 'densest' subpath collects at least \( \frac{\lambda}{2\alpha_1} n^{\frac{1}{\gamma}} \) 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 the techniques from this proof automatically suggest an algorithm for Orienteering as well: use our TSP algorithm to find a path through all \( n \) points, and then find the length \( \lambda \) sub-path which collects the most points. This achieves the bound given in Theorem 2 with very high probability.
Chapter 4

Results for Symmetric Vehicles

In this chapter, we prove our stronger results concerning symmetric vehicles. Since we now have two settings – one where the target points are distributed stochastically (i.i.d. according to distribution \( F \)) and the other where they are distributed adversarially – we will be proving four main propositions (an upper and a lower bound for each setting).

4.1 Vehicles with Symmetric Constraints

Consider the tiling algorithm described in Section 3.1. If the vehicle is symmetric, to visit more than one target point located in the tile contained by \( D_{\tau}(t) \), it doesn’t have to loop back to configuration \( \tau(t) \) every time: instead, beginning at \( \tau(t) \), it can pick up one point (with a trajectory of length at most \( 2\epsilon \)), then reverse back to \( \tau(t) \), and continue in this fashion until it picks up every point in the tile, at which point it can proceed to the next tile. This observation gives us the following result as a corollary of Theorem 1:

**Corollary 1** (Guaranteed Upper-Bound for TSP with a Symmetric Vehicle). Let \( \Pi \) be a symmetric dynamical system satisfying the assumptions made in Section 1.5, and let \( \mathcal{X}_{\text{target}} \subseteq \mathcal{X} \) be bounded and \( \Pi \)-bounded (as per the definition given in assumption 3 in Section 1.5). Then, for any \( X_1, X_2, \ldots, X_n \in \mathcal{X}_{\text{target}} \)

\[
\text{TSP}_\Pi(X_1, X_2, \ldots, X_n) = O(n^{\frac{3}{2} - \frac{1}{\gamma}}).
\]

**Proof.** We first note that we can still apply the simplifications detailed in Section 2.3 without losing the generality of the result. We then use the tiling given in Section 3.1.1, and travel from cell to cell as described. However, in each cell we make use of the symmetry property to collect all the target points without the need to make multiple visits. Consider the cell contained in \( D_{\tau}(t, \tau) \); the cell-to-cell traveling procedure ensures we arrive at \( \tau(t) \). For any target \( X_i \) in this cell, we can visit \( X_i \) from \( \tau(t) \) with a trajectory of at length most \( 2\epsilon \). We can thus visit each target in turn, returning to \( \tau(t) \) each time, with a trajectory of length at most \( 4\epsilon = \Theta(n^{-\frac{1}{2}}) \) per target point (\( 2\epsilon \) to visit \( X_i \) and \( 2\epsilon \) to return to \( \tau(t) \)). As detailed in Section 3.1.1, traveling through all the \( \tau \) in our collection (in order to visit every cell) takes a trajectory of length \( \Theta(n^{\frac{3}{2} - \frac{1}{\gamma}}) \); add to that \( \Theta(n^{-\frac{1}{2}}) \) for each of \( n \) target points, and we get a TSP tour of length \( \Theta(n^{\frac{3}{2} - \frac{1}{\gamma}}) \). Thus, the shortest tour has length \( O(n^{\frac{3}{2} - \frac{1}{\gamma}}) \).

We can extend the results for symmetric vehicles further by investigating how \( \text{TSP}_\Pi(X_1, X_2, \ldots, X_n) \) relates to the distribution \( F \) from which the target points are drawn; for nonsymmetric vehicles, it is easy to see that no simple relationship holds (see Appendix C).

Note that we can view this as being about an 'unconstrained vehicle' (which is symmetric). We can then extend this result to all symmetric vehicles. We restate Theorem 3:

**Theorem** (Stronger Bounds on TSP\(_\Pi\) for Symmetric Vehicles). Let \( \Pi \) be a symmetric set of dynamical constraints and \( F \) be a distribution on \( \mathcal{X} \) (with density function \( f \)) satisfying the assumptions in Section 1.5.
Let \( X_1, X_2, \ldots, X_n \sim F \) independently. Then:

\[
\text{TSP}_\Pi(X_1, X_2, \ldots, X_n) = \Theta \left( n^{\frac{2-\gamma}{d}} \int_X f(x)^{\frac{2-\gamma}{d}} g(x)^{-\frac{1}{d}} \, dx \right)
\]

with very high probability.

We also note its obvious similarity to the Beardwood-Halton-Hammersley Theorem (which we will refer to as the BHH Theorem from now on), which was shown in 1959 [2]:

**The Beardwood-Halton-Hammersley Theorem.** Let \( F \) be a probability distribution over \( \mathbb{R}^d \) with bounded support and density function \( f \), and let \( X_1, X_2, \ldots \sim F \) independently. Let \( \text{TSP}(X_1, X_2, \ldots, X_n) \) denote the length of the shortest tour through \( X_1, X_2, \ldots, X_n \) by a continuous path (with no dynamic constraints). Then there exists a constant \( \beta_d \) (depending only on the dimension \( d \) of the space) such that

\[
\lim_{n \to \infty} \frac{\text{TSP}(X_1, X_2, \ldots, X_n)}{n^{d-1}} = \beta_d \int_{\mathbb{R}^d} f(x)^{\frac{d-1}{d}} \, dx
\]

with probability 1.

If we consider the subject of this theorem as a “dynamic vehicle” without any constraints (other than that its velocity must be at most 1), we can quickly note that \( \gamma = d \) and \( g \) is constant over the whole space. However, Theorem 3 is not quite a full extension, because we do not show that the ratio between \( \text{TSP}_\Pi(X_1, X_2, \ldots, X_n) \) and \( n^{\frac{2-\gamma}{d}} \int_X f(x)^{\frac{2-\gamma}{d}} g(x)^{-\frac{1}{d}} \, dx \) converges to a specific constant (we only show that it resides within a constant interval with very high probability).

**Remark:** Another thing to note is that the upper bound here is once again probabilistic – the only deterministic guarantees are in relation to the function \( g \) over the support of distribution \( F \) (as a deterministic guarantee by definition should have nothing to do with \( F \), except for the support \( X_F \)), which we will discuss in Section 4.5.

### 4.2 Lower-Bound for TSP in the Stochastic Setting

In this section, we prove the following proposition:

**Proposition 3.** Let \( \Pi \) be a symmetric set of dynamical constraints and \( F \) be a probability distribution (with density function \( f \)) on \( X \), both satisfying the assumptions in Section 1.5. Then there is a constant \( \beta^- > 0 \) such that if \( X_1, X_2, \ldots, X_n \sim F \) independently then:

\[
\beta^- \left( n^{\frac{2-\gamma}{d}} \int_X f(x)^{\frac{2-\gamma}{d}} g(x)^{-\frac{1}{d}} \, dx \right) \leq \text{TSP}_\Pi(X_1, X_2, \ldots, X_n)
\]

with very high probability.

Because the proof of this proposition requires the construction of a new version of Orienteering, we will not use the standard “proof environment”. However, the remainder of this section constitutes a formal proof. We will (as we later note) skip the final step of the proof, as it is completely identical to an earlier proof (for convenience we will cite exactly which proof we refer to).

#### 4.2.1 The Cost-Balanced Orienteering Problem

As before, we will prove Theorem 3 by considering an Orienteering problem (in much the same way that Theorems 1 and 2 were used to prove each other). However, in this case it is not quite enough to just use the Orienteering objective in the same situation. This is because we are now concerned with the constant factor in the bounds given in Theorem 1, and how it relates to \( F \). In the standard Orienteering problem, the best thing to do is to collect points in areas with high density of points and such that \( g(x) \) is as large as possible (i.e. where the vehicle has the greatest maneuverability), in order to maximize the number of
4.2. LOWER-BOUND FOR TSP\textsubscript{\Pi} IN THE STOCHASTIC SETTING

targets which can be visited by a short trajectory — we call a point \( x \) lucrative if \( f(x)g(x) \) is large (relative to other points in \( \mathcal{X} \)). However, in the TSP, the vehicle must collect all points, many of which will be in areas which are not so lucrative.

To avoid this, we define a modified version of the Orienteering problem, where the objective is structured in such a way as to encourage the use of less lucrative regions (so as to avoid bias towards any particular region). We do this by introducing a cost function over the space, and then asking for a trajectory with at most \( \lambda \) cost.

**Definition 24** (Cost Function and the Cost of a Trajectory). We define the cost function \( h : \mathcal{X} \rightarrow \mathbb{R} \) as

\[
h(x) = \left[ f(x)g(x) \right]^{\frac{1}{\gamma}}.
\]

Then, for a trajectory \( \pi \in \Pi \), its cost is

\[
\epsilon^\text{c}(\pi) \triangleq \int_0^{\epsilon(\pi)} h(\pi(t)) dt.
\]

We use the ‘\( \epsilon \)’ symbol in general to denote cost-denominated versions of definitions from the previous section. We now give the formal definition of the Cost-Balanced Orienteering (CBO) problem (so named because the cost function is intended to balance out ‘lucrative’ and ‘non-lucrative’ areas):

**Definition 25** (Cost-Balanced Orienteering). We first define the cost-bounded trajectory set

\[
\Pi^\lambda_\epsilon \triangleq \{ \pi \in \Pi : \epsilon^\text{c}(\pi) \leq \lambda \}
\]

(we include the \( F \) term in the notation because the cost is dependent on the distribution of the target points). Then, the Cost-Balanced Orienteering problem is defined by

\[
\text{CBO}_\Pi(X_1, X_2, \ldots, X_n; \lambda, F) = \max_{\pi \in \Pi^\lambda_\epsilon}(|\pi \cap \{X_i\}_{i=1}^n|)
\]

i.e. the maximum number of targets which can be visited by a trajectory of cost at most \( \lambda \).

We then have the following theorem for this problem, analogous to Theorem 2:

**Theorem 7** (Bounds for Cost-Balanced Orienteering). Let \( \Pi \) be a symmetric constraint set satisfying the assumptions in Section 1.5. Then, there exist constants \( \beta_2^\gamma \) and \( \beta_2^\gamma \) (where \( 0 < \beta_2^\gamma < \beta_2^\gamma \)) such that for any distribution \( F \) with \( \Pi \)-bounded support and bounded density, if \( X_1, X_2, \ldots, X_n \sim F \) identically, then

\[
\beta_2^\gamma \lambda n^{-\frac{1}{\gamma}} \leq \text{CBO}_\Pi(X_1, X_2, \ldots, X_n; \lambda, F) \leq \beta_2^\gamma \lambda n^{-\frac{1}{\gamma}}
\]

with very high probability.

We now give some intuition for using this particular cost function. In particular, we note that our previous logic on Orienteering (which we will use again, with slight modifications) was based on the fact that for some \( \epsilon \) proportional to \( n^{-\frac{1}{\gamma}} \), the \( \epsilon \)-time reachable set from any \( x \in \mathcal{Z} \) has volume of at most \( 1/n \). Thus, for every distance \( \epsilon \) we travel, we can expect to pick up one point — we refer to this as traversing an \( \epsilon \)-reachable set — and so to pick up all \( n \) points, we need to travel a distance of \( n \cdot \epsilon = \Theta(n^{\gamma-1/\gamma}) \). The trick using the Union and Chernoff bounds showed that there isn’t a way to do significantly (i.e. more than a constant factor) better than this ‘average’ path.

However, we now want a tighter argument where we are concerned with the exact ratio between \( \epsilon \) (for which an \( \epsilon \)-reachable set has volume \( 1/n \)) and \( n^{-\frac{1}{\gamma}} \) as \( n \to \infty \). We cannot escape the fact that this ratio depends on the configuration that we are in as well as the distribution \( F \). The cost function above makes this ratio the same for every point (more specifically, for the reachable-set-maximizing configuration at every point \( x \in \mathcal{X} \)).

We show this mathematically as follows. For any \( x \in \mathcal{X}_F \), we define

\[
z^*(x) = \arg\max_{z \in [x]} \left( \lim_{\epsilon \to 0} \frac{\text{vol}_X(\tilde{R}_\epsilon(z))}{\epsilon^\gamma} \right)
\]
i.e. the configuration at $x$ which maximizes the volume of the $\epsilon$-time reachability set as $\epsilon \to 0$. Then by the definition of $g(x)$,

$$g(x) = \lim_{\epsilon \to 0} \frac{\text{vol}_x(\hat{R}_\epsilon(z^*(x)))}{\epsilon^\gamma}.$$ 

We now assume, without loss of generality, that the density function $f$ is continuous. For $\epsilon$ very near 0 and any $y$ within a distance $\epsilon$ of $x$ we can approximate $f(y)$ as $f(x)$. Thus, the probability that a target point (distributed according to $F$) falls into a region within the $\epsilon$-neighborhood of $x$ can be approximated as the volume of the region times $f(x)$. Thus, for any constant $c$, if $\epsilon = c \cdot n^{-\frac{1}{2}}$, then

$$\Pr_{X \sim F}[X \in \hat{R}_\epsilon(z^*(x))] \approx f(x) g(x) c^\gamma (1/n)$$

so for $c = f(x)^{-\frac{1}{2}} g(x)^{-\frac{1}{2}}$, this comes out to just $1/n$. Thus, if we set our cost function to be the inverse of this, the cost of traversing a $\epsilon$-reachable set covering $1/n$ of the probability space becomes constant everywhere (rather than being cheaper in lucrative areas, as in the original Orienteering problem) – specifically, the cost becomes $n^{-\frac{1}{2}}$.

### 4.2.2 Upper-Bounding CBO\_II

We give the upper bound of the Cost-Balanced Orienteering problem as the following proposition:

**Proposition 4.** Let $\Pi$ be a symmetric constraint set satisfying the assumptions in Section 1.5. Then, there exists a constant $\beta_2^+ > 0$ such that for any distribution $F$ satisfying our assumptions, if $X_1, X_2, \ldots, X_n \sim F$ i.i.d., then

$$\text{CBO}_{\Pi}(X_1, X_2, \ldots, X_n; \lambda, F) \leq \beta_2^+ \lambda n^{1/2}$$

with very high probability.

To show the upper bound of $\text{CBO}_{\Pi}$, we need to find cost-constrained analogues of Lemmas 9 and 10. We first define the $\epsilon$-cost reachable set, which is a direct generalization of the $\epsilon$-reachable set. In these definitions, we let the distribution $F$ and the constraint set $\Pi$ be fixed (so $f$ and $g$ are fixed), so that we don’t have to specify them in our definitions.

We define the $\epsilon$-cost reachable sets in much the same way that we defined the $\epsilon$-reachable sets.

**Definition 26 ($\epsilon$-Cost Reachable Sets).** For any $z \in Z$, let

$$R^\epsilon(z) \triangleq \{z' \in Z : \exists \pi \in \Pi^\epsilon \text{ which goes from } z \text{ to } z'\}.$$ 

We define a workspace variant of this as

$$\hat{R}^\epsilon(z) \triangleq [R^\epsilon(z)]_x.$$ 

We now give the extensions of Lemmas 9 and 10.

**Lemma 12 (Polynomial Cover of $Z_F$).** There exists a polynomial $p$ such that for any $\epsilon > 0$, there exists a collection of $p(1/\epsilon)$ configurations $y_1^\epsilon, y_2^\epsilon, \ldots, y_{p(1/\epsilon)}^\epsilon \in Z$ such that

$$Z_F \subseteq \bigcup_{i=1}^{p(1/\epsilon)} R^\epsilon(y_i^\epsilon).$$

We refer to this set as $Y^*_\epsilon$.

In short, as $\epsilon \to 0$, $Z_F$ can be covered by a collection of polynomially (in $1/\epsilon$) many $\epsilon$-reachability sets.

**Proof.** Consider the collection of straight, parallel flexible trajectories whose existence we assumed in Section 2.3; let $\tau$ be any of these trajectories.

Then, by the Ball-Box Theorem (Lemma 3), $R^\epsilon_\tau(\tau(t))$ 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 \([0,1]^d\). We can then simply carry out the tiling with these cubes.

Lemma 13 (Fixed-Size Cover of \( R_{\epsilon'}^2(z) \)). There exists a positive integer \( b \) such that, for any \( \epsilon > 0 \) and \( z \in \mathcal{Z} \), there exists a set of \( b \) configurations \( y_1, y_2, \ldots, y_b \in \mathcal{Z} \) such that

\[
R_{\epsilon'}^2(z) \subseteq \bigcup_{i=1}^b R_{\epsilon'}^2(y_i).
\]

As before, we refer to the set \( Y(z) := \{y_1, \ldots, y_b\} \) as the ‘menu’ at \( z \).

Proof. This is a corollary of Lemma 10, since we assume that the probability density function \( f \) is Lipschitz-continuous, and our assumptions for the dynamic system give a Lipschitz-continuous \( g \) bounded away from \( 0 \) — and since both of these are bounded above, we get a piecewise Lipschitz-continuous \( h \). While there are some small issues concerning the boundaries between pieces, as \( n \to \infty \) (meaning the sizes of the reachable sets tend to 0) they become increasingly negligible.

The proof then goes exactly as the proof for the upper-bound for ORNT went (Proposition 2); since the proof is identical, we will not re-write it here.

4.2.3 Lower-Bounding \( \text{TSP}_n \) from \( \text{CBO}_n \)

We now use the above prove our desired \( \text{TSP}_n \) lower bound. However, we note that because \( \text{CBO} \) is not directly related to \( \text{TSP} \) in the way that \( \text{ORNT} \) is, we will have to use a different trick to convert our \( \text{CBO} \) upper-bound proof into a \( \text{TSP} \) lower-bound proof. As a brief note, in the following theorems we will be referring to integrals over the whole workspace, whereas previously we were discussing \( \mathcal{X}_F \). These integrals are really the same thing because outside of \( \mathcal{X}_F \), the function \( f \) takes the value 0.

Proposition 5. There exists some \( \beta^*_1 \) such that for every \( F \) satisfying our assumptions,

\[
\beta^*_1 n^{\frac{2}{\gamma}} \int_{\mathcal{X}} f(x)^{\frac{2}{\gamma}} g(x)^{-\frac{1}{\gamma}} dx \leq \text{TSP}_n(X_1, X_2, \ldots, X_n)
\]

Proof. Suppose we have our optimal trajectory \( \pi^* \), of length \( L := \ell(\pi^*) \), which must pass through each target point \( X_i \). We now consider the cost function \( h \) evaluated at these points. In particular, let us define \( t_i \) to be the time such that \( \pi^*(t_i) = X_i \) (in the edge-case where there is more than one such \( t_i \), we pick one arbitrarily. We now consider the segment of time \( I_i := [t_i - \frac{Lh(x)^{-1}}{2}, t_i] \) (we need to fudge if \( t_i < \frac{Lh(x)^{-1}}{2} \) by introducing a small “dummy segment” at the beginning, but we will ignore this technicality); for sufficiently small \( \lambda \), the cost function varies only slightly along a section of any trajectory of length \( \frac{Lh(x)^{-1}}{2} \) for any \( x \), and therefore the cost incurred by \( \pi^* \) during the time segment \( I_i \) (for any \( i \)) is at most \( \lambda \) (we divided by 2 as a fudge factor).

We now consider any \( t \in [0, L] \), and consider the segment of \( \pi^* \) starting at \( t \) and having cost of \( \lambda \). By Theorem 7 we know that with very high probability there is no such segment which visits more than \( \beta^*_2 \lambda n^{\frac{1}{\gamma}} \) target points. Since \( \pi^* \) incurs a cost of at most \( \lambda \) over \( I_i \) for all \( i \), this implies that for any \( t \)

\[
|\{i : t \in I_i\}| \leq \beta^*_2 \lambda n^{\frac{1}{\gamma}}
\]

Therefore, since these segments can only cover \( \pi^* \) up to \( \beta^*_2 \lambda n^{\frac{1}{\gamma}} \) layers “deep”, we can conclude that

\[
\sum_{i=1}^n \frac{\lambda h(X_i)^{-1}}{2} \leq L \beta^*_2 \lambda n^{\frac{1}{\gamma}} \quad \Rightarrow \quad \sum_{i=1}^n h(X_i)^{-1} \leq (2 \beta^*_2)L n^{\frac{1}{\gamma}}
\]

with very high probability. But now we can simply consider \( h(X_i)^{-1} \) as a random variable, and hence as \( n \to \infty \), we can apply concentration bounds (specifically, because \( f, g \) are bounded both above and below,
away from 0 and 1, we can use the Chernoff-Hoeffding Bound) to show that with very high probability 
\[
\sum_{i=1}^{n} h(X_i)^{-1} \approx nE_{X \sim F}[h(X)^{-1}].
\]
But we know that since \( h(x) = [f(x)g(x)]^{\frac{1}{2}} \), we have
\[
E_{X \sim F}[h(X)^{-1}] = \int_X f(x)^{\frac{2}{\gamma}}g(x)^{-\frac{1}{2}} dx
\]
Therefore, letting \( \beta_1 = \frac{1}{2\beta_2} \), we can put this all together to get
\[
\beta_1 n^{\frac{2\gamma}{\gamma - 1}} \int_X f(x)^{\frac{2}{\gamma}}g(x)^{-\frac{1}{2}} dx \leq L
\]
Thus, with very high probability, any trajectory which visits all \( n \) target points must satisfy this lower bound.

### 4.3 Upper Bound of \( TSP_\Pi \) in the Stochastic Setting

We now discuss the upper bound given in Theorem 3. As we did for non-symmetric vehicles, we will show the upper bound by using a cell-searching strategy; however, in the case of a symmetric vehicle, this is considerably simplified by the fact that we can visit all target points in a cell without having to leave the cell and return later. As before, we will use as a rule of thumb that every cell should contain \( \sim n^{-1} \) of the probability mass under \( F \). We note that this means the radius \( \epsilon \) of the cell depends on both \( f \) and \( g \).

**Proposition 6.** Let \( \Pi \) be a symmetric set of dynamical constraints and \( F \) be a probability distribution (with density function \( f \)) on \( X \), both satisfying the assumptions in Section 1.5. Then there is a constant \( \beta^+ > 0 \) such that if \( X_1, X_2, \ldots, X_n \sim F \) independently then:
\[
TSP_\Pi(X_1, X_2, \ldots, X_n) \leq \beta^+ \left(n^{\frac{2\gamma}{\gamma - 1}} \right) \int_X f(x)^{\frac{2}{\gamma}}g(x)^{-\frac{1}{2}} dx
\]
with very high probability.

**Proof.** We begin by breaking up the area \( X_F \) into \( m \) (constant independent from \( n \)) small (roughly hyper-rectangular) pieces \( P_1, \ldots, P_m \) for which \( f(x) \) and \( g(x) \) are roughly constant. Formally, we simply select a small \( \delta \) (say \( \delta = 0.0001 \)) such that there is some \( f_j, g_j \in \mathbb{R}_{>0} \) such that for any \( x \in P_j \),
\[
f(x) \in [f_j, (1 + \delta)f_j] \text{ and } g(x) \in [g_j, (1 + \delta)g_j]
\]
and furthermore where there is some “best” \( s_j \in S \) such that for any \( x \in P_j \),
\[
g^2(x, s_j) \in [g_j, (1 + \delta)g_j]
\]
(this ensures that a consistent configuration produces the most agility). Provided we pick a sufficiently large \( m \), this is always possible.

We then tile each \( P_j \) with cells in a similar manner as we did for the general vehicles case; however, one difference is we are now inscribing these rectangular cells in the reachable sets of the anchor points, rather than the deviation sets. This is because the vehicle is symmetric, so when we visit a target point, we can return to the anchor point rather than needing to proceed directly to the next anchor point.

We now define \( \epsilon_j := a(f_jg_jn)^{-\frac{1}{4}} \) where \( a \) is a constant whose purpose we explain in a minute; this definition is made so that
\[
\text{vol}_X \hat{R}_{\epsilon_j}(x, s_j) \approx g_j \epsilon_j^2 = a^2 g_j (f_jg_jn)^{-1} = a^2 f_j^{-1} n^{-1}
\]
which in turn implies that the probability mass covered by the reachable set is approximately \( \alpha^\gamma n^{-1} \) since \( f_j \) is approximately the density of \( F \) on \( P_j \). The \( \alpha \) term is there so that each cell covers \( n^{-1} \) (because the reachable set itself is generally not rectilinear).

We then do the following within each piece \( P_j \): (1) tile the piece with cells of radius \( \epsilon_j \) anchored at points of the form \( (x, s_j) \), so that each tile covers approximately \( n^{-1} \) of the probability mass of \( F \); (2) pass through each cell collecting all target points contained within by starting at the anchor, getting the first target point, returning to the anchor, getting the second target, and so forth.

We can therefore divide the generated path \( \pi \) into three kinds of segments:
4.3. UPPER BOUND OF TSP$_n$ IN THE STOCHASTIC SETTING

1. segments joining one piece to another;
2. segments from an anchor to a target point, and back;
3. segments from an anchor to the next within a piece.

For type (1), because there are only a fixed number pieces, these can contribute at most a constant length and can therefore be ignored as negligible as $n \to \infty$.

For type (2), we note that going from an anchor to a target point requires a trajectory of length at most $\epsilon_j$, and therefore the total cost of segments of this type is at most $2\epsilon_j n_j$ where $n_j$ is the number of target points in $P_j$.

For type (3) we note that if two cells are adjacent – that is, they share a facet in $X$ – then the reachable sets that generated them overlap in $Z$. Therefore, if we have anchors $z_1, z_2$, we simply find $z' \in R_{\epsilon_j}(z_1) \cap R_{\epsilon_j}(z_2)$ and traverse $z_1 \to z' \to z_2$. By definition of reachable sets, this motion takes at most $2\epsilon_j$ time. And by Lemma 7, we can tile in such a way that the adjacency graph is connected. We can therefore make a tree and traverse it as if doing a depth-first search, which visits all nodes while traversing all edges at most twice.

Therefore, we can upper-bound the length of the trajectory this algorithm generates by summing these three types of segments. The first, as mentioned, is negligible and we ignore it.

The second is a random variable. Let $\epsilon(X_i)$ denote the radius of the cell in which $X_i$ falls; then the total length of segments of type (2) is upper-bounded by $\sum_{i=1}^{n} 2\epsilon(X_i)$. However, we note that $\epsilon(X_i)$ is a random variable, and $\epsilon(X_1), \ldots, \epsilon(X_n)$ are distributed i.i.d. – and that they fall within a well defined range, since they can only take $m$ distinct values. Therefore, using our small $\delta$ again (it doesn’t have to be the same number, but it might as well be) and applying Hoeffding’s inequality, we know that with very high probability,

$$\sum_{i=1}^{n} 2\epsilon(X_i) \in [2(1 - \delta)n\mathbb{E}\epsilon(X), 2(1 + \delta)n\mathbb{E}\epsilon(X)]$$

(where $X \sim F$ for the two expectations).

We now have to compute $\mathbb{E}\epsilon(X)$; however, this is fairly easy. We note that if $X$ falls into a cell anchored at $z = (x, s_j)$, then it must necessarily be physically close to $x$ (and in fact in the same piece $P_j$). Therefore, $f_j \approx f(X_j)$ and $g_j \approx g(X_j)$. Therefore, we can substitute $(f(X)g(X)n)^{-\frac{1}{2}}$ for $\epsilon(X)$, and simply writing out the expected value gives

$$\mathbb{E}\epsilon(X) \approx \alpha \int_{X} f(x) (f(x)g(x)n)^{-\frac{1}{2}} dx = \alpha n^{-\frac{1}{2}} \int_{X} f(x)^{\frac{2}{n}} g(x)^{-\frac{1}{2}} dx$$

(the quality of this approximation depends on the $\delta$ we chose earlier, and for ease of reading we omit it). Therefore, combining this with the approximation above, we get

$$\sum_{i=1}^{n} 2\epsilon(X_i) \approx (2n)\left(\alpha n^{-\frac{1}{2}} \int_{X} f(x)^{\frac{2}{n}} g(x)^{-\frac{1}{2}} dx\right) = 2\alpha n^{\frac{2}{n}} \int_{X} f(x)^{\frac{2}{n}} g(x)^{-\frac{1}{2}} dx$$

with very high probability.

Finally, we can move on to the third type of segment. However, by solving the second type, we have already done most of the heavy lifting regarding the third type. We note that what we are really looking for is the sum over all cells of their radii (times 2); however, since each cell covers $n^{-1}$ of the probability mass of $F$, this is just $n\mathbb{E}\epsilon(X)$, which we already know from the above. However, in the worst case, we have to traverse each edge in the tree twice. Hence, the total length of all segments of $\pi$ of type (3) is also approximately

$$4\alpha n^{\frac{2}{n}} \int_{X} f(x)^{\frac{2}{n}} g(x)^{-\frac{1}{2}} dx$$

(though in reality since the tiling is regular, we only need to traverse one edge per cell, so we can halve this, plus a negligible factor for possible double-traversals around the edges of the tiling).

Thus, summing these up, we find that with very high probability, our algorithm produces a trajectory of length at most

$$4\alpha n^{\frac{2}{n}} \int_{X} f(x)^{\frac{2}{n}} g(x)^{-\frac{1}{2}} dx$$
visiting all $n$ target points (plus a term that doesn’t depend on $n$ and can therefore be ignored). \hfill \Box

## 4.4 Orienteering for Symmetric Vehicles in the Stochastic Setting

We now briefly state and show stronger bounds on $\text{ORN}_{\Pi}$ for symmetric vehicles in the stochastic setting. Unlike in the general case, $\text{ORN}_{\Pi}$ is not merely the opposite of $\text{TSP}_{\Pi}$—because the vehicle is not expected to collect every target point, the vehicle can stay in the most “lucrative” areas of the workspace (i.e., where its agility and the density of the target points is the highest), which means that the number of points which can be collected depends on the maximum of these values rather than their average. We restate our result (Theorem 5):

**Theorem (Stronger Bounds on ORN$^T$ for Symmetric Vehicles).** Let $\Pi$ be a symmetric set of dynamical constraints satisfying the assumptions in Section 1.5, and let $F$ be a full-dimensional probability distribution over $\mathcal{X}$ which is bounded and has a bounded density function $f$. Let $X_1, X_2, \ldots, X_n \sim F$ independently.

Then:

$$\text{ORN}_{\Pi}(X_1, X_2, \ldots, X_n; \lambda) = \Theta(\lambda n^{\frac{1}{2}} \sup_{x \in \mathcal{X}} (f(x)g(x))^{\frac{1}{2}})$$

with very high probability.

**Proof.** First, let us define $h^* := \sup_{x \in \mathcal{X}} (f(x)g(x))^{\frac{1}{2}}$ (which we note is also the same thing as $\sup_{x \in \mathcal{X}} ((f(x)g(x))^{\frac{1}{2}})$, so we can avoid ambiguities).

We prove this by defining an arbitrary $\delta > 0$. We then define the following set:

$$\mathcal{X}_\delta = \{x : (f(x)g(x))^{\frac{1}{2}} \geq h^* - \delta\}$$

By the definition of $h^*$ as the supremum and $f, g$ as piecewise-continuous with finitely many pieces, all of which are full-dimensional, we know that $\mathcal{X}_\delta$ contains some open ball $\mathcal{X}_\delta^*$ in $\mathcal{X}$. Furthermore, $n$ does not appear anywhere in the definition of $\mathcal{X}_\delta^*$, so this ball is of a fixed size as $n \to \infty$.

We now consider what $\epsilon$ is on this set. Recall that $\epsilon$ is chosen so that a cell covers $n^{-1}$ of the probability mass; therefore, for a reachable set anchored at $x$, we have that $f(x)g(x)e^{\gamma} \approx n^{-1}$ (with some fixed constant factor). This means that for $x \in \mathcal{X}_\delta^*$, we get

$$\epsilon \approx c(f(x)g(x)n)^{-\frac{1}{2}} \leq (h^* - \delta)^{-\frac{1}{2}} n^{-\frac{1}{2}}$$

where $c$ is a fixed constant representing the fraction of the reachable set the inscribed cell occupies.

Furthermore, as $n \to \infty$ the number of cells that fit into $\mathcal{X}_\delta^*$ grows linearly with $n$ (this is a direct consequence of the fact that each cell covers probability mass inversely proportional to $n$) while $\epsilon$ goes to 0 only proportional to $n^{-\frac{1}{2}}$; therefore, the sum of the radii of the cells grows proportionally to $n^{\frac{2 \gamma - 1}{\gamma}}$, meaning that no matter how large $\lambda$ is, at some point using the algorithm given above for TSP on just the cells in $\mathcal{X}_\delta^*$ will take more than $\lambda$ time.

Therefore, if we run this algorithm on $\mathcal{X}_\delta^*$ and terminate after we have made a trajectory of length $\lambda$, we note that we visit a target point once per travel-time of at most $4\epsilon$ (we get this by allotting $2\epsilon$ for anchor-to-anchor travel and $2\epsilon$ for anchor-to-target-and-back travel), except when we encounter empty cells (which make us waste $2\epsilon$ time without visiting any targets at all). To simplify the analysis, we change the algorithm slightly: whenever the vehicle visits a cell, it takes $4\epsilon$ time, and it visits one target point if that cell contains at least one. Therefore, the total number of cells visited is

$$\frac{\lambda}{4\epsilon} \geq \frac{\lambda}{4\epsilon} (h^* - \delta)^{\frac{1}{2}} n^{\frac{1}{2}}$$

and the number of target points visited is the number of those cells which contain at least one target point.

However, it is well-known that as $n \to \infty$, if we distributed $n$ target points into cells, each of which covers $n^{-1}$ probability mass, the probability that any given cell is empty approaches $e^{-1}$ (where $e$ denotes Euler’s constant) and that with very high probability the number of target points in any collection of $\Theta(n^{\frac{1}{2}})$ cells will be approximately the expected number (since $\gamma \geq 2$); in particular, with very high probability, at most
half of them will be empty (we're choosing sub-optimal constants just to make the proof easier to write). Therefore, using this admittedly suboptimal algorithm, we can conclude that

\[ \text{ORNT}_\Pi(X_1, \ldots, X_n; \lambda) \geq \frac{\lambda}{8c}(h^* - \delta)^{\frac{1}{2}}n^{\frac{1}{2}} \]

with very high probability. But since this works for any \( \delta > 0 \) (for smaller \( \delta \), it just takes a larger \( n \) to get good probabilities), we can conclude that we have shown the lower-bound of Theorem 5.

To prove the upper-bound (that no path of length \( \lambda \) can do significantly better), we can simply re-use the proof of Proposition 2 with effectively no alteration, other than to note that \( \epsilon \) has a lower bound proportional to \( (h^*n)^{-\frac{1}{2}} \). Thus, we have shown both the lower and upper bounds and completed our proof. \( \square \)

**Remark:** This logic does not apply to the general vehicles case because a trajectory of length \( \lambda \) might be forced to leave the most lucrative area since it cannot reverse back into it (and in fact might even be forced to waste most of its length on empty regions).

## 4.5 Symmetric Vehicles and Adversarially-Distributed Targets

We now consider a more accurate deterministic upper bound than the one given in Corollary 1 for symmetric constraints. When speaking of a deterministic upper bound, it does not make any sense to speak of a probability distribution, since the distribution itself is irrelevant; what is relevant, however, is the support set (i.e. the set \( \mathcal{X}^* \subseteq \mathcal{X} \) which must contain the target points \( X_1, X_2, \ldots, X_n \)). We thus formulate the deterministic problem as one of selecting \( X_1, X_2, \ldots, X_n \) from this \( \mathcal{X}^* \) (which we assume is full-dimensional everywhere); the what determines the constant factor is now only the agility function \( g \) of the vehicle.

**Proposition 7** (Tight Deterministic Upper Bound). Let \( \Pi \) be a symmetric vehicular constraint set. Then there exists positive constants \( \beta_-^* \) and \( \beta_+^* \) such that, for any full-dimensional \( \mathcal{X}^* \subseteq \mathcal{X} \),

\[
\beta_-^* n^{\frac{1}{2}} \left( \int_{\mathcal{X}^*} g(x)^{-1} dx \right)^{\frac{1}{2}} \leq \max_{X_1, \ldots, X_n \in \mathcal{X}^*} \text{TSP}_\Pi(X_1, X_2, \ldots, X_n) \leq \beta_+^* n^{\frac{1}{2}} \left( \int_{\mathcal{X}^*} g(x)^{-1} dx \right)^{\frac{1}{2}}.
\]

Furthermore, \( \beta_-^* \geq \beta_+^* \) (the lower constant from Theorem 3).

This is intuitively logical, since the length of the path is inversely proportional to the agility of the vehicle over the allowable set \( \mathcal{X}^* \), and increases if \( \mathcal{X}^* \) is expanded.

**Remark:** We note that unlike in Beardwood-Halton-Hammersley and the previous theorems, we have an exponent outside the integral term. This is because in the previous problem, where \( F \) and \( \Pi \) are fixed, the space can be partitioned into small pieces; the cost of traversing the points in each piece is then independent of the cost in each other piece (close to independent as \( n \to \infty \), that is, since the fact that there is a fixed number \( n \) of target points makes them dependent to a very small degree), thus leading (as the size of the pieces tend to 0) to an integral (with no outside exponents). However, when the target points are distributed adversarially (i.e. \( F \) is no longer fixed), we can no longer treat each piece independently — specifically, the agility of the vehicle on one piece might affect the number of points which are placed in a different piece, since the adversary will want to place more points in areas where the vehicle is less agile.

### 4.5.1 The Lower Bound on the Worst Case

In order to show the lower bound, we will use a slightly weaker strategy. Instead of choosing a fixed \( X_1, X_2, \ldots, X_n \), we will select a continuous \( f \) which maximizes the quantity from Theorem 3, which obviously serves as a lower bound. We thus have to solve the following problem:

\[
\begin{align*}
\text{maximize} & \quad \int_{\mathcal{X}^*} f(x)^{\frac{1}{n}} g(x)^{-\frac{1}{2}} dx \\
\text{subject to} & \quad \int_{\mathcal{X}^*} f(x)dx = 1, \text{ and } f(x) \geq 0 \text{ for all } x \in \mathcal{X}^* 
\end{align*}
\] (4.1)


(i.e. the constraint is that \( f \) is a probability density function over \( X^* \)).

**Lemma 14.** Expression (4.1) is maximized by \( f^*(x) \propto g(x)^{-1} \), i.e.

\[
 f^*(x) = \frac{g(x)^{-1}}{\int_{X^*} g(y)^{-1} dy}.
\]

This trivially satisfies the two conditions in expression 4.1.

We note that plugging \( f^* \) into expression 4.1 gives

\[
 \int_{X^*} f^*(x)^{\gamma-1} g(x)^{-\frac{1}{\gamma}} dx = \int_{X^*} \left( \frac{g(x)^{-1}}{\int_{X^*} g(y)^{-1} dy} \right)^{\gamma-1} g(x)^{-\frac{1}{\gamma}} dx
\]

\[
 = \int_{X^*} \frac{g(x)^{-1}}{\left( \int_{X^*} g(y)^{-1} dy \right)^{\gamma-1}} dx = \left( \int_{X^*} g(x)^{-1} dx \right)^{\frac{1}{\gamma}}.
\]

We also note that if we scale \( g \), the optimal \( f \) won’t change; thus, we assume without loss of generality that

\[
 \int_{X^*} g(y)^{-1} dy = 1, \text{ so } f(x) = g(x)^{-1}
\]

(of course, we rescind this assumption when we need to plug the solution back into expression 4.1).

**Proof of Lemma 14.** We will show this using Hölder’s Inequality (in fact, the lemma practically *is* Hölder’s Inequality). We first make the following definitions:

\[
 \tilde{f}(x) = f(x)^{\frac{\gamma-1}{\gamma}} \text{ and } \tilde{g}(x) = g(x)^{-\frac{1}{\gamma}}
\]

and note that since by definition \( f(x), g(x) \geq 0 \) for all \( x \), we know that \( \tilde{f}(x), \tilde{g}(x) \geq 0 \) for all \( x \), so we can ignore the absolute value function in the statement of Hölder’s Inequality. We then define the constant \( \alpha \) (we re-use previous notation since we don’t require it anymore) \( \alpha = \frac{\gamma}{\gamma-1} \). Note that \( \frac{1}{\alpha} + \frac{1}{\gamma} = 1 \), as required by Hölder’s Inequality. Thus:

\[
 \int_{X^*} \tilde{f}(x) \tilde{g}(x) dx \leq \left( \int_{X^*} \tilde{f}(x)^{\alpha} dx \right)^{1/\alpha} \left( \int_{X^*} \tilde{g}(x)^{\gamma} dx \right)^{1/\gamma}.
\]

But, using the definitions from above,

\[
 \tilde{f}(x)^{\alpha} = (f(x)^{\frac{\gamma-1}{\gamma}})^{\gamma/(\gamma-1)} = f(x) \text{ and } \tilde{g}(x)^{\gamma} = (g(x)^{-\frac{1}{\gamma}})^{\gamma} = g(x)^{-1}
\]

so we can rewrite the inequality as

\[
 \int_{X^*} \tilde{f}(x) \tilde{g}(x) dx \leq \left( \int_{X^*} f(x) dx \right)^{1/\alpha} \left( \int_{X^*} g(x)^{-1} dx \right)^{1/\gamma}; \tag{4.2}
\]

however, by the condition that \( f \) is a probability density function and our assumption (without loss of generality) about \( g(x)^{-1} \), we know that

\[
 \int_{X^*} f(x) dx = \int_{X^*} g(x)^{-1} dx = 1
\]

implying that the right hand side of expression 4.2 is just 1. Thus,

\[
 \int_{X^*} f(x)^{\gamma-1} g(x)^{-1} dx = \int_{X^*} f(x) \tilde{g}(x) dx \leq 1
\]

for any probability density function \( f \). But, using \( f^* \) as defined in Lemma 14, it is trivial to see that

\[
 \int_{X^*} f^*(x)^{\gamma-1} g(x)^{-1} dx = 1
\]

thus showing that \( f^* \) is the maximizing density function.

Therefore, if we distribute the points \( X_1, X_2, \ldots, X_n \) according to \( f^* \) (i.i.d.), by Theorem 3, we will with high probability get a length of the scale needed by Proposition 7 (by the computations given after Lemma 14). We thus need to show only that we cannot pick \( X_1, X_2, \ldots, X_n \) that make the TSP tour significantly longer than picking them randomly according to \( f^* \).
4.5. SYMMETRIC VEHICLES AND ADVERSARILY-DISTRIBUTED TARGETS

4.5.2 The Upper Bound on the Worst Case

First, let \( F^\ast \) be the probability distribution with the ‘worst-case’ density function \( f^\ast \) given above, so \( X^\ast = X_{F^\ast} \) by definition. We then note that \( f^\ast \) is exactly the probability distribution which keeps the cost function from the CBO constant over all \( x \in X^\ast \), as

\[
h(x) = \left[ f^\ast(x) g(x) \right]^{\frac{1}{2}} = \left( \frac{1}{\int_{X^\ast} g(y)^{-1} dy} \right)^{\frac{1}{2}} = \left( \int_{X^\ast} g(y)^{-1} dy \right)^{-\frac{1}{2}}
\]

for all \( x \in X^\ast \).

For each \( x \in X^\ast \), let \( \epsilon_n(x) > 0 \) be such that

\[
\text{vol}_{X^\ast} \left[ R_{\epsilon_n(x)}^\ast (q^\ast(x)) \right] = 1/n.
\]

Then \( \epsilon_n(x) \) is roughly the same for each \( x \in X^\ast \); formally, for any \( x, y \in X^\ast \),

\[
\lim_{n \to \infty} \frac{\epsilon_n(x)}{\epsilon_n(y)} = 1.
\]

The probabilistic aspect of the upper bound of Theorem 3 came from the fact that some cells have larger \( \epsilon \) than others – so if an unusually large number of target points fall in large-\( \epsilon \) cells, this increases the overall length of the path. However, in this case, all cells have (approximately) the same \( \epsilon = \epsilon_n(x) \) by default, so whichever cells the target points fall in does not change the length of the tour returned by the algorithm.

Thus, the upper bound for distribution \( F^\ast \) as given in Theorem 3 is actually deterministic (thanks to the special properties of \( F^\ast \)), and hence, this upper bound is a direct corollary of Theorem 3, i.e. the algorithm given in Section 4.3 collects all the targets with a path of length at most

\[
\beta^+ n^{\frac{1}{5}} \left( \int_{X^\ast} g(x)^{-1} dx \right)^{\frac{1}{2}}
\]

with no chance of failure.

4.5.3 Orienteering with Adversarial Targets

We also show what happens for the Orienteering problem with adversarial target placement. In this case, since the adversary is effectively trying to make it so no place is especially “lucrative”, ORNT becomes the same problem as CBO (since the adversary must do this by balancing the lucrativity over the space \( X^\ast \). We restate the our result (Theorem 6):

**Theorem** (Worst Case ORNT for Symmetric Vehicles). Let \( \Pi \) be a symmetric vehicular constraint set. Then for any full-dimensional \( \lambda \subseteq X^\ast \),

\[
\min_{X_1, \ldots, X_n \in X^\ast} \text{ORNT}_\Pi(X_1, X_2, \ldots, X_n; \lambda) = \Theta \left( \lambda n^{\frac{1}{2}} \left( \int_{X^\ast} g(x)^{-1} dx \right)^{-\frac{1}{2}} \right)
\]

(this is guaranteed).

**Proof.** We first note that if we can visit any \( n \) adversarially-placed target points with a trajectory of length at most

\[
\beta^+ n^{\frac{1}{2}} \left( \int_{X^\ast} g(x)^{-1} dx \right)^{\frac{1}{2}}
\]

then the average sub-trajectory of length \( \lambda \) visits

\[
\Theta \left( \lambda n^{\frac{1}{2}} \left( \int_{X^\ast} g(x)^{-1} dx \right)^{-\frac{1}{2}} \right)
\]

target points, and therefore the best sub-trajectory does at least as well, immediately establishing our lower bound.
To establish the upper bound, we simply use the “adversarial” probability distribution $F^*$ (as given in Section 4.5) whose density is defined as:

$$f^*(x) = \frac{g(x)^{-1}}{\int_{\mathcal{X}^*} g(y)^{-1} dy}$$

We can then apply Theorem 5 with this density function to get that with high probability, the best trajectory of length $\lambda$ collects at most

$$\Theta(\lambda \frac{1}{\gamma} \sup_{x \in \mathcal{X}^*} (f^*(x)g(x))^{\frac{1}{\gamma}}) = \Theta(\lambda \frac{1}{\gamma} \left( \int_{\mathcal{X}^*} g(x)^{-1} dx \right)^{-\frac{1}{\gamma}})$$

of the target points. But if this happens with high probability, the worst case will be upper-bounded by this.

Therefore we have established upper and lower bounds which differ by only a constant factor, finishing our proof.

4.6 Targets on Sets of Lesser Dimension

We now show an interesting corollary to Theorem 4 (the worst-case bound for symmetric vehicles), concerning a distribution of targets on sets of lesser dimension. This corollary covers most of the commonly discussed subsets of lesser dimension:

**Corollary 2.** Let $\Pi$ be a dynamic system satisfying our assumptions and representing a symmetric vehicle and $\gamma$ be its small-time constraint factor, and let $\mathcal{X}^*$ be a subset of $\mathcal{X}$. Define

$$\mathcal{X}_\delta^* := \bigcup_{x \in \mathcal{X}^*} B_\delta(x)$$

(where the $B_\delta(x)$ is the ball of radius $\delta$ around $x$ in $\mathcal{X}$), i.e. the set of points which are at a distance of at most $\delta$ from $S$ in $\mathcal{X}$. If $\lim_{\delta \to 0} \text{vol}_x(\mathcal{X}_\delta^*) = 0$, then:

$$\max_{X_1, \ldots, X_n \in \mathcal{X}^*} \text{TSP}_\Pi(X_1, \ldots, X_n) = o(n^{\frac{\gamma - 1}{\gamma}})$$

(where $a(n) = o(b(n))$ means $\lim_{n \to \infty} \frac{a(n)}{b(n)} = 0$).

Note that the condition on $\mathcal{X}^*$ requires it to be bounded. This corollary covers most normally considered subsets of lesser dimension (strange space-filling constructions and the like are of course a different matter).

**Proof.** To prove this, we simply apply Theorem 4 to $\mathcal{X}_\delta^*$ as $\delta \to 0$ (since $\mathcal{X}^* \subset \mathcal{X}_\delta^*$ by definition) and obtain an upper bound of

$$\lim_{n \to \infty} \frac{\max_{X_1, \ldots, X_n \in \mathcal{X}^*} \text{TSP}_\Pi(X_1, \ldots, X_n)}{n^{\frac{\gamma - 1}{\gamma}}} \leq \beta \left( \int_{\mathcal{X}_\delta^*} g(x)^{-1} dx \right)^{\frac{1}{\gamma}} \leq \beta \max_{x \in \mathcal{X}_\delta^*} (g(x)^{-\frac{1}{\gamma}})(\text{vol}_x^\gamma \mathcal{X}^*_\delta)^{\frac{1}{\gamma}} \to 0 \text{ as } \delta \to 0$$

(the last inequality is just setting everything to the maximum of $g(x)^{-1}$ in the set to make the integral cleaner).

Therefore, for any $c > 0$, we can find a sufficiently small $\delta > 0$ such that $\beta \max_{x \in \mathcal{X}_\delta^*} (g(x)^{-\frac{1}{\gamma}})(\text{vol}_x^\gamma \mathcal{X}^*_\delta)^{\frac{1}{\gamma}} < c$ and we can conclude that because $\mathcal{X}^* \subset \mathcal{X}_\delta^*$, for any $c > 0$

$$\lim_{n \to \infty} \frac{\max_{X_1, \ldots, X_n \in \mathcal{X}^*} \text{TSP}_\Pi(X_1, \ldots, X_n)}{n^{\frac{\gamma - 1}{\gamma}}} \leq \lim_{n \to \infty} \frac{\max_{X_1, \ldots, X_n \in \mathcal{X}_\delta^*} \text{TSP}_\Pi(X_1, \ldots, X_n)}{n^{\frac{\gamma - 1}{\gamma}}} < c$$

and therefore

$$\lim_{n \to \infty} \frac{\max_{X_1, \ldots, X_n \in \mathcal{X}^*} \text{TSP}_\Pi(X_1, \ldots, X_n)}{n^{\frac{\gamma - 1}{\gamma}}} = 0,$n \to \infty \frac{\max_{X_1, \ldots, X_n \in \mathcal{X}^*} \text{TSP}_\Pi(X_1, \ldots, X_n)}{n^{\frac{\gamma - 1}{\gamma}}} = o(n^{\frac{\gamma - 1}{\gamma}})$$

Note that because this is an upper bound on adversarial placement of points, it also bounds any random distribution on such a set.
Chapter 5

Future Work and Conclusion

5.1 Future Work

5.1.1 Multidimensional Surfaces

Our results, which build on the techniques developed in Arias-Castro et al. [1], suggest a further application to fitting multidimensional surfaces to a set of target points. In particular, we can view the trajectory of a vehicle as a one-dimensional surface under some constraints and reformulate the question as fitting a surface of the smallest area to the points. This question can then be generalized to multidimensional surfaces, e.g. a 2-dimensional surface with curvature constraints. In fact, in their paper, Arias-Castro et al. apply their technique to twice-differentiable $k$-dimensional hypersurfaces in $\mathbb{R}^d$ (which they find can contain $\Theta(n^{k/(2d-k)})$ uniformly-distributed points) [1]. As for 1-dimensional trajectories, their technique requires a custom discretization for each set of constraints one wishes to study; however, our work suggests that there may be a general method for automatically generating a suitable discretization which can then be used to show a general theorem applying to all constrained surfaces (or a significant class of constrained surfaces). These results may have applications in fields such as machine-learning.

5.1.2 Non-Full-Dimensional $F$ for Symmetric Vehicles

For general vehicles, it is not possible in general to derive meaningful bounds on the tour length through target points (see Appendix C.1) randomly distributed on a non-full-dimensional subset of $X$. However, as seen in Corollary 2, we can show such results for symmetric vehicles.

Corollary 2 specifically showed that for most commonly-described subsets of $X$ of lesser dimension, the rate of growth of the length of the shortest tour is $o(n^{2-\varepsilon})$ (i.e. strictly smaller than for full-dimensional subsets), even if the points are distributed adversarially. However, it is obvious that the rate of growth depends on the specific characteristics of the subset in question. Therefore, a natural follow-up question is to compute the rate of growth of $\text{TSP}_{nP}(X_1, \ldots, X_n)$ given the subset $\mathcal{X}^*$ (and distribution $F$ if we are in the stochastic case).

5.1.3 Better Constant Bounds

Although we have shown that these constant bounds exist, it is possible that if one applies our method to any given scenario, the resulting upper and lower bounds will differ by a large constant factor. We can thus imagine that it might be possible to improve our methods to achieve better constant bounds, in particular better upper bounds, on the length of the TSP tour (and in the process achieve an algorithm which returns shorter tours).

5.1.4 A Full Extension of Beardwood-Halton-Hammersley

We also note that while we managed to extend the Beardwood-Halton-Hammersley Theorem to achieve the same distribution-dependent term on the length of the path, we are still left with two constants representing
We did not show that as the number of points grows to infinity, the constant converges to a particular limit rather than merely staying within the given interval.

We can then formally define the goal of this particular direction: we wish to show that for every constraint set $\Pi$ with agility function $g$ and small-time constraint factor $\gamma$, there is a particular constant $\beta_\Pi$ such that, for any probability distribution $F$ with density function $f$:

$$\lim_{n \to \infty} \frac{\text{TSP}_\Pi(X_1, X_2, \ldots, X_n)}{n^{-\gamma}} = \beta_\Pi \int_{\mathcal{X}} f(x)^{\frac{\gamma-1}{\gamma}} g(x)^{-\frac{1}{\gamma}} dx$$

Once such a result is achieved, it might even be possible to describe the relationship between $\beta_\Pi$ and the constants $\beta_d$ given in the Beardwood-Halton-Hammersley Theorem.

### 5.1.5 Guaranteed Approximation Algorithms

We also note that while our algorithms provide a constant-factor approximation of the shortest tour with high probability, the approximation is not guaranteed. In particular, it is very possible – even in the case of symmetric vehicles – that the points have been arranged in such a way that a very short tour is in fact possible, but that our algorithms will only achieve the upper bounds we showed. We remark that the typical approximation algorithms for the TSP (such as the Minimum Spanning Tree approximation and Christofides’ algorithm) may not apply, because when a vehicle is at some target point $X_j$, it can still take many different configurations which change its distances to other target points (and restricts the way it can collect those points).

### 5.2 Conclusion

In this work, we united two independent approaches (one originating in the motion planning community, the other in applied probability) to the seminal Traveling Salesman Problem for dynamic vehicles and stochastically-placed target points, allowing us to re-derive the most powerful known theorems concerning this problem with a simplified method.

Along the way to the main result, we formulate and solve an interesting combinatorial problem (the Ball-Collection Problem in Appendix A). Although our TSP algorithm is the only direct application of this problem that we are aware of, we believe that the general structure of this problem should make it applicable to other settings in the future.

We then extended our methods to show novel results in the case of symmetric vehicles, showing not only more precise bounds on the length of the shortest path through randomly-distributed points, but also bounds on the length of the shortest path through adversarially-distributed points (and equivalent bounds for Orienteering in these settings). These results have deep links to a classic theorem on the (unconstrained) TSP through randomly-distributed points in $\mathbb{R}^d$.

Finally, we discussed in some detail a number of interesting unsolved problems, including one (concerning more restrictive distributions of target points) for which our method already provides a basic but suggestive result.
Bibliography


Appendix A

Analyzing the Abstraction

In this chapter, we formally define and solve the abstract problem on which our algorithm for general vehicles rests.

A.1 The Ball Collection Problem

A.1.1 A Representation of the TSP Algorithm

We formally define the abstract problem we used to model the basic cell-merging framework of our algorithm, and prove that we can solve this problem for a fixed cost using the strategy described in Section 3.1.4. The intuitive idea behind this abstraction is that we can view the cells from Section 3.1 as buckets, into which $n$ balls (representing the target locations) are distributed. We can then model the type of recursive cell-merging algorithm described in Section 3.1 as an optimization problem, where we wish to collect all the balls (which represents visiting all the targets) with as low a cost (which represents the length of the path in the original problem) as possible. We will solve this problem as a stand-alone.

Since we are moving to a new, discrete setting and treating this problem by itself, we will be using completely separate notation in this appendix (i.e. we will re-use variable names here which do not refer to the same values as in the rest of the paper). The only two exceptions are $\gamma$ (the small-time constraint factor, which translates to the merging factor in this problem) and $n$ (the number of points distributed, which in this problem translates to the number of buckets and the number of balls distributed among them), because these are the inputs to the abstract problem which come directly from our TSP algorithm.

A.1.2 The Problem

We first describe a 'generic' version of our problem, in which the way the balls are distributed is part of the input. We then formally define the distributions we are interested in (as we will discuss several variants of this problem).

The Generic Ball Collection Problem (BCP)

Let a collection of balls be randomly distributed into a group of $n$ buckets according to some distribution $F$; we allow $F$ to randomize not only the placement of the balls, but also the total number of balls distributed. We then want to remove all the balls from the buckets for as little cost as possible (which we will explicitly define soon). The problem has one parameter, $\gamma$; note that this is the same $\gamma$ as in the main body of this paper, so $\gamma$ is an integer and $\gamma \geq 2$. There is the size of the problem instance, $n$. Finally, there is a cost term $c$, which is initially 1 and decreases (as described below) as operations are carried out. There are two operations we can perform on the set of buckets; for simplicity, we assume that $n = 2^{\ell}$ for some positive integer $\ell$.

- **Collect**: Removes one ball from each nonempty bucket; this requires us to pay a cost of $c$ (for the whole operation, not per bucket).
- **Merge**: Merges buckets in groups of $2^7$ and decreases $c$ by a factor of $2^7 - 1$; this operation requires no cost. Buckets are merged in groups which are fixed (and known) in advance (i.e. the player does not get to choose how to merge the buckets). Merging is not possible once only one bucket remains.

By our assumption about $n$, the number of buckets is always divisible by $2^7$ so merging leaves no remainder. Note that exactly $\ell$ merges are possible before there is only one bucket.

**Definition 27** ($\text{BCP}(n, \gamma, F)$). We define $\text{BCP}(n, \gamma, F)$ to be the random variable corresponding to the optimal (i.e. minimum) cost for removing all the balls from the buckets with parameters $n$ and $\gamma$ and ball distribution $F$.

The problem boils down to the following: faced with a particular collection of balls in the $n$ buckets, how many times should we collect between each merge, so that the total cost paid over the course of the game is minimized? To this end, we will break down a solution strategy into stages which take place between each merge. We denote the number of times we collect between the $i$th merge and the $(i + 1)$th merge as $a_i$ (where $a_0$ denotes the number of times we collect before any merges and $a_\ell$ denotes the number of times we collect after the final $\ell$th merge); any of these can be 0 if we don’t want to collect at all during that stage. Thus, a strategy can be described by the sequence $(a_0, a_1, \ldots, a_\ell)$, with the corresponding cost

$$a_0 + 2^{-1}a_1 + \cdots + 2^{-1}a_\ell = \sum_{i=0}^{\ell} 2^{-1}a_i.$$ 

In a general strategy, we would wait to see the actual distribution of the balls into the buckets before deciding on $a_i$; however, we will restrict ourselves here to considering a blind strategy, where the values $a_i$ are fixed beforehand. This has the benefit of being easier to analyze - in particular, the cost of the strategy will be immediately known, by the above - but this means that the strategy is not guaranteed to remove all the balls. Thus, we will have to analyze the probability that it successfully removes all the balls.

**The Different Distributions of Balls into Buckets**

Our main goal will be to bound $\text{BCP}(n, \gamma, F)$, where $F$ is the distribution which places $n$ balls independently and uniformly at random into the $n$ buckets; we refer to this as the Fixed distribution, since the total number of balls is fixed beforehand. However, to attain this bound, we will discuss the Poisson distribution of balls, which selects the number of balls at random from a Poisson distribution and then places them in the buckets (independently and uniformly at random).

**Definition 28** (Fixed$[n]$ and Poisson$[n]$). We define Fixed$[n]$ and Poisson$[n]$ to be particular distributions of balls into $n$ buckets.

- **Fixed$[n]$**: $n$ balls are each independently deposited into a bucket uniformly at random.

- **Poisson$[n]$**: a number $k$ is drawn from the Poisson distribution with mean $n$; then, the balls are distributed according to Fixed$[k]$.

We note that due to the properties of the Poisson distribution, Poisson$[n]$ is equivalent to the following procedure: each bucket is independently assigned a number of balls drawn from the Poisson distribution with mean 1. In particular, this means that under this distribution, the number of balls in each bucket is independent.

**A.1.3 An Algorithm for Low-Cost Ball Removal (ouch!)**

We now present the main results of this appendix, which are used to construct the TSP algorithm:

**Theorem 8** (Main Problem). For any positive integer parameters $n, \gamma$ (where $\gamma \geq 2$),

$$\text{BCP}(n, \gamma, \text{Fixed}[n]) < 6$$

with very high probability (i.e. the probability that this bound is violated goes to 0 exponentially as $n \to \infty$).
We will show this through the following result:

**Proposition 8** (Poissonized Variant). For any positive integer parameters \( n, \gamma \) (where \( \gamma \geq 2 \)),

\[
\text{BCP}(n, \gamma, \text{Poisson}[n]) < 6
\]

with very high probability.

In particular, our results state that Strategy 1 (given in Section 3.1.4) succeeds with very high probability. We restate it here:

**Strategy 1.** Strategy 1 consists of 3 phases:

- **Initialization:** Collect once before the first merge;
- **Main Stages:** Collect \( 2^\gamma \) times after each of the \( \ell \) merges;
- **Cleanup:** Collect an additional \( 2^{(\gamma-1)\ell} \) times after the final merge.

Thus, this strategy corresponds to \( a_0 = 1, a_1 = 2^\gamma, a_2 = 2^\gamma, \ldots, a_\ell = 2^\gamma + 2^{(\gamma-1)\ell} \) in the notation established above; however, for the purposes of the analysis we will think of \( \ell \)th stage as being split into a ‘main’ part and a ‘cleanup’ part, as described.

### A.1.4 Analysis

We now prove Theorem 8, via Proposition 8. We do this in the following steps:

1. show that Strategy 1 has cost < 6;
2. show that Strategy 1 with high probability removes all the balls if they are distributed according to \( \text{Poisson}[n] \) (thus proving Proposition 8);
3. show that this implies that Strategy 1 with high probability removes all the balls if they are distributed according to \( \text{Fixed}[n] \) (thus proving Theorem 8).

#### Cost of Strategy 1

We first analyze the cost of Strategy 1 (which is fixed, since the strategy is blind).

**Lemma 15** (Cost of Strategy 1). The cost of Strategy 1 is at most 6.

**Proof.** We analyze the cost of each phase:

- **Initialization:** Total cost of 1, since we make one collection for a cost of 1.
- **Main Stages:** Total cost < 4, since at stage \( i \) we make \( 2^\gamma \) collections so the total cost is

\[
\sum_{i=1}^{\ell} 2^\gamma \cdot 2^{-(\gamma-1)i} = 2 \sum_{j=0}^{\ell-1} 2^{-(\gamma-1)j} < 2 \sum_{j=0}^{\infty} 2^{-(\gamma-1)j} = \frac{2}{1 - 2^{-(\gamma-1)}} \leq 4
\]

where the final inequality follows from \( \gamma \geq 2 \).

- **Cleanup:** Total cost of 1, since we make \( 2^{(\gamma-1)\ell} \) collections at a cost of \( 2^{-(\gamma-1)\ell} \) each.

Thus, the total cost of the whole strategy is less than 6. \( \square \)
APPENDIX A. ANALYZING THE ABSTRACTION

The Poisson Case: Definitions and Lemmas

We now need to argue that this strategy succeeds with high probability. We first note the following: at every stage, each bucket is the union of a subset of the original (stage 0) buckets, and these subsets are disjoint. Thus, since the number of balls in each bucket is independent at stage 0, it remains true at each stage $i$ (if the balls are distributed according to Poisson[$n$]). We now consider the following statistical value:

**Definition 29 (Exponential Moment).** For any random variable $W$, its exponential moment is

$$
\mu_{exp}[W] \triangleq \mathbb{E}[e^W].
$$

We characterize two useful properties of the exponential moment as follows; we use $(x)^+$ to denote $\max(x, 0)$:

**Lemma 16 (Properties of $\mu_{exp}$).** The exponential moment satisfies the following:

1. If $W_1, W_2, \ldots, W_k$ are independent random variables (with finite exponential moments), then

$$
\mu_{exp}\left[\sum_{j=1}^{k} W_j\right] = \prod_{j=1}^{k} \mu_{exp}[W_j].
$$

2. Let $W$ be a nonnegative integer random variable with finite exponential moment and $k$ be a nonnegative integer; note that $(W - k)^+$ is a random variable related to $W$. Then

$$
\mu_{exp}[(W - k)^+] \leq \frac{\mu_{exp}[W] - 1}{e^k} + 1,
$$

i.e. as we subtract more from a random variable, its exponential moment decreases exponentially towards 1.

**Proof.**

1. This follows trivially from the fact that the expected value of the product of independent random variables is the product of their expected values.

2. Let $w_j = \Pr[W = j]$ for all nonnegative integers $j$. Then, by definition:

$$
\mu_{exp}[W] - 1 = \sum_{j=0}^{\infty} w_j (e^j - 1).
$$

Note that when $W \leq k$, $(W - k)^+ = 0$; otherwise, $(W - k)^+ = W - k$. This means that $(W - k)^+ = 0$ with probability $w_0 + w_1 + \ldots + w_k$, and for all $j > k$,

$$
\Pr[(W - k)^+ = j - k] = w_j.
$$

Thus, putting this together, we get that:

$$
\mu_{exp}[(W - k)^+] - 1 = \sum_{j=0}^{k} w_j (e^0 - 1) + \sum_{j=k+1}^{\infty} w_j (e^{j-k} - 1) = \sum_{j=k+1}^{\infty} w_j (e^{j-k} - 1).
$$

However, this means that

$$
e^k (\mu_{exp}[(W - k)^+] - 1) = \sum_{j=k+1}^{\infty} w_j (e^j - e^k) \leq \sum_{j=0}^{\infty} w_j (e^j - 1) = \mu_{exp}[W] - 1
$$

where the inequality is due to both the addition of (nonnegative) missing terms and the fact that we are subtracting $e^k$ from each term on the left and only subtracting 1 on the right. Rearranging this inequality gives us

$$
\mu_{exp}[(W - k)^+] \leq \frac{\mu_{exp}[W] - 1}{e^k} + 1.
$$

We note that the proof of property 2 works in much the same way (except with an integral, instead of a sum) even if $W$ is not an integer random variable and $k$ is not an integer; however, we only need the restricted form to obtain our desired result.
The Poisson Case: Main Analysis

We now have the tools to tackle Proposition 8 directly.

Proof of Proposition 8. Since we already proved that Strategy 1 has a total cost of less than 6 in Lemma 15, if we can show it removes all the balls with high probability then we have proved Proposition 8. This is because the minimum cost of removing all the balls is by definition at most the cost of any given strategy which succeeds - thus, if Strategy 1 succeeds with very high probability, then BCP(n, γ, Poisson[n]) < 6 with very high probability.

We begin by defining some values which allow us to outline our proof; for these definitions, we assume that we are committing to Strategy 1. For any bucket J at stage i (‘J’ is capitalized to remind us that it stands for a stage-i bucket, i.e. the union of $2^j$ of the n original buckets), we define the following:

- $Y_0^J$ is the random variable denoting the number of balls in bucket J at the beginning.
- $Y_i^J$ is the random variable denoting the number of balls in bucket J just after the $i$th merge (i.e. at the beginning of stage $i$) for $i = 1, 2, \ldots, \ell$.
- $Z_i^J$ is the random variable denoting the number of balls in bucket J just before the $(i + 1)$th merge (i.e. at the end of stage $i$) for $i = 1, 2, \ldots, \ell$.
- $Z_\ell$ is the random variable denoting the number of balls at the beginning of the Cleanup phase (after the first $2^k$ collections at the $\ell$th stage); since there is only one bucket at this point, we omit the term J specifying the bucket.

Note that $Y_i^J$ (for all i, including 0) denotes the number of balls in bucket J at the beginning of stage i.

The idea behind Strategy 1 is to keep $\mu_{exp}[Z_i^J]$ small for each i (note that by symmetry, $\mu_{exp}[Z_i^J]$ is the same for each J), so that when we end the Main Stages phase, the final bucket has a probabilistically limited number of balls. In particular, the Initialization phase is exactly what is needed to get $\mu_{exp}[Z_0^J] \leq \epsilon$; then, given this, the Main Stages each inductively keep $\mu_{exp}[Z_i^J] \leq \epsilon$ for each $i = 1, 2, \ldots, \ell$ - specifically, $\mu_{exp}[Z_\ell] \leq \epsilon$; finally, the Cleanup phase removes the remaining balls (with very high probability).

To facilitate the proof, we define the following values (recalling that by symmetry, the random variables $Y_i^J$ and $Z_i^J$ are identical across all buckets J for each level-i so these definitions apply for all level-i buckets J):

$$y_i := \mu_{exp}[Y_i^J] \text{ and } z_i := \mu_{exp}[Z_i^J].$$

We first note that any stage-$(i + 1)$ bucket is the union of $2^i$ stage-i buckets, which contain independent numbers of balls. Thus, by property 1 from Lemma 16, for any $i = 0, 1, \ldots, \ell - 1$,

$$y_{i+1} = (z_i)^{2^i}. \quad (A.1)$$

Furthermore, by definition $Z_i^J = (Y_i^J - 2^i)^+$ (except for when $i = 0$, since stage 0 has only one collect operation, so $Z_0^J = (Y_0^J - 1)^+$). Thus, for all $i = 0, 1, \ldots, \ell - 1$, by property 2 from Lemma 16,

$$z_{i+1} \leq \frac{y_{i+1} - 1}{e^{2^i}} + 1. \quad (A.2)$$

Putting expressions (A.1) and (A.2) together gives the result that

$$z_{i+1} \leq \frac{(z_i)^{2^i} - 1}{e^{2^i}} + 1,$$

which gives us the following statement: for all $i = 0, 1, \ldots, \ell$,

$$z_i \leq \epsilon \implies z_{i+1} \leq \frac{(z_i)^{2^i} - 1}{e^{2^i}} + 1 \leq \frac{e^{2^i} - 1}{e^{2^i}} + 1 < 2 \implies z_{i+1} \leq \epsilon.$$

Thus, by induction, we know that $z_0 \leq \epsilon \implies z_\ell \leq \epsilon$. 

To prove that $z_0 \leq e$ in the first place, we simply recall that the distribution Poisson$n$ is equivalent to having each bucket independently assigned a number of balls drawn from the Poisson distribution with mean $1$ (which we denote Pois$(1)$); hence, by definition of the exponential moment,

$$y_0 = \mu_{\text{exp}[\text{Pois}(1)]} = \sum_{k=0}^{\infty} \frac{e^k}{k!} = e^{\sum_{k=0}^{\infty} \frac{1}{k!} e^k} = e^{e - 1}$$

as the value of the summation in the fourth expression in the preceding line is the classic infinite-sum definition of the exponential. Thus (again, by property 2 from Lemma 16):

$$z_0 \leq \frac{e^{e - 1} - 1}{e} + 1 < e$$

(for those who are curious, $e^{e - 1} - 1 + 1 \approx 2.683$ which is close to but still less than $e$). Thus, $z_0 \leq e$.

We have now concluded that at the start of the Cleanup phase, there is one large bucket left with a random number $Z_\ell$ of balls left such that $\mu_{\text{exp}[Z_\ell]} = z_\ell \leq e$. We then note that since Cleanup consists of $2^{(\gamma - 1)\ell}$ collections at the $\ell$th stage, the final number of balls (which we denote by the random variable $U$) is $U = (Z_\ell - 2^{(\gamma - 1)\ell})^+$. The strategy succeeds if and only if $U = 0$. However, since $U$ is by definition a nonnegative integer random variable, we can define the event of the strategy failing by the following:

$$U \neq 0 \iff e^U - 1 \geq e - 1.$$ 

While this may seem a strange way to define failure, it allows us to take advantage of the Markov Inequality. In particular, by property 2 of Lemma 16 and our result that $z_\ell \leq e$,

$$\mathbb{E}[e^U - 1] = \mu_{\text{exp}[U]} - 1 \leq \left(\frac{z_\ell - 1}{e^{2^{(\gamma - 1)\ell}}}\right) + 1 - 1 \leq \frac{e - 1}{e^{2^{(\gamma - 1)\ell}}}.$$ 

But then by the Markov Inequality, since $e^U - 1$ is a nonnegative random variable (since $U$ is nonnegative), we can conclude that

$$\Pr[e^U - 1 \geq e - 1] \leq \frac{e - 1}{e^{2^{(\gamma - 1)\ell}}} \cdot \frac{1}{e - 1} = \frac{1}{e^{2^{(\gamma - 1)\ell}}}.$$

To see that this does approach $0$ exponentially fast as $n \to \infty$, we can simply recall that since $n = 2^{\gamma \ell}$, the upper bound on the probability of failure is

$$\frac{1}{e^{2^{(\gamma - 1)\ell}}} = e^{-\left(\frac{2^{(\gamma - 1)\ell}}{\gamma}ight)}$$

thus concluding the proof. \(\square\)

**The Fixed-Number Case**

Finally, we can prove Theorem 8; to do this, we first define one more distribution of balls into buckets:

**Definition 30 (The One-Fourth Distribution $H[n]$).** We define the distribution $H[n]$ of balls into $n$ buckets as follows:

- with probability $1/4$, Fixed$[n]$ is executed (i.e. $n$ balls are distributed into the $n$ buckets uniformly and independently at random);
- otherwise (with probability $3/4$) no balls at all are placed into the buckets.

Equipped with this definition, we can now show Theorem 8.

**Proof of Theorem 8.** We note that the only difference between the three distributions Fixed$[n]$, Poisson$[n]$ and $H[n]$ is how they generate the number of balls to be distributed; once this number is generated, the balls are always placed uniformly and independently into the buckets. Under this general rule, if the number of balls generated under one scheme $F'$ stochastically dominates the number of balls generated under
another scheme \( F'' \), then the random variable \( BCP(n, \gamma, F') \) stochastically dominates the random variable \( BCP(n, \gamma, F'') \).

However, we can then note that for all \( n \geq 4 \), the probability that a Poisson random variable with mean \( n \) exceeds \( n \) is at least \( 1/4 \); thus, the Poisson random variable stochastically dominates the random variable which is \( n \) with probability \( 1/4 \) and \( 0 \) otherwise. Thus, by the above

\[
BCP(n, \gamma, \text{Poisson}[n]) \text{ stochastically dominates } BCP(n, \gamma, H[n]),
\]

which in particular implies that

\[
\Pr[BCP(n, \gamma, \text{Poisson}[n]) > 6] \geq \Pr[BCP(n, \gamma, H[n]) > 6].
\]

But we note that \( 1/4 \) of the time, running distribution \( H[n] \) is just simulating \( \text{Fixed}[n] \), and the rest of the time no balls are placed in the bucket so total removal cost is 0. Thus,

\[
\Pr[BCP(n, \gamma, H[n]) > 6] = \frac{1}{4} \cdot \Pr[BCP(n, \gamma, \text{Fixed}[n]) > 6].
\]

Putting these two statements together yields

\[
4 \cdot \Pr[BCP(n, \gamma, \text{Poisson}[n]) > 6] \geq \Pr[BCP(n, \gamma, \text{Fixed}[n]) > 6].
\]

But by Proposition 8, the first probability in this inequality goes to 0 exponentially as \( n \to \infty \); thus, the second probability (which is the one we are interested in) must do the same, concluding our proof. \( \square \)

While we have shown that the minimum cost of removing all the balls in the Fixed case is indeed at most 6 with high probability, we would also like to know if there is a strategy which achieves it; however, an identical stochastic dominance argument shows that since Strategy 1 works in the Poisson case with very high probability, it works with very high probability in the Fixed case as well. Thus, we have not only shown that a low-cost strategy exists, Strategy 1 achieves this.

**Remark:** We note that it is impossible to do better than constant cost as \( n \) tends to infinity, so Strategy 1 is a constant-factor approximation of the optimal ball-removal cost (with very high probability). In fact, it is never possible to remove all \( n \) balls for a cost of less than 1. This is due to the fact that each collection operation can only remove as many balls as there are buckets - and the number of buckets decreases faster than the cost (a factor of \( 2^i \) against a factor of \( 2^{i-1} \) at each merge). Thus, each ball costs at least \( 1/n \) to remove (the cost of removing a ball is defined here to be the cost of the collection step it was removed in, divided by the total number of balls removed during that step), so the cost of removing all the balls is at least 1.
Appendix B

Bounding Binomials

In order to generalize our technique from a very strict case (uniform distribution of targets on a hypercube) to a more flexible one, we divided the space into finitely many regions which approximated our strict assumptions, and analyzed each piece separately. However, this introduces a subtle change in the problem: now there is not a fixed number of target points, but a binomially distributed random number of such points for each piece (specifically, if piece $P_j$ covers $p_j$ of the probability mass of the target point distribution $F$, then the number of targets that falls into $P_j$ is distributed according to bin$(n, p_j)$). However, for simplicity we assumed that for large $n$ this is effectively the same thing; here, we will rigorously justify this simplification.

We will only be discussing general vehicles here; our arguments (specifically Cost-Balanced Orienteering and the algorithm dividing the whole space up into cells) for the symmetric case have already covered this for us. We will also talk a lot about “very high probability”, whose formal definition was given in Definition 16. However, for ease of reading, we will restate it here.

**Definition (Very High Probability).** Let $A(n)$ be a random event parametrized by the value $n$. Then $A(n)$ occurs with very high probability if there is some $\xi > 0$ such that

$\Pr[A(n) \text{ doesn't happen}] \leq e^{-\xi n}$

We will prove our correctness in two ways: (1) quick and dirty (but sufficient to obtain our results), and (2) with a bit more precision.

**B.1 Quick and Dirty**

Because we are only determining dependence of TSP$_\Pi(X_1, \ldots, X_n)$ on the number of target points $n$, our bounds can be extremely loose in terms of constant factors.

**B.1.1 Upper Bound**

To get our upper-bound, we consider the scenario where $n$ target points are distributed on every piece (which is obviously an upper bound on the total length of the tour). Since each piece will require (with very high probability) a tour length of $\Theta(n^{\frac{2}{\gamma} + 1})$ and there are a fixed finite number of pieces, we can use the union bound to show that with very high probability the whole tour (which is upper-bounded by the sum of the tours on the pieces, plus a constant to get from piece to piece) will take a trajectory of length $O(n^{\frac{2}{\gamma} + 1})$.

**B.1.2 Lower Bound**

To get our lower bound, we simply consider one piece $P$, and define $p = \Pr_{X \sim F}[X \in P] > 0$. Therefore, the number of target points which are distributed on $P$ is a random variable with distribution bin$(n, p)$, which has expected value $np$. The probability that we get at least $\frac{n p}{2}$ as $n$ gets large is very high (well-known property of the binomial distribution). Therefore, if we distribute our $n$ target points one at a
time \((X_1, \text{then } X_2, \text{etc.})\) we can ask for a trajectory which visits the first \(\frac{np}{2}\) targets to land in \(P\); these will be distributed independently randomly according to \(X \sim (F|X \in P)\) (or in other words \(F\) restricted to \(P\)). By our piece-by-piece analysis, the shortest trajectory visiting all of these target points will have length \(\Theta\left(\left(\frac{np}{2}\right)^{\frac{\gamma - 1}{\gamma}}\right) = \Theta\left(n^{\frac{\gamma - 1}{\gamma}}\right)\) (since \(p\) is a constant), and therefore the tour through all \(n\) target points is lower-bounded by this.

**B.2 More Satisfying Technique**

We can also do this using the fact that the number of targets that falls on each piece \(P_j\) is a binomial, as described above, with expectation \(np_j\) (where \(p_j = \Pr_{X \sim F}[X \in P_j]\)). This method produces a “better” bound in the sense that it doesn’t require the obvious inefficiencies and redundancies of the first method; nevertheless, it doesn’t produce any sort of guarantee on how well it approximates the ‘true’ constant factor. We include it here merely for completeness.

It is well known that as \(n \to \infty\), the binomial distribution begins to approximate the normal distribution. Specifically:

\[
\frac{\text{bin}(n, p_j) - np_j}{\sqrt{np_j}} \to \mathcal{N}(0, 1) \text{ as } n \text{ goes to } \infty
\]

(formally, it “converges in distribution”, but the distinctions between various ways distributions can converge does not affect this analysis).

The point is that as \(n \to \infty\), the number of targets that will fall into piece \(P_j\) is approximately \(np_j\), with a \(\Theta(\sqrt{n})\) deviation. We are still going to be a little crude about it, by using a constant-factor deviation instead of the \(\sqrt{-}\) specifically, for any \(\delta > 0\), there are at most \((1 + \delta)np_j\) target points in region \(P_j\) with very high probability, i.e. the probability of having more than \((1 + \delta)np_j\) shrinks exponentially to 0 as \(n \to \infty\). Because there are only finitely many regions, we can apply the Union Bound to show that the probability that any region has more than this many points goes to 0 exponentially fast. Therefore, with very high probability, we can drop \((1 + \delta)np_j\) (that is, a fixed number) target points into each region \(P_j\) (distributed according to \(F\) restricted to \(P_j\)). The length of the shortest tour through these points will serve as a with-high-probability stochastic upper bound, since with very high probability each piece has more targets in this scenario than in the original \(n\) i.i.d. target points scenario (and each target point that falls in \(P_j\) in the original case can be thought of as distributed according to \(F\) restricted to \(P_j\)).

We can also get a lower bound by distributing \((1 - \delta)np_j\) target points for each piece. The logic is exactly the same.

Therefore, we apply our solution to a finite collection of pieces, each of which has \((1 + \delta)p_jn\) (for the upper bound) or \((1 - \delta)p_jn\) (for the lower bound) target points in it, which then translates into a tour length of

\[
\Theta\left(\sum_{j=1}^{m} ((1 - \delta)np_j)^{\frac{\gamma - 1}{\gamma}}\right) \leq \text{TSP}_n(X_1, \ldots, X_n) \leq \Theta\left(\sum_{j=1}^{m} ((1 + \delta)np_j)^{\frac{\gamma - 1}{\gamma}}\right)
\]

which is just \(\Theta(n^{\frac{\gamma - 1}{\gamma}})\).
Appendix C

Curious Counterexamples

In this chapter we give some examples showing why certain assumptions are necessary for our theorems.

C.1 Lesser-Dimensional $F$

An important assumption for our results is that the distribution $F$ is full-dimensional. If this is violated, then all sorts of failures can happen; the simplest is where $F$ is just a point-mass distribution at one point (i.e. all targets $X_i$ are “randomly” selected to be a specific $x \in \mathcal{X}$), at which point the vehicle can visit all the points with a trajectory of 0. However, there are a number of subtler cases that shed light on the behavior of the problem, and even one which suggests a related open problem of some interest.

C.1.1 Exploding Tour Length for General Vehicles

Restricting $F$ to non-full-dimensional sets can also lead to trouble in the other direction for general vehicles – the expected tour length can grow to $\Theta(n)$. To show this, we consider the Dubins car in $\mathbb{R}^2$ with a curvature bound of 1. We now suppose that the target points are uniformly distributed on a circle of radius 0.001 (centered at the origin), which we will call $S$. When the vehicle passes through the $S$, it can only intersect it at two points (or pass tangent and intersect at one point), and therefore can only collect two target points before having to circle around with a trajectory of length at least 3 (really the shortest length possible would have it be close to the numerical value $\pi$, but because $S$ is not a point we lower bound this by 3 to be safe). Therefore, the length of the optimal trajectory is lower-bounded by $\frac{3}{2}(n - 2)$ (since it can start out on $S$ and pick up two points very quickly).

Remark: Note that this is not the case for symmetric vehicles – restricting the dimensionality of $F$ cannot cause the rate of growth of the tour length to increase. We can easily show this by applying the bounds we derived when the target points are distributed adversarially (Theorem 4). In fact, as we showed in Section 5.1.2, this implies that the rate of growth of the tour’s length in this case is strictly lower than for full-dimensional $F$.

C.2 Unbounded $S$

One of the important assumptions was that $S$ be bounded, so that a polynomially-sized (in $n$) set of “initial configurations” could cover the whole space, allowing us to establish the lower bound.

Consider a constraint set $\Pi$ with workspace $\mathcal{X} = \mathbb{T}^2$ (the two-dimensional torus, which we’ll represent as $[0, 1]^2$) and configuration space $\mathcal{Z} = \mathbb{T}^2 \times \mathbb{R}$. To be fully rigorous, we would have to embed $\mathcal{X}$ in $\mathbb{R}^3$ and $\mathcal{Z}$ in $\mathbb{R}^4$; but to make things simpler to understand, we’ll just represent $\mathbb{T}^2$ as $[0, 1]^2$ with the understanding that going above 1 in either axis takes you back to 0. We therefore represent a configuration as a point in $[0, 1]^2 \times \mathbb{R}$.
If takes in one control input $u \in [-1, 1]$ and is defined by the following: $\dot{z} = (1, z_3, 0) + u \cdot (0, 0, 1)$. In short, the vehicle always advances along the $x$-axis of the workspace at uniform speed; its configuration is its location plus its speed along the $y$-axis, which it can alter at uniform speed. Note that the length of the trajectory is not the distance it traverses in the 2-dimensional torus, but only the distance it travels on the $x$-axis.

The target points are uniformly distributed over the workspace.

Indeed, this situation satisfies all our constraints except that $S$ is unbounded, since it consists of the whole real line. A quick examination shows that the reachable set always has size $\text{vol}(\tilde{R}(z)) = \Theta(e^3)$, similar to a Dubins Car (in fact, the reachable set $\tilde{R}(z)$ always has a volume of precisely $e^3/3$). And yet, there is almost surely a trajectory of length 1 which visits all $n$ points! Let our target points be $X_1, X_2, \ldots, X_n$ with $X_i = (x_i, y_i)$; and suppose that $x_i \neq x_j$ for all $i \neq j$ (which happens with probability 1). Then, for any $\delta > 0$, there exists a line represented by the function $h(x) = ax + b$ such that for all $i$, the distance between $[h(x_i)] := h(x_i) - \lfloor h(x_i) \rfloor$ (the fractional part of $h(x_i)$) and $y_i$ is at most $\delta$, i.e. they can be made arbitrarily close.

We now also have to show that the existence of such a line implies that the vehicle can reach all points with a trajectory of length 1. Because each point has a different $x$-axis value, we can define $\epsilon = \min_{i \neq j}(|x_i - x_j|)$. We now consider what happens when starting from some $z$, we apply the control $u = +1$ for a distance of $\lambda$, then $u = -1$ for a distance of $2\lambda$, and finally $u = +1$ for another distance of $\lambda$; the result is that at the end ($t = 4\lambda$) we arrive at the same place as if we had applied no control, but at the midpoint ($t = 2\lambda$) reach a point which is $\lambda^2$ different above where we would have been had we had a control of 0 for the whole time. To show this, we note that we can make this computation when the starting velocity on the $y$-axis is 0, that is $Z_3 = 0$ at time 0, and the conclusion applies for all starting configurations. This is for the following reason: suppose we have it so that at $t = 0$, $Z_3 = v$; we can then run the vehicle from the same position except with $Z_3 = 0$ and add $vt$ from the position of the vehicle (on the $y$-axis) at time $t$ to obtain its position. We also note that we can go below the line by negating the controls given above.

Thus, suppose we have some target point $X_i = (x_i, y_i)$. By the definition of $\epsilon$, we know that no other target point has an $x$-coordinate in the range $(x_i - \epsilon, x_i + \epsilon)$. We also know that we have our line $h(x) = ax + b$ where $|h(x_i) - y_i| \leq \delta$ (here we are considering only the small portion of the line which passes by $x_i$, and therefore we are removing the integer component and considering only the fractional one). Thus, use the above deviation strategy to visit $X_i$, with $\lambda = \sqrt{\delta}$ and beginning at $x_i - 2\lambda$ and ending at $x_i + 2\lambda$. As long as $2\lambda \leq \epsilon/2$, none of the deviations "clash" with one another (we return to the line before the next deviation begins).

So, as long as we have $\delta \leq \epsilon^2/16$, this is possible. But because $\epsilon > 0$ is fixed once the target points are chosen, and we can find a line which achieves as small a $\delta$ as we like, we are done. Thus, we can find a trajectory of length 1 which visits all $n$ target points with probability 1.

---

1formally, we have to deal with the possibility that they may be close because one is close to 1 and the other close to 0; we can do this by saying that for all i, $\min(|[h(x_i)] - y_i|, |1 - |[h(x_i)] - y_i||) \leq \delta$