Special Issue: RSS2023

Non-Euclidean motion planning with graphs of geodesically convex sets

Thomas Cohn[1], Mark Petersen[2], Max Simchowitz[1] and Russ Tedrake[1]

The International Journal of Robotics Research
2024, Vol. 0(0) 1–23

DOI: 10.1177/02783649241302419
journals.sagepub.com/home/ijr
Sage

Abstract

Computing optimal, collision-free trajectories for high-dimensional systems is a challenging and important problem. Sampling-based planners struggle with the dimensionality, whereas trajectory optimizers may get stuck in local minima due to inherent nonconvexities in the optimization landscape. The use of mixed-integer programming to encapsulate these nonconvexities and find globally optimal trajectories has recently shown great promise, thanks in part to tight convex relaxations and efficient approximation strategies that greatly reduce runtimes. These approaches were previously limited to Euclidean configuration spaces, precluding their use with mobile bases or continuous revolute joints. In this paper, we handle such scenarios by modeling configuration spaces as Riemannian manifolds, and we describe a reduction procedure for the zero-curvature case to a mixed-integer convex optimization problem. We further present a method for obtaining approximate solutions via piecewise-linear approximations that is applicable to manifolds of arbitrary curvature. We demonstrate our results on various robot platforms, including producing efficient collision-free trajectories for a PR2 bimanual mobile manipulator.

Keywords

Motion planning, optimization, Riemannian geometry

Received 10 January 2024; Revised 3 October 2024; Accepted 23 October 2024

1. Introduction

Planning the motion of robots through environments with obstacles is a long-standing and ever-present problem in robotics. In this paper, we aim to find the shortest path between a start and goal configuration with guaranteed collision avoidance. We are particularly motivated by planning for bimanual mobile manipulators, such as the PR2 (Willow Garage). Such robots are well suited for a variety of tasks in human environments but present various challenges for existing motion planning algorithms.

Most popular approaches for this task fall into two categories: sampling-based planners and trajectory optimizers. The trajectory optimization problem is inherently nonconvex when there are obstacles in the scene, so solvers frequently get stuck in local minima. In that case, they may output a path that is longer than the global optimum or even fail to produce a valid path even when one exists.

On the other hand, sampling-based planners can avoid getting stuck in local minima, but the path may be locally suboptimal, resulting in jerky and uneven motion. Sampling-based planners may also suffer from the so-called "Curse of Dimensionality." Because they rely on covering the configuration space with discrete samples, in the worst case, the number of samples required may increase exponentially with the dimension. The PR2 has two 7-DoF arms and a mobile base, and sampling-based planners struggle with the instances we study here.

Recently, Marcucci et al. (2023) described a new type of motion planning, based on a decomposition of the collision-free subset of configuration space (C-Free) into convex sets. They leverage a new optimization framework, a *Graph of Convex Sets* (GCS), where each vertex is associated with a convex set and each edge is associated with a convex function (Marcucci et al., 2021). Motion planning becomes a shortest-path problem in this space. This *GCS Trajectory Optimization* approach (abbreviated as *GcsTrajOpt*) has been successfully applied to challenging, high-dimensional problems, including bimanual manipulation problems.

[1] Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
[2] School of Engineering and Applied Sciences, Harvard, Cambridge, MA, USA

Corresponding author:
Thomas Cohn, Computer Science and Artificial Intelligence Laboratory, MIT, 32 Vassar Street, Rm 380, Cambridge, MA 02139, USA.
Email: tcohn@mit.edu

However, GcsTrajOpt is limited to Euclidean configuration spaces. A mobile manipulator's configuration space is inherently non-Euclidean due to the mobile base: the robot can rotate through a full 360°, and its configuration is identical to when it started. Continuous revolute joints present a similar issue. Although the configuration spaces of interest are inherently non-Euclidean, they are still "locally" Euclidean, leading to elegant descriptions as differentiable manifolds. With a Riemannian metric, which allows one to measure distance on a manifold, the concepts of convexity generalize to nonlinear spaces. This in turn allows optimization on manifolds with rigorous guarantees, analogous to those obtained from convex optimization on Euclidean spaces.

A conference version of this paper is published in Cohn et al. (2023). This manuscript extends the results to more general classes of robot configuration spaces via a piecewise-linear approximation. This encompasses closed kinematic chains, whose configuration spaces can naturally be represented as the (generally nonconvex) feasible set of a system of nonlinear equality constraints. This also allows planning over the group of 3D rotations, enabling our method to handle ball joints and free bodies. (The approximation strategy is required in this case due to the inherent nonconvexity of the distance function; see Subsection 6.5 for details).

In this paper, we formulate the general problem of shortest-path motion planning around obstacles on Riemannian manifolds. We define a graph of *geodesically* convex sets (GGCS), the analogue to GCS on a Riemannian manifold. We prove that this formulation has all the requisite properties needed to inherit the same guarantees as (Euclidean) GCS for a certain class of robot configuration spaces, encompassing open kinematic chains with continuous revolute joints and mobile bases. In this case, our theoretical developments lead to simple and elegant modifications to the original GcsTrajOpt. We entitle this generalization *GGCS Trajectory Optimization* (abbreviated as *GgcsTrajOpt*), and demonstrate its efficacy with several challenging motion planning experiments.

2. Related work

In the world of continuous motion planning around obstacles, most popular techniques fall into two categories: sampling-based planners and trajectory optimizers.

Sampling-based motion planners partially cover C-Free with a large number of discrete samples. Two of the foundational sampling-based planning algorithms are Probabilistic Roadmaps (PRMs) (Kavraki et al., 1996) and Rapidly-Exploring Random Trees (RRTs) (LaValle, 1998). Such algorithms are probabilistically complete, that is, with enough samples, they will always find a valid path (if one exists). However, these algorithms are only effective if a valid plan can be produced with a reasonable number of samples. Hence, the "curse of dimensionality" is a potential obstacle to sampling-based planning, and such techniques have struggled with high-dimensional problems such as bimanual manipulation. In most cases, planning for bimanual tasks is accomplished by planning for one arm, then planning the second arm independently while treating the first arm as a dynamic obstacle. This is a reasonable heuristic for some tasks, but it sacrifices even probabilistic completeness.

An alternative approach is to formulate motion planning as an optimization problem. This requires parametrizing the space of all trajectories and defining constraints and cost functions that describe the suitability of each trajectory. Examples of kinematic trajectory optimization include B-spline parametrizations using constrained optimization (Tedrake, 2022, §7.2), CHOMP (Zucker et al., 2013), STOMP (Kalakrishnan et al., 2011), and KOMO (Toussaint, 2014). Trajectory optimization approaches do not suffer from the curse of dimensionality, and are suitable for much more complex robotic systems. But the optimization landscape is inherently nonconvex, so trajectory optimization methods cannot guarantee global optimality, and often fail to produce feasible trajectories altogether.

The use of mixed integer programming (MIP) to solve motion planning problems to global optimality has recently seen an increase in popularity as new theoretical results, greater computational resources, and powerful commercial solvers (Gurobi Optimization LLC, 2023; MOSEK ApS, 2019) have been brought to bear. The survey paper of Ioan et al. (2021) provides an overview of the use of MIP in motion planning. Besides the work of Marcucci et al. (2021), Landry et al. (2016) used MIP to plan aggressive quadrotor flights through obstacle-dense environments. MIP has been used to plan footstep locations for humanoid robots (Deits and Tedrake, 2014) and for quadrupeds (Aceituno-Cabezas et al., 2017; Valenzuela, 2016). Dai et al. (2019) used MIP to globally solve the inverse kinematics problem. Finally, MIP has seen extensive use in hybrid task and motion planning (Adu-Bredu et al., 2022; Chen et al., 2022; Saha and Julius, 2017; Tika et al., 2022; Yi et al., 2022).

Mixed integer programs can take a long time to solve in the worst case, but it is often possible to mitigate this problem with appropriate relaxations or approximations (Marcucci et al., 2023; Suh et al., 2020). GCS in particular uses an MIP formulation with a small number of integer variables, making branch-and-bound tractable. Furthermore, the convex relaxation is tight, enabling efficient approximation by solving only a convex problem combined with a randomized rounding strategy. Marcucci et al. (2023) argued that for single-arm manipulators, this approach can find more optimal plans in less time than PRMs. These valuable properties carry over to our extension of GCS.

Another recent trend in motion planning has been the use of Riemannian geometry to model the problem. Riemannian Motion Policies (RMPs) (Ratliff et al., 2018) combine acceleration-based controllers across different task spaces into a single unified controller. A Riemannian metric in each task space determines the priority of a given controller, and smooth maps between the manifolds enable the averaging of controllers. RMPs have seen continued improvement (Cheng et al., 2018; Rana et al., 2021) and generalization (Bylard et al., 2021; Van Wyk et al., 2022). However, RMPs and their extensions are still known to struggle with local minima, similar to nonconvex trajectory optimizers. Klein et al. (2022) envision Riemannian geometry as a tool for generating and combining elegant motion synergies for complex robotic systems.

One can solve optimization problems in these spaces with rigorous guarantees by using a generalization of convexity to Riemannian manifolds, called *geodesic convexity* (g-convexity). Unfortunately, existing research into g-convex optimization often focuses on specific classes of manifolds that do not encompass the configuration spaces of interest (Bacak, 2014; Zhang and Sra, 2016). In addition, there is little existing literature studying mixed-integer Riemannian convex optimization, and techniques commonly used in the Euclidean case, such as cutting planes (Marchand et al., 2002), may not generalize to Riemannian manifolds.

3. Preliminaries

In this section, we cover some of the relevant mathematical background. We supply intuitive definitions; for further reference on Riemannian geometry, see the textbooks of Do Carmo (1992) and Lee (2013, 2018). Boumal (2022) provides an excellent treatment of optimization over manifolds. We use the notation $[n] = \{1, \ldots, n\}$.

3.1. Riemannian geometry

A d-dimensional (topological) *manifold* $\mathcal{M}$ is a locally Euclidean topological space: for any $p \in \mathcal{M}$, there is an open neighborhood $\mathcal{U}$ of p and a continuous map $\psi : \mathcal{U} \to \mathbb{R}^d$ which is a homeomorphism onto its image. The pair $(\mathcal{U}, \psi_{\mathcal{U}})$ is called a *coordinate chart*, and for any pair of overlapping charts $(\mathcal{U}, \psi_{\mathcal{U}})$ and $(\mathcal{V}, \psi_{\mathcal{V}})$, we have a *transition* map

$$\tau_{\mathcal{U},\mathcal{V}} = \psi_{\mathcal{V}} \circ \psi_{\mathcal{U}}^{-1}\big|_{\psi_{\mathcal{U}}(\mathcal{U}\cap\mathcal{V})}. \tag{1}$$

A collection of charts whose domains cover the manifold is called an *atlas*. We only consider $\mathcal{C}^\infty$*-smooth* manifolds, where all transition maps in the atlas are $\mathcal{C}^\infty$.

For each $p \in \mathcal{M}$, the *tangent space* $\mathcal{T}_p\mathcal{M}$ is a d-dimensional vector space representing the set of directional derivatives at p. Given a differentiable curve $\gamma : (-\epsilon, \epsilon) \to \mathcal{M}$ with $p = \gamma(0)$, this affords an interpretation of the *velocity* of γ at p, $\gamma'(0)$, as an element of $\mathcal{T}_p\mathcal{M}$. For a smooth map of manifolds $f : \mathcal{M} \to \mathcal{N}$, the *pushforward* of f at p is a linear map $f_{*,p} : \mathcal{T}_p\mathcal{M} \to \mathcal{T}_{f(p)}\mathcal{N}$, generalizing the Jacobian matrix (Lee, 2013: 55). The pushforward is defined so that, with γ defined as above, $f_{*,p}\,(\gamma'(0)) = (f \circ \gamma)'(0)$.

A *Riemannian metric* g is a smoothly-varying positive-definite bilinear form over $\mathcal{M}$ that gives each tangent space $\mathcal{T}_p\mathcal{M}$ an inner product $\langle \cdot, \cdot \rangle_p^{(\mathcal{M},g)}$. This inner product induces a norm

$$||v||_p = \sqrt{\langle v, v\rangle_p^{(\mathcal{M},g)}} \tag{2}$$

for $v \in \mathcal{T}_p\mathcal{M}$. The pair $(\mathcal{M}, g)$ is a *Riemannian manifold*, and we frequently refer to $\mathcal{M}$ exclusively when the choice of metric is unambiguous. (Similarly, we drop the superscript from the Riemannian inner product and norm when the choice of manifold and metric is unambiguous.) A Riemannian metric allows one to measure the length of a curve, invariant to reparametrizations (Lee, 2018: 34); if $\gamma : [a, b] \to \mathcal{M}$ is piecewise continuously differentiable, then

$$L(\gamma) = \int_a^b \sqrt{\langle \gamma'(s), \gamma'(s)\rangle_{\gamma(s)}^{(\mathcal{M},g)}}\,\mathrm{d}s. \tag{3}$$

We call the integrand the *speed* of γ. The distance between any two points $p, q \in \mathcal{M}$ is defined as the infimum of the arc length of all curves connecting them,

$$\begin{aligned}&\mathrm{d}_{\mathcal{M}}(p,q) = \\ &\inf\left\{L(\gamma) | \gamma \in \mathcal{C}^1_{\mathrm{pw}}([0,1],\mathcal{M}), \gamma(0) = p, \gamma(1) = q\right\},\end{aligned} \tag{4}$$

where $\mathcal{C}^1_{\mathrm{pw}}([0,1],\mathcal{M})$ is the set of parametric piecewise-continuously differentiable curves from the interval [0, 1] to $\mathcal{M}$. A curve that achieves this infimum need not exist in general (Do Carmo, 1992: 146). We also define $\mathrm{d}_{\mathcal{U}}(p, q)$ for $p, q \in \mathcal{U} \subseteq \mathcal{M}$ to be the infimum of the length of paths whose image is contained in $\mathcal{U}$.

If $\mathcal{M}$ is connected, it is a metric space with respect to $\mathrm{d}_{\mathcal{M}}$. Given two Riemannian manifolds $(\mathcal{M}, g)$ and $(\mathcal{N}, h)$, a smooth function $f : \mathcal{M} \to \mathcal{N}$ is a *local isometry* if

$$\langle u, v\rangle_p^{(\mathcal{M},g)} = \left\langle f_{*,p}(u), f_{*,p}(v)\right\rangle_{f(p)}^{(\mathcal{N},h)} \tag{5}$$

$\forall p \in \mathcal{M}$, $\forall u, v \in \mathcal{T}_p\mathcal{M}$. If f is also a diffeomorphism, and $\mathcal{M}$ and $\mathcal{N}$ are connected, then f preserves distances (Lee, 2018: 37), and is an *isometry* of metric spaces. The converse is also true (Myers and Steenrod, 1939).

A *geodesic* is a locally length-minimizing curve, parameterized to be constant speed. Locally length-minimizing means that for two points on the geodesic that are close enough, the geodesic traces out the shortest path between them. For example, geodesics in Euclidean space with the natural metric are lines, rays, and line segments, and geodesics on the sphere (with the induced metric from Euclidean space) are great circles. Constructing the shortest geodesic between two points is a variational calculus problem, so the solution must satisfy the Euler–Lagrange system of differential equations. Thus, initial conditions $p \in \mathcal{M}$ and $v \in \mathcal{T}_p\mathcal{M}$ uniquely define a geodesic, such that v is the velocity of the geodesic as it passes through p. This is used to define the *exponential map* $\exp_p : \mathcal{T}_p\mathcal{M} \to \mathcal{M}$, where the direction of a vector v defines the direction of the geodesic, and the magnitude of v determines how far to move in that direction away from p.

A Riemannian metric induces *curvature* on a manifold, capturing how local geometry differs from the standard Euclidean case. The *sectional curvature* at a point p is a real-valued function defined on two-dimensional subspaces of the tangent space $\mathcal{T}_p\mathcal{M}$ (Do Carmo, 1992, §4.3). (We write $\mathcal{K}(u, v)$ for any vectors u and v that span the subspace). Informally, the sectional curvature corresponds to the distortion of angles in triangles, as shown in Figure 1. Manifolds that have everywhere-zero curvature are called *flat*, and are locally isometric to Euclidean space.

The Cartesian product of two Riemannian manifolds is itself a Riemannian manifold. The curvature of the component manifolds influences the curvature of the product. Importantly, *the product of flat manifolds is flat* (Atceken and Keles, 2003). As we explain in Section 5, this implies that a robot with a mobile base and (potentially many) continuous revolute joints has a flat configuration space.

3.2. Convex analysis on manifolds

To define convexity on a Riemannian manifold $(\mathcal{M}, g)$, we replace the notion of lines with geodesics. In general, there is not a unique geodesic (or even a unique shortest geodesic) between two points, so a more intricate definition is required. A subset $\mathcal{U} \subseteq \mathcal{M}$ is *strongly geodesically convex* (or *g-convex*) if $\forall p, q \in \mathcal{U}$, there is a unique length-minimizing geodesic connecting p and q, and it is entirely contained in $\mathcal{U}$. This definition ensures that the intersection of g-convex sets is g-convex, and that there is a unique shortest path between any pair of points in a g-convex set. Weaker definitions used in other works (Vishnoi, 2018; Zhang and Sra, 2016) do not provide these guarantees. Boumal (2022, §11.3) presents a detailed comparison of the various definitions of geodesic convexity.

G-convex neighborhoods exist around every point (Do Carmo, 1992: 77). For any $p \in \mathcal{M}$, there is a *convexity radius* $r_p > 0$, such that the open ball

$$B_r(p) = \left\{ \exp_p(q) \mid q \in \mathcal{T}_p\mathcal{M}, ||q||_p < r \right\} \tag{6}$$

is g-convex for any $r < r_p$ (where the norm is induced by the Riemannian metric). Intuitively, the convexity radius quantifies how large a set can be before minimizing geodesics can go "the wrong way around" the manifold. On the product of two Riemannian manifolds, each geodesic is naturally the product of geodesics on its components. Thus, the product of g-convex sets is g-convex in the product manifold.

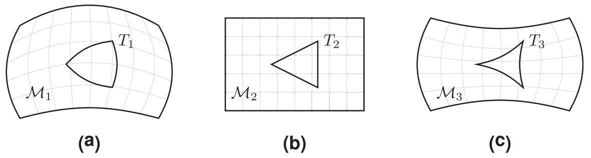

Figure 1. Examples of geodesic triangles T_i in manifolds $\mathcal{M}_i$ with various sectional curvatures. In positive curvature spaces, the interior angles sum to more than 180°, and the edges bow outwards. The opposite is true in negative curvature spaces. (a) Positive. (b) Zero (flat). (c) Negative.

A function $f : \mathcal{M} \rightarrow \mathbb{R}$ is said to be *geodesically convex* (*g-convex*) on $\mathcal{U} \subseteq \mathcal{M}$ if, for any geodesic $\gamma : [0, 1] \rightarrow \mathcal{U}$, $(f \circ \gamma)$ is a convex function on [0, 1]. That is, $\forall t \in [0, 1]$,

$$f(\gamma(t)) \leq (1 - t)f(\gamma(0)) + tf(\gamma(1)). \tag{7}$$

The primary manifolds of interest in this paper are compact, which presents a major obstacle to using g-convexity. If $\mathcal{M}$ is compact and $f : \mathcal{M} \rightarrow \mathbb{R}$ is g-convex on all of $\mathcal{M}$, then f must be constant on each connected component of $\mathcal{M}$ (Boumal, 2022, Cor. 11.10). For example, consider the function which returns the squared geodesic distance from a given point to (0, 1) on the unit circle. This function attains a local maximum at the point (0, −1), so it is clearly not g-convex. However, it is g-convex when restricted to the subset of points on the circle with positive y-value.

We say that f is *locally* g-convex if for any $p \in \mathcal{M}$, there exists a neighborhood $\mathcal{U}_p$ of p such that the restriction of f to $\mathcal{U}$ is g-convex. We make careful use of local g-convexity in our optimization framework: given a locally g-convex objective function f, we construct an atlas such that f is g-convex when restricted to each chart. Importantly, the geodesic distance function $d_{\mathcal{M}^2}$ is locally g-convex if $\mathcal{M}$ is flat.

3.3. Shortest paths in graphs of convex sets

A *Graph of Convex Sets* is a directed graph $G = (V, E)$ together with associated convex sets $\mathcal{X}_v$, $\forall v \in V$, and nonnegative convex functions $\ell_e : \mathcal{X}_u \times \mathcal{X}_v \rightarrow \mathbb{R}$, $\forall e = (u, v) \in E$. Given two vertices $p, q \in V$, a *path* π from p to q is a sequence of distinct vertices $(v_k)_{k=0}^K$ such that $v_0 = p$, $v_K = q$, and $(v_{k-1}, v_k) \in E$, $\forall k \in [K]$. The *shortest path problem* entails finding a path π from p to q, along with continuous variables

$$x_\pi = (x_0, \ldots, x_K) \in \mathcal{X}_\pi := \prod_{k=0}^{K} \mathcal{X}_{v_k}, \tag{8}$$

so as to achieve a minimum *length*, which is defined to be

$$\ell_\pi(x_\pi) = \sum_{k=1}^{K} \ell_{(v_{k-1}, v_k)}(x_{v_{k-1}}, x_{v_k}). \tag{9}$$

If we let Π describe the set of paths from p to q, then the shortest path problem on a graph of convex sets can be concisely stated as

$$\min_{\pi \in \Pi} \min_{x_\pi \in \mathcal{X}_\pi} \ell_\pi(x_\pi). \tag{10}$$

This formulation implicitly admits edge constraints of the form $(x_u, x_v) \in \mathcal{X}_{(u,v)}$, where $\mathcal{X}_{(u,v)}$ is convex, by defining $\ell_{(u,v)}(x_u, x_v) = \infty$ for $(x_u, x_v) \notin \mathcal{X}_{(u,v)}$. Vertex costs are also admissible, as they can be applied as edge costs to all outgoing edges.

Marcucci et al. (2021) presented a mixed-integer nonlinear programming (MINLP) formulation of this problem, and then constructed a tight mixed-integer convex reformulation. The MINLP formulation builds upon the standard formulation of a graph shortest path problem as a network flow problem (Hillier and Lieberman, 2015: 387). Associate to each $e \in E$ a *flow variable* $\lambda_e \in \{0, 1\}$, and let $\mathcal{E}_v^{\text{out}}$ (resp. $\mathcal{E}_v^{\text{in}}$) denote the outgoing (resp. incoming) edges of a vertex $v \in V$. Then the shortest path problem can be written as the network flow problem

$$\begin{aligned} \min \quad & \sum_{e=(u,v)\in\mathcal{E}} \lambda_e \ell_e(x_u, x_v) \\ \text{s.t.} \quad & \sum_{e\in\mathcal{E}_p^{\text{out}}} \lambda_e = \sum_{e\in\mathcal{E}_q^{\text{in}}} \lambda_e = 1, \\ & \sum_{e\in\mathcal{E}_v^{\text{out}}} \lambda_e = \sum_{e\in\mathcal{E}_v^{\text{in}}} \lambda_e, && \forall v \in V\backslash\{p, q\}, \\ & x_v \in \mathcal{X}_v, && \forall v \in V, \\ & \lambda_e \in \{0, 1\}, && \forall e \in E. \end{aligned} \tag{11}$$

When this problem is solved to optimality, the nonzero flow variables will describe a path π, and the corresponding x_π will be optimal for (10).

The product terms $\lambda_e \ell_e(x_u, x_v)$ make the objective function nonconvex. Marcucci et al. (2021) describes a tight mixed-integer convex reformulation of (11), which can be handled by standard mathematical optimization solvers, including Gurobi Optimization LLC (2023) and MOSEK ApS (2019). Furthermore, the convex relaxation of this MICP (obtained by replacing the constraints $\lambda_e \in \{0, 1\}$ with $\lambda_e \in [0, 1]$, $\forall e \in E$) is generally very tight.

Building off of this fact, Marcucci et al. (2023) presented a heuristic strategy for quickly obtaining suboptimal solutions. If the optimal value of each flow variable (from the convex relaxation) is interpreted as the likelihood that the shortest path traverses that edge, one can obtain candidate paths via a randomized depth-first search. For each candidate path π, one solves a small convex program to determine the optimal x_π, and the best such path is returned. This approach generally produces high quality trajectories for motion planning problems, although they may not be globally optimal.

4. Problem statement

We may now precisely state our kinematic planning problem in the language of Riemannian geometry developed thus far. Let $(\mathcal{Q}, g)$ be the configuration space of a robot, realized as a connected Riemannian manifold, and further assume that $\mathcal{Q}$ is *complete* (Lee, 2013: 598) with respect to the metric induced by g. Suppose that the set of collision-free configurations is a bounded open subset $\mathcal{M} \subseteq \mathcal{Q}$, and without loss of generality, assume that $\mathcal{M}$ is path-connected. (If $\mathcal{M}$ is not path-connected, then we restrict ourselves to planning within a single connected component).

Suppose we want to find the shortest path between two points p and q in $\overline{\mathcal{M}}$, the closure of $\mathcal{M}$ (i.e., the smallest closed set containing $\mathcal{M}$). This can be written as the optimization problem

$$\begin{aligned} \operatorname{argmin} \quad & L(\gamma) \\ \text{s.t.} \quad & \gamma \in \mathcal{C}^1_{pw}([0, 1], \overline{\mathcal{M}}), \\ & \gamma(0) = p, \\ & \gamma(1) = q, \end{aligned} \tag{12}$$

where L is the Riemannian arc length, given in eq. (3). In the following sections, we develop machinery to solve optimization problems of this form.

5. Graphs of geodesically convex sets

We now introduce a *graph of geodesically convex sets* (GGCS), a Riemannian optimization framework that, in Section 6, we show is general enough to encompass Problem (12). A GGCS is a directed graph $G = (V, E)$ with certain properties, designed as a generalization of ordinary (Euclidean) GCS from Marcucci et al. (2021, §2) to Riemannian manifolds. The definition of a GGCS closely mirrors that of a GCS, as described in Subsection 3.3. Each vertex $v \in V$ has a corresponding g-convex subset $\mathcal{Y}_v$ of some Riemannian manifold $(\mathcal{M}_v, g_v)$. With each edge $e = (u, v) \in E$, we associate a cost function $\ell_e^{\mathcal{Y}} : \mathcal{Y}_u \times \mathcal{Y}_v \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, which must be g-convex with respect to the product metric on $\mathcal{M}_u \times \mathcal{M}_v$. For all problems considered in this paper, every g-convex set will be a subset of a single Riemannian manifold.

Given distinct source and target vertices p, $q \in V$, a *path* π from p to q is a sequence of distinct vertices $(v_k)_{k=0}^K$ such that $v_0 = p$, $v_K = q$, and $(v_{k-1}, v_k) \in E$ for all $k \in [K]$. Associate to this path a sequence of points $y_\pi = (y_0, \ldots, y_K)$ such that each $y_v \in \mathcal{Y}_v$; then the length of this path is

$$\ell_\pi^{\mathcal{Y}}(y_\pi) = \sum_{k=1}^{K} \ell^{\mathcal{Y}}_{(v_{k-1}, v_k)}(y_{k-1}, y_k). \tag{13}$$

Let Π denote the set of all paths from p to q, and for any $\pi \in \Pi$, define its feasible vertices as $\mathcal{Y}_\pi = \mathcal{Y}_{v_0} \times \cdots \times \mathcal{Y}_{v_K}$. The problem of finding the shortest path from p to q can be written as

$$\min_{\pi\in\Pi} \min_{y_\pi\in\mathcal{Y}_\pi} \ell_\pi^{\mathcal{Y}}(y_\pi). \tag{14}$$

Note that any GCS is an instance of a GGCS, since we can take each $(\mathcal{M}_v, g_v)$ to be an ordinary Euclidean space.

Solving Problem (14) to optimality is intractable in complete generality, so we propose to transform it into an ordinary GCS problem. To each $v \in V$, associate a chart ψ_v, and define $\mathcal{X}_v = \psi_v(\mathcal{Y}_v)$. For each edge $e = (u, v) \in E$, we define the edge cost on $(x_u, x_v) \in \mathcal{X}_u \times \mathcal{X}_v$:

$$\ell_e(x_u, x_v) = \ell_e^{\mathcal{Y}}(\psi_u^{-1}(x_u), \psi_v^{-1}(x_v)). \tag{15}$$

This construction is shown in Figure 2. To apply the GCS machinery, we require that the sets $\mathcal{X}_v$ and edge costs $\ell_e(x_u, x_v)$ are convex. As we show in Subsection 6.5, this is hopeless for manifolds with positive curvature. Luckily, for flat manifolds, convexity can be ensured, as will be shown in Subsection 6.1.

Importantly, many robot configuration spaces can be realized as flat manifolds. SE (2) is flat, all one-dimensional manifolds are flat (Lee, 2018: 47) (this encompasses continuous revolute joints), and products of flat manifolds are flat. Thus, any robotic system whose configuration can be described using a series of single-degree-of-freedom joints (and potentially a mobile base) will have a flat configuration space, and thus can be handled by our methodology. 2-DoF universal joints can also be handled, as they can be perfectly represented as two juxtaposed 1-DoF joints. 3-DoF ball joints cannot be handled perfectly, because decomposing a ball joint into 1-DoF joints distorts the underlying geometry. Instead, one can use a piecewise-linear approximation of this configuration space—see Subsection 6.5 for further discussion. The configuration spaces of certain closed linkages can be explicitly represented as a flat manifold; one constructs a parametrization, and defines the metric on the manifold as the pushforward of the metric in Euclidean space Cohn et al. (2024). Like decomposing a ball joint into 1-DoF revolute joints, this distorts the underlying geometry, so the optimal solutions of the resulting GGCS problem will not be the true shortest paths according to the original metric.

6. Motion planning with GGCS

We want to use GGCS to make motion plans on Riemannian manifolds by solving Problem (12). Thus, we must prove that the optimal value is achieved by some trajectory that is feasible for a GGCS problem. We use the initialism ROSC (Riemannian Open Subset Closure) to describe closures of open subsets of Riemannian manifolds, notably $\overline{\mathcal{M}}$. ROSCs are topological manifolds-with-boundary, but the boundary may not be smooth; for example, polytopic obstacles lead to corners on the boundary of $\overline{\mathcal{M}}$. The theory of manifolds-with-corners is not well developed in full generality, so for the sake of completeness, we confirm some expected properties of paths through ROSCs.

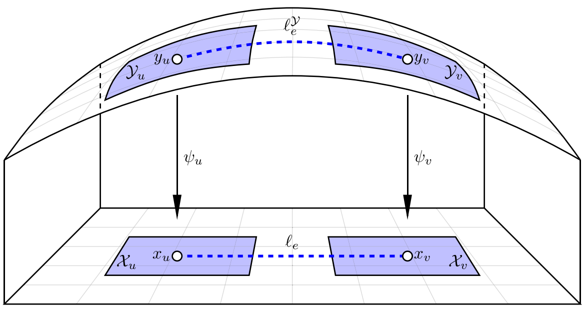

Figure 2. Moving edges and sets from Riemannian manifolds to Euclidean spaces with coordinate charts. In this diagram, $\mathcal{Y}_u$ and $\mathcal{Y}_v$ are visualized as part of the same Riemannian manifold, although this need not be true in general.

Theorem 1. ***(Existence of Optimal Trajectories)*** *For any $p, q \in \overline{\mathcal{M}}$, there exists a continuous curve γ connecting them such that $L(\gamma) = \mathsf{d}(p, q)$.*

Proof. The proof follows by verifying that $\overline{\mathcal{M}}$ satisfies the preconditions of Theorem 2.5.23 of Burago et al. (2001: 50). We defer the details to Appendix A.1.

Assumption 1. *We are given a finite atlas $\mathcal{A} = \{(\mathcal{Z}_v, \psi_v)\}_{v \in V}$ of $\mathcal{M}$, and nonempty open sets $\mathcal{Y}_v \subseteq \mathcal{Z}_v$. For each v, the closure $\overline{\mathcal{Y}}_v$ is g-convex as a subset of $\mathcal{Q}$, and we have $\overline{\mathcal{Y}}_v \subseteq \mathcal{Z}_v$. Additionally, $\bigcup_{v \in V} \overline{\mathcal{Y}}_v = \overline{\mathcal{M}}$.*

These requirements will not hold in general, but we will discuss how to construct such an atlas in Subsection 6.2. With this information, we can prove a strong result about the shortest paths in $\overline{\mathcal{M}}$.

Theorem 2. ***(Piecewise Geodesic Optimal Paths)*** *Let $p, q \in \overline{\mathcal{M}}$, and suppose the sets $\overline{\mathcal{Y}}_v$ satisfy Assumption 1. Then there is a curve $\gamma^* \in C^1_{pw}([a, b], \overline{\mathcal{M}})$ connecting them, such that the following are true:*

- *$L(\gamma^*) = \mathsf{d}(p, q)$.*
- *γ^* is a piecewise geodesic of $\mathcal{Q}$.*
- *Each geodesic segment is contained in some $\overline{\mathcal{Y}}_v$.*
- *γ^* passes through each $\overline{\mathcal{Y}}_v$ at most once.*

Proof. Let γ_0 be a continuous minimizing path connecting p to q (guaranteed to exist by Theorem 1); we will use this to construct an appropriate γ^*. Select an arbitrary order $v_1, \ldots, v_{|V|}$ to iterate over all of the vertices in V. We will construct a sequence of curves $\gamma_1, \ldots, \gamma_{|V|}$, such that $\gamma_{|V|}$ has the desired properties.

For each i, if γ_{i-1} does not pass through $\overline{\mathcal{Y}}_{v_i}$, let $\gamma_i = \gamma_{i-1}$. Otherwise, let $T_i = \left\{ t \mid \gamma_{i-1}(t) \in \overline{\mathcal{Y}}_{v_i} \right\}$, let $a_i' = \min(T_i)$, and let $b_i' = \max(T_i)$. Then by the g-convexity of $\overline{\mathcal{Y}}_{v_i}$, there is a unique minimizing geodesic $\alpha_i : [a_i', b_i'] \to \overline{\mathcal{Y}}_{v_i}$ connecting $\gamma_{i-1}(a_i')$ and $\gamma_{i-1}(b_i')$. Let γ_i be a new curve, defined by

$$\gamma_i(t) = \begin{cases} \gamma_{i-1}(t) & t \notin [a_i', b_i'] \\ \alpha_i(t) & t \in [a_i', b_i'] \end{cases}. \tag{16}$$

Because $L(\alpha_i) \leq L(\gamma_{i-1}|_{[a_i', b_i']})$, we have $L(\gamma_i) \leq L(\gamma_{i-1})$, and since γ_{i-1} is of minimum length, we must have $L(\gamma_i) = L(\gamma_{i-1})$. Continue until we have iterated over all of the $v \in V$. Then by construction, $L(\gamma_{|V|}) = \mathsf{d}(p, q)$, $\gamma_{|V|}$ is piecewise geodesic in $\mathcal{Q}$, each geodesic segment is contained in some $\overline{\mathcal{Y}}_v$, and $\gamma_{|V|}$ passes through each $\overline{\mathcal{Y}}_v$ at most once.

6.1. Formulation as a GCS problem

To transform the GGCS problem into a GCS problem, we require that the sets and edge costs are convex in Euclidean space. The following is sufficient (and still encompasses robots with mobile bases and continuous revolute joints):

Assumption 2. *$\mathcal{Q}$ is flat. Also, each ψ_v is a local isometry into Euclidean space, viewed as a Riemannian manifold with the canonical Euclidean metric.*

Assumptions 1 and 2 together yield three important results:

- $\mathcal{X}_v = \psi_v(\overline{\mathcal{Y}}_v)$ is convex
- $\forall y_0, y_1 \in \overline{\mathcal{Y}}_v$, $\mathsf{d}(y_0, y_1) = ||\psi_v(y_0) - \psi_v(y_1)||_2$
- $\tau_{u,v}$ is a Euclidean isometry (see Lemma 4 in Appendix A.2), and hence affine (Väisälä, 2003).

The first two results are true because $\overline{\mathcal{Y}}_v$ is g-convex, ψ_v maps geodesics to geodesics (Lee, 2018: 125), and geodesics are unique in Euclidean space. For most robotic configuration spaces we consider, $\mathcal{Q}$ can be decomposed as the product of one-dimensional manifolds. In this case, the coordinate systems can be globally aligned, so that every transition map is a translation.

To formulate the problem with GCS, we follow an approach similar to Marcucci et al. (2023), where decision variables describe line segments contained within each convex set. For each set $\overline{\mathcal{Y}}_v$, there is a corresponding vertex $v \in V$, whose convex set in the GCS is $\mathcal{X}_v^2$. $\forall v \in V \setminus \{v_p, v_q\}$, we have $x_v = (x_{v,0}, x_{v,1}) \in \mathcal{X}_v^2$, where $x_{v,0}$ is the start point of the line segment, and $x_{v,1}$ is the end point. The edge set is

$$E = \{(u, v) \mid \overline{\mathcal{Y}}_u \cap \overline{\mathcal{Y}}_v \neq \emptyset\}. \qquad (17)$$

Recall that edges are directed, so if two sets $\overline{\mathcal{Y}}_u$ and $\overline{\mathcal{Y}}_v$ intersect, then $(u, v), (v, u) \in E$. For each edge $e = (u, v) \in E$, we define the edge cost to be the length of the segment associated with the starting vertex,

$$\ell_e(x_u, x_v) = \mathsf{d}(\psi_u^{-1}(x_{u,0}), \psi_u^{-1}(x_{u,1})) = ||x_{u,0} - x_{u,1}||_2. \qquad (18)$$

This will ensure that the optimal solution has minimal arc length among all possible feasible trajectories, as shown in Theorem 3. We also encode equality constraints to ensure the endpoints of adjacent segments are in agreement:

$$\psi_u^{-1}(x_{u,1}) = \psi_v^{-1}(x_{v,0}) \quad \Leftrightarrow \quad \tau_{u,v}(x_{u,1}) = x_{v,0}. \qquad (19)$$

This constraint is convex because $\tau_{u,v}$ is affine.

Let $p, q \in \mathcal{M}$ be the start and goal points, and suppose $p \in \overline{\mathcal{Y}}_{v_0}$ and $q \in \overline{\mathcal{Y}}_{v_1}$. We define a new vertex set $V' = V \cup \{v_p, v_q\}$, where v_p and v_q are associated with the convex set $\{\psi_{v_0}(p)\}$ and $\{\psi_{v_1}(q)\}$, respectively. The corresponding edge set is $E' = E \cup \{(v_p, v_0), (v_1, v_q)\}$. Both (v_p, v_0) and (v_1, v_q) have equality constraint (19). $\ell_{(v_1, v_q)}$ is defined as in (18), and $\ell_{(v_p, v_0)}$ is defined to be zero. Thus, the cost of the final g-convex set traversed by a trajectory is represented by the edge cost from that set to the "goal" vertex. The start and goal positions are fixed by the equality constraints on the new edges.

Altogether, the sets and edges above describe the following GCS problem:

$$\begin{aligned}
\min \quad & \sum_{v \in \pi \setminus \{v_p, v_q\}} ||x_{v,0} - x_{v,1}||_2 \\
\text{s.t.} \quad & \pi = (v_p, v_0, \ldots, v_K, v_q) \in \Pi, \\
& x_{v,0}, x_{v,1} \in \mathcal{X}_v, && \forall v \in V, \\
& \tau_{v_{i-1}, v_i}(x_{v_{i-1},1}) = x_{v_i,0}, && \forall i \in [K], \\
& x_{v_0,0} = \psi_{v_0}^{-1}(p), \\
& x_{v_K,1} = \psi_{v_K}^{-1}(q),
\end{aligned} \qquad (20)$$

where Π denotes the set of all paths in (V', E') from v_p to v_q. (20) closely mirrors the original optimization problem (12) that we sought to solve. The objective function penalizes the length of a path; the remaining constraints enforce continuity, differentiability at all but finitely many points, and the start and goal points.

By using the relaxation strategy of Marcucci et al. (2021), (20) can be solved as a mixed-integer convex program. Alternatively, it can be solved approximately by solving the convex relaxation and using a randomized rounding strategy (Marcucci et al., 2023). After solving the GCS problem, we obtain a path

$$x_\pi = (x_{0,0}, x_{0,1}, x_{1,0}, x_{1,1}, \ldots, x_{K,0}, x_{K,1}) \qquad (21)$$

with $x_{0,0} = \psi_0(p)$, $x_{K,1} = \psi_K(q)$, and $\psi_i(x_{i,1}) = \psi_{i+1}(x_{i+1,0})$, $\forall i \in \{1, \ldots, K-1\}$. Such a path naturally lifts to a path on $\overline{\mathcal{M}}$,

$$\begin{aligned}
y_\pi &= (y_0 = p, y_1, y_2, \ldots, y_K, y_{K+1} = q) \\
&= (\psi_0^{-1}(x_{0,0}), \psi_1^{-1}(x_{1,0}), \ldots, \psi_K^{-1}(x_{K,0}), \psi_K^{-1}(x_{K,1})),
\end{aligned} \qquad (22)$$

where we have removed duplicate points from the trajectory, as $\psi_{i-1}^{-1}(x_{i-1,1}) = \psi_i^{-1}(x_{i,0})$ for each $i \in [K]$. This process is visualized for a simple cylinder manifold $\mathbb{S}^1 \times [0, 1]$ in Figure 3.

For each $i \in \{0, \ldots, K\}$, y_i and y_{i+1} are contained in a g-convex set $\overline{\mathcal{Y}}_i$, so there is a unique minimizing geodesic γ_i connecting them and completely contained in $\overline{\mathcal{Y}}_i$. Thus, a path y_π uniquely defines a piecewise geodesic γ_π connecting p to q that is completely contained in $\overline{\mathcal{M}}$. With this fact, we can formally prove the equivalence of the GGCS problem and the GCS problem.

Theorem 3. ***(Proof of Problem Equivalence)*** *If the path x_π given in eq. (21) is optimal for the GCS problem defined by eqs. (18) and (19), then the piecewise geodesic γ_π defined by eq. (22) is optimal for Problem (12).*

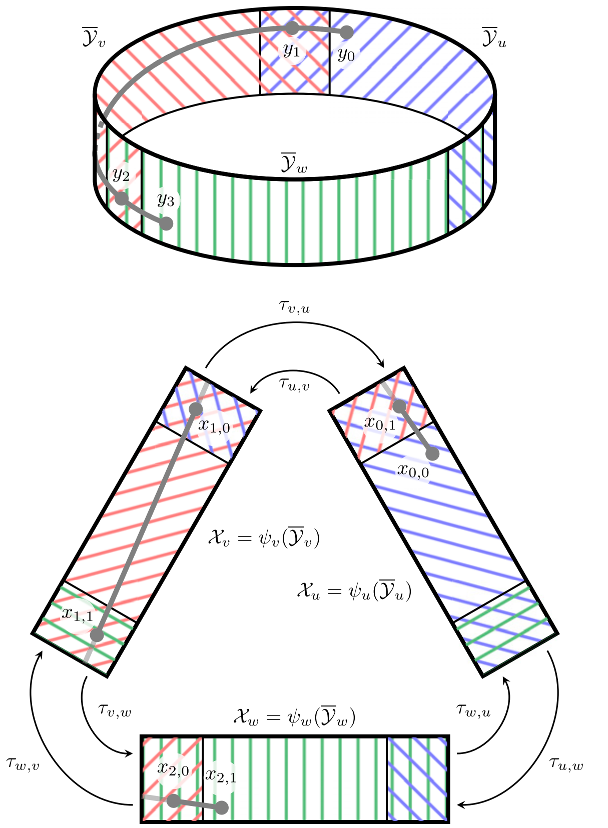
$\overline{\mathcal{Y}}_v$
$\overline{\mathcal{Y}}_u$
y_1
y_0
$\overline{\mathcal{Y}}_w$
y_2
y_3
$\tau_{v,u}$
$\tau_{u,v}$
$x_{1,0}$
$x_{0,1}$
$x_{0,0}$
$\mathcal{X}_v = \psi_v(\overline{\mathcal{Y}}_v)$
$\mathcal{X}_u = \psi_u(\overline{\mathcal{Y}}_u)$
$x_{1,1}$
$\tau_{v,w}$
$\mathcal{X}_w = \psi_w(\overline{\mathcal{Y}}_w)$
$\tau_{w,u}$
$\tau_{w,v}$
$\tau_{u,w}$
$x_{2,0}$
$x_{2,1}$

Figure 3. The process of transforming a GGCS problem into a GCS problem for a simple cylinder manifold. Each of the three charts maps to a Euclidean space, with transition maps encoding the equality constraints across chart domains. The line segments then lift to a piecewise geodesic on the manifold.

Proof. The proof strategy is as follows. First, we show that feasible solutions to the GCS problem have cost greater than or equal to the optimal solution for Problem (12). Then, we show that optimal solutions for Problem (12) are feasible solutions to the GCS problem. Thus, optimal solutions to the GCS problem are optimal solutions to Problem (12).

In particular, any feasible path x_π for the GCS problem yields a piecewise continuously differentiable curve γ_π whose image is contained in $\overline{\mathcal{M}}$ and connecting p to q. Then the length of this curve satisfies

$$L(\gamma_\pi) = \sum_{i=0}^{K} \mathsf{d}(y_i, y_{i+1}) = \sum_{i=0}^{K} ||x_{i,0} - x_{i,1}||_2 = \ell_\pi(x_\pi).$$

Thus, the optimal value of Problem (12) is no worse than the optimal value of the GCS problem.

Now, consider an optimal γ^* for Problem (12), with the properties of Theorem 2. Then γ^* is the concatenation of geodesics $\gamma_1, \ldots, \gamma_K$, where $\gamma_i : [0, 1] \to \overline{\mathcal{Y}}_{v_i}$ for $i = 1, \ldots, K$, and each v_i is distinct. Define x_π by

$$(x_{i,0}, x_{i,1}) = (\psi_i(\gamma_i(0)), \psi_i(\gamma_i(1))),$$

$\forall i \in [K]$. By construction, $\ell_\pi(x_\pi) = L(\gamma^*)$. $\gamma_i(1) = \gamma_{i+1}(0)$, and the v_i are distinct, so x_π is feasible for the GCS problem. Thus, the GCS problem achieves the optimal value of Problem (12).

6.2. Construction of the atlas

A key part of motion planning with GGCS is the construction of an appropriate atlas $\mathcal{A} = \{(\mathcal{Z}_v, \psi_v)\}_{v \in V}$ of $\mathcal{M}$ and sets $\overline{\mathcal{Y}}_v$. Recall that $\mathcal{A}$ must be finite, each $\overline{\mathcal{Y}}_v$ must be g-convex, and each ψ_v must be a local isometry. Here, we present a procedure for constructing such an atlas for a robot with continuous revolute joints (and potentially a mobile base).

As in Marcucci et al. (2023), we construct an inner approximation of C-Free using the extension of IRIS (Amice et al., 2023, Alg. 2) to handle nonconvex obstacles. This algorithm takes in a "seed" point and grows a convex collision-free polytope around it, but it is only applicable to robots with joint limits. Given a seed point in $\mathcal{Q}$, we add artificial joint limits to all continuous revolute joints (and the rotational degree of freedom of the mobile base, if it is present). In this way, we grow the region in Euclidean space, but we have an implicit mapping to the manifold, allowing us to construct the transition maps needed for (19).

To ensure the set is g-convex when lifted to $\mathcal{Q}$, we bound the region by the convexity radius on a per-joint basis. If r_i is the convexity radius of the ith joint's configuration space, we constrain that joint to take values within an *open* ball of radius r_i, centered at the seed point. (Computationally, we use a closed ball of radius $r_i - \epsilon$, with a small $\epsilon > 0$).

Consider the case of the cylinder manifold $\mathbb{S}^1 \times [0, 1]$ shown in Figure 3. Representing $\mathbb{S}^1$ as the unit circle with circumference 2π, the convexity radius along the first dimension is $\pi/2$, and the convexity radius along the second dimension is infinite. Thus, at least three g-convex sets are necessary to cover the manifold. The sets shown in Figure 3 are of the form $[x + \epsilon, x + \pi/3 - \epsilon]$, for $x \in \{0, \pi/3, 2\pi/3\}$ and a small positive constant ϵ.

For revolute joints without limits and the rotational degree of freedom of mobile bases, the convexity radius is $\pi/2$ (as their configuration space is precisely $\mathbb{S}^1$). For a 1-DoF joint, the closed ball that must bound the set is just the interval $[x - r_i + \epsilon, x + r_i - \epsilon]$, where x is the given seed point. If the manifold is flat, this guarantees g-convexity (see proof in Appendix A.2).

Theorem 4. *Suppose $\mathcal{Q} = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_m$, where each $\mathcal{Q}_i$ has a convexity radius r_i. Let $(\mathcal{Z}, \psi)$ be a coordinate chart, $\mathcal{Y} \subseteq \mathcal{Z}$ as in Assumption* 1, *with ψ a local isometry and $\psi(\overline{\mathcal{Y}})$ convex in Euclidean space. If $\mathcal{Q}$ is flat and the diameter of* $\mathrm{proj}_{\mathcal{Q}_i}(\overline{\mathcal{Y}})$ *is less than $2r_i$ for each $i \in [m]$, then $\overline{\mathcal{Y}}$ is g-convex.*

Given two regions $\mathcal{X}_u$ and $\mathcal{X}_v$ generated by this approach, we must determine if they intersect on $\mathcal{Q}$, and if so, what their transition map is. We compute the corresponding axis-aligned bounding boxes $\mathcal{B}_u$ and $\mathcal{B}_v$. Along each dimension i corresponding to a continuous revolute joint, we search for an *offset* o such that $(\mathcal{B}_u + o) \cap \mathcal{B}_v \neq \emptyset$. The ith component of o must be a multiple of 2π if the ith degree of freedom is a continuous revolute joint (or corresponds to the rotational degree of of freedom of the mobile base); otherwise, o_i must be 0.

If such an o exists, we then solve the small convex program

$$\begin{aligned} \text{find} \quad & x \\ \text{s.t.} \quad & x \in \mathcal{X}_u, \\ & x + o \in \mathcal{X}_v. \end{aligned} \tag{23}$$

If such an x exists, then the regions overlap, and their transition map is $x_u \mapsto x_u + o$. If no such offset exists, or if (23) returns infeasible, then the regions do not overlap.

We also assumed full coverage of $\overline{\mathcal{M}}$ by the union of the $\overline{\mathcal{Y}}_v$. In scenarios where we only have an inner approximation of C-Free, we treat all points outside of that approximation as obstacles. Thus, our planner finds the globally optimal path within "certified" C-Free, which is a subset of the whole C-Free.

In general, the computational complexity of the resulting GCS problem grows with both the number of g-convex sets and the density of their overlap graph. So atlases with fewer, larger sets generally lead to faster solve times. Methods used for generating individual convex subsets of C-Free generally attempt to maximize the volume of the resulting sets (Amice et al., 2023; Deits and Tedrake, 2015; Petersen and Tedrake, 2023; Yang and Lavalle, 2004). Werner et al. (2024) attempt to construct an inner covering of an arbitrarily high fraction of C-Free with as few convex sets as possible. Wu et al. (2024) attempt to cover the part of C-Free surrounding a known collision-free trajectory, toward improving that path. In general, building an approximation of C-Free that is ideal for motion planning is an area of active research that is beyond the scope of this paper.

6.3. Additional costs and constraints

Marcucci et al. (2023) extended GcsTrajOpt to parametrize trajectories as piecewise Bézier curves, in order to handle a greater variety of costs and constraints. This includes penalizing the duration and energy of a trajectory, adding velocity bounds, and requiring the trajectory to be differentiable a certain number of times. Bézier curves generalize naturally to Riemannian manifolds by interpolating along the minimizing geodesics between control points (Park and Ravani, 1995; Popiel and Noakes, 2007). Because we restrict ourselves to flat manifolds, the local geometry is unchanged from Euclidean space. Thus, all costs and constraints that operate on individual segments of the piecewise Bézier curve trajectory can be used with no changes.

To enforce the differentiability of the overall trajectory where two segments connect, we must compare tangent vectors across different coordinate systems. In particular, suppose we need differentiability η times for an edge (i, j), with transition map $\tau_{i,j}$. Let γ_i and γ_j be adjoining Bézier curve segments in $\mathcal{Y}_i$ and $\mathcal{Y}_j$, and let their kth derivatives be $v_i^{(k)}$ and $v_j^{(k)}$ at the point where they connect, called w. Using the pushforward of the transition map at w, this constraint can be written as

$$(\tau_{i,j})_{*,\psi_i^{-1}(w)}\left(v_i^{(k)}\right) = v_j^{(k)}, \quad \forall l \in [\eta]. \tag{24}$$

Because the transition map is a Euclidean isometry, its pushforward is a linear transformation described by an orthogonal matrix, and if the coordinate systems are globally aligned (as described in Subsection 6.1), then the pushforward is the identity map. When $\mathcal{Q}$ is flat, the derivative of a Bézier curve is a linear expression of its control points, so eq. (24) is a convex constraint.

6.4. Beyond flat manifolds

The guarantees afforded by GgcsTrajOpt derive from Assumptions 1 and 2. Assumption 1 affects the completeness and optimality of the algorithm, and it may be impossible to construct an appropriate atlas if the boundary of $\overline{\mathcal{M}}$ is not a piecewise-totally-geodesic submanifold. However, we can always construct a finite atlas of g-convex sets to cover all but an arbitrarily small subset of $\overline{\mathcal{M}}$. In practice, we simply work with an inner approximation of $\overline{\mathcal{M}}$, and GgcsTrajOpt will find the shortest path contained within that inner approximation.

Assumption 2 is used to guarantee that the resulting optimization problem is convex; without it, we may have nonconvex costs or constraints, and can make no guarantees of finding an optimal (or even feasible) solution. But many manifolds of interest in robotics do not satisfy these requirements. Certain configuration spaces are inherently not flat manifolds. Examples include SO(3), which is the configuration space of a ball joint and SE(3), which is the configuration space of a free rigid body. Planning problems where general kinematic or dynamic constraints have been imposed can also yield a constrained configuration space as an embedded non-flat submanifold of the full configuration space.

6.4.1. Piecewise-linear approximations. To handle arbitrary manifolds, we turn our attention to piecewise-linear (PL) approximations. In particular, we consider a triangulation of the manifold: a simplicial (or polytopic) mesh of appropriate dimension whose topology matches the manifold. For certain known manifolds, obtaining PL approximations is straightforward; for example, one can tessellate the two-sphere by iteratively subdividing the faces of an icosahedron, and lifting the new vertices to the surface (Dahl et al., 2014, §2.B.1). For an arbitrary manifold which has a known atlas, one can tessellate the image of each chart in Euclidean space, and lift that tessellation to the manifold (Cohn et al., 2022). For implicitly defined manifolds, there is extensive literature on the method of continuation, where a single point known to lie on the manifold is used to generate an explicit PL approximation (Dankowicz and Schilder, 2013; Ghosh, 2012; Henderson, 2002). Continuation in particular has been successfully integrated with sampling-based motion planning algorithms, in order to solve motion planning problems with nonlinear equality constraints (Porta and Jaillet, 2010).

The GCS machinery can be used to plan along a PL approximation; indeed, the original GCS paper directly considers piecewise-affine systems (Marcucci et al., 2021, §9.3). In the context of approximating a smooth manifold, we will treat each simplex as a chart domain, replace the transition map with the mapping between two adjacent simplices, and approximate the arc length on the manifold with the arc length along the PL approximation.

We consider a similar problem setup to Jaillet and Porta (2013). The configuration space $\mathcal{Q}$ is a d-dimensional embedded submanifold of $\mathbb{R}^n$, defined as the zero level set of a smooth function $F : \mathbb{R}^n \rightarrow \mathbb{R}^{n-d}$ whose Jacobian is full rank everywhere. We are given a piecewise-linear approximation of $\mathcal{Q}$ composed of convex polytopes $\mathcal{P}_i \subseteq \mathbb{R}^n$. For each polytope $\mathcal{P}_i$, the orthogonal projection $\psi_i : \mathcal{M} \rightarrow \operatorname{aff}(\mathcal{P}_i)$ (where $\operatorname{aff}(\mathcal{P}_i)$ denotes the affine hull of $\mathcal{P}_i$) forms a coordinate chart when its domain is appropriately restricted, due to the quantitative implicit function theorem (Liverani, 2019). As in Jaillet and Porta (2013), the preimage of each polytope serves as a conservative approximation of the domain of the chart.

Given an atlas of $\mathcal{Q}$, we can produce an atlas of $\overline{\mathcal{M}}$ by taking polytopic subsets of $\mathcal{P}_i$, whose image is a collision-free set. The IRIS algorithm naturally extends to growing polytopic regions through a general nonlinear coordinate chart (Cohn et al., 2024, §3.E). The nonlinear continuation approaches used to generate the atlas of $\mathcal{Q}$ do not enforce g-convexity of the chart domains, but compactness of $\overline{\mathcal{M}}$ will guarantee the convexity radius is finite, so we can simply partition any oversized charts into g-convex pieces. We label these new polytopes $\mathcal{P}_v$, and obtain a finite atlas $\mathcal{A} = \{(\mathcal{Z}_v, \psi_v)\}_{v \in V}$ and sets $\overline{\mathcal{Y}}_v$ satisfying Assumption 1. For convenience of notation, we define the inverse $\Psi_v = \psi_v^{-1}$, restricted to the appropriate domain.

We cannot satisfy Assumption 2, as $\mathcal{Q}$ may not be flat, and the chart mappings ψ_v are not isometries. However, our sets $\mathcal{P}_v$ are convex in Euclidean space, and we can replace each transition map $\tau_{u,v}$ with the affine mapping from one $\mathcal{P}_u$ to its neighbor $\mathcal{P}_v$. Any point x on the boundary between two polytopes $\mathcal{P}_u$ and $\mathcal{P}_v$ will be lifted to two different representatives y_u and y_v on the manifold by their corresponding charts. We ensure paths are well defined as they cross the boundary between two polytopes by requiring that

$$\Psi_u(\mathcal{P}_u \cap \mathcal{P}_v) \subseteq \operatorname{dom} \psi_v.$$

We will leverage this fact in the procedure for lifting a path x_π in the PL approximation to a path y_π on $\overline{\mathcal{M}}$. So if we mirror the transformation procedure described in Subsection 6.1, we obtain a valid GCS problem.

Solving the shortest path problem on the PL approximation yields a path x_π as in (21). However, we generally do not have $\Psi_u(x_{u,1}) = \Psi_v(x_{v,0})$, so the process for lifting to a path y_π on $\overline{\mathcal{M}}$ is more complex. For each $i \in [K] \cup \{0\}$, we define $y_{i,0} = \Psi_i(x_{i,0})$ and $y_{i,1} = \Psi_i(x_{i,1})$. We begin at $y_{0,0}$ and trace the minimizing geodesic from $y_{0,0}$ toward $y_{0,1}$. At the transition from chart $i - 1$ to i, if $y_{i-1,1} \in \Psi_i(\mathcal{P}_i)$, and if

$$||\psi_i(y_{i-1,1}) - x_{i,1}||_2 \leq ||x_{i,0} - x_{i,1}||_2,$$

then we trace the minimizing geodesic from $y_{i-1,1}$ to $y_{i,1}$. Otherwise, we simply trace the minimizing geodesic from $y_{i-1,1}$ to $y_{i,0}$ and then to $y_{i,1}$.

Overall, solving the shortest path problem gives us a path $(x_{0,0}, x_{0,1}, \ldots, x_{K,0}, x_{K,1})$, where $x_{i-1,1} = x_{i,0}$ for each $i \in [K]$. We lift it to a sequence of waypoints $(y_{0,0}, y_{0,1}, \ldots, y_{K,0}, y_{K,1})$, where we may have $y_{i-1,1} \neq y_{i,0}$ for some $i \in [K]$. Finally, for each $i \in [K]$, if it would be shorter to skip $y_{i,0}$ and go straight from $y_{i-1,1}$ to $y_{i,1}$, we drop $y_{i,0}$ from the path. Such a "skip" occurs when paths along the PL approximation are longer than those along the manifold. As an example, consider a circle manifold, inscribed within a regular hexagon (which serves as the PL approximation). If a path along the hexagon were projected onto the circle, then traversing from $y_{i-1,1}$ to $y_{i,0}$ would require doubling back—the path is shorter if one goes straight from $y_{i-1,1}$ to $y_{i,1}$.

We now endeavor to prove this GCS problem provides a useful approximation of Problem (12). We introduce the following notation to describe the length of a path along the PL approximation:

$$L_{\mathrm{PL}}(x_\pi) = \sum_{i=0}^{K} ||x_{i,0} - x_{i,1}||_2. \tag{25}$$

The PL approximation then becomes a metric space with distance function d_{PL} by taking the infimum of the arc lengths, analogous to (4). Let $p \in \mathcal{P}_p$ and $q \in \mathcal{P}_q$. Solving our approximation to global optimality yields a path x_π^* whose length $L_{\mathrm{PL}}(x_\pi^*)$ is minimal among all paths x_π from p to q. In practice, we use the relax-and-round strategy, so we do not assume that $L_{\mathrm{PL}}(x_\pi^*) = \mathsf{d}_{\mathrm{PL}}(p, q)$.

We will provide an upper bound on $\mathsf{d}(\Psi_p(p), \Psi_q(q))$ in terms of $L_{\mathrm{PL}}(x_\pi^*)$, K (x_π^* traverses $K+1$ sets per (21)), and the following numerical quantities derived from the PL approximation:

- ϵ_H: The maximum chart Hausdorff distance $\max_i \mathsf{d}_H(\mathcal{P}_i, \Psi_i(\mathcal{P}_i))$.
- $\alpha_{\max}$: The largest principal angle (Knyazev and Zhu, 2012) between any pair of adjacent charts.

Hausdorff distance is a common metric to quantify the difference between two approximations of a geometric object (Cignoni et al., 1998). The largest principal angle between charts, together with the radius of the smallest chart, presents a coarse bound on the curvature of the manifold. Taking a finer discretization generally leads to both quantities decreasing; for example, see Figure 4. Furthermore, compactness of the manifold ensures that the curvature is bounded, so ϵ_H and $\alpha_{\max}$ can be made arbitrarily small with finitely many charts.

We leverage two lemmata in deriving a global upper bound on the path length (see proofs in Appendix A.3).

Lemma 1. *Fix a chart* $(\mathcal{Z}_i, \psi_i)$ *and points* $x_0, x_1 \in \psi_i(\overline{\mathcal{Y}}_i)$. *Let* $y_0 = \Psi_i\,(x_0)$ *and* $y_1 = \Psi_i\,(x_1)$. *Then*

$$\mathsf{d}(y_0, y_1) \le ||x_0 - x_1||_2 \left(1 + \max_{x \in \psi_i(\overline{\mathcal{Y}}_i)} ||D\Psi_i(x) - I||_{\text{op}}\right).$$

This demonstrates that the error within a chart is upper bounded by how much the total derivative of the chart mapping deviates from the identity mapping. When constructing charts as in Jaillet and Porta (2013), the polytope is constructed so as to lie along the tangent space of some point on the manifold. Thus, there is always some $x \in \psi_i(\overline{\mathcal{Y}}_i)$ such that $D\Psi(x) = I$. By smoothness, this entails that $\max_{x \in \psi(\overline{\mathcal{Y}})} ||D\Psi(x) - I||_{\text{op}}$ can be made arbitrary small by using a finer atlas $(\mathcal{Z}_v, \psi_v)$ and sets $\overline{\mathcal{Y}}_v$ whose images have smaller diameter.

Still, an upper bound on the whole path across multiple charts can also incur error along the chart transitions. This is because we have replaced the transition maps with approximations, so the portions of the path on the manifold that cross the boundary between charts may be collapsed to zero length. The following lemma quantifies this error by projecting that portion of the path onto one of the charts.

Lemma 2. *For subsequent charts i, $i + 1$ traversed by* x_π,

$$\mathsf{d}(y_{i,1}, y_{i+1,0}) \le \epsilon_H \sin \alpha_{\max} \left(1 + \max_{x \in \mathcal{P}_i} ||D\Psi_i(x) - I||_{\text{op}}\right).$$

We now have an upper bound on the error within a chart and across subsequent charts. Thus, by solving the PL-approximation, we obtain an optimal path length $L_{\text{PL}}(x_\pi^*)$, and use this to compute an upper bound on the true distance along the manifold $\mathsf{d}(\Psi_p(p), \Psi_q(q))$.

Theorem 5. (*Approximation Upper Bound*)

$$\mathsf{d}(\Psi_p(p), \Psi_q(q)) \le \left(L_{\text{PL}}(x_\pi^*) + K\epsilon_H \sin \alpha_{\max}\right) \left(1 + \max_{\substack{v \in V \\ x \in \mathcal{P}_v}} ||D\Psi_v(x) - I||_{\text{op}}\right), \tag{26}$$

where $K + 1$ is the number of charts traversed by x_π^*.

Proof. Let x_π^* be the optimal path. Then we leverage the triangle inequality for the manifold distance metric, along with Lemmata 1 and 2, to obtain the desired bound:

$$\begin{aligned}
\mathsf{d}(p, q) &\le \sum_{i=0}^{K} \mathsf{d}(y_{i,0}^*, y_{i,1}^*) + \sum_{i=0}^{K-1} \mathsf{d}(y_{i,1}^*, y_{i+1,0}^*) \\
&\le \sum_{i=0}^{K} ||x_{i,0}^* - x_{i,1}^*||_2 \left(1 + \max_{x \in \mathcal{P}_i} ||D\Psi_i(x) - I||_{\text{op}}\right) \\
&\quad + \sum_{i=0}^{K-1} \epsilon_H \sin \alpha_{\max} \left(1 + \max_{x \in \mathcal{P}_i} ||D\Psi_i(x) - I||_{\text{op}}\right) \\
&\le \left(L_{\text{PL}}(x_\pi^*) + K\epsilon_H \sin \alpha_{\max}\right) \\
&\quad \left(1 + \max_{\substack{v \in V \\ x \in \mathcal{P}_v}} ||D\Psi_i(x) - I||_{\text{op}}\right).
\end{aligned}$$

□

Roughly speaking, $\epsilon_H \sin \alpha_{\max}$ is a linear approximation of the portion of the path not accounted for within each polytope. If this quantity can be bounded above a multiplicative factor of the portion of the path in each polytope, we can obtain a multiplicative upper bound on $\mathsf{d}(\Psi_p(p), \Psi_q(q))$.

Corollary 1. *If* $\exists c > 0$ *such that* $\epsilon_H \sin \alpha_{\max} \le c \cdot \mathsf{d}(y_{i,0}, y_{i,1})$ *for all i, then*

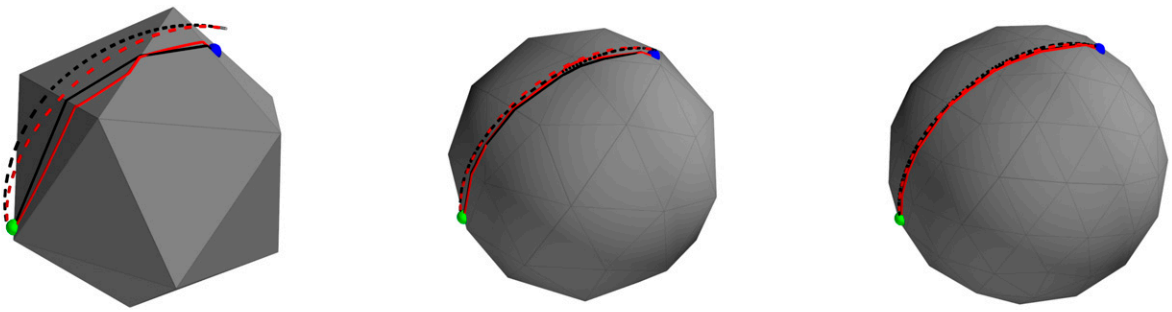

Figure 4. A shortest path problem on the unit sphere solved with GgcsTrajopt using an increasingly fine discretization. The blue and green markers denote the start and end configurations, respectively. The solid black path denotes the shortest path along the mesh (found with GgcsTrajOpt), and the dotted black path is the result obtained when that path is lifted back to the sphere. Conversely, the red dotted path is the ground truth shortest path along the sphere, and the solid red path is its projection onto the PL approximation.

$$\mathsf{d}(\Psi_p(p), \Psi_q(q)) \leq L_{\mathrm{PL}}(x_\pi^*)(1+c) \left(1 + \max_{\substack{v \in V \\ x \in \mathcal{P}_v}} \|D\Psi_v(x) - I\|_{\mathrm{op}}\right). \quad (27)$$

6.5. Planning over SO(3)

The Lie group SO(3) is of great interest in robotics, and thus merits special consideration. For example, the configuration space of a ball joint is a subset of SO(3), and the configuration space of a free moving object in $\mathbb{R}^3$ is $\mathrm{SE}(3) \cong \mathrm{SO}(3) \times \mathbb{R}^3$. Under the standard Riemannian metric, the manifold has positive curvature (a direct result of Corollary 3.19 of Cheeger et al. (1975)), and unfortunately, even a single point of positive curvature implies that the Riemannian distance function is not g-convex, even on arbitrarily small neighborhoods of that point (see proof in Appendix A.4).

Theorem 6. *Let $\mathcal{M}$ be a Riemannian manifold, let $A_1 \in \mathcal{M}$ and $u, v \in T_{A_1}\mathcal{M}$ such that $\mathcal{K}(u, v) > 0$. Then for any neighborhood $\mathcal{U}$ containing A_1, $\mathsf{d} : \mathcal{M}^2 \to \mathbb{R}$ is nonconvex on $\mathcal{U}^2$.*

One way to get around this problem is to instead use Euler angles, where a rotation is represented as the product of three rotations, about the x, y, and z axes in a given coordinate system. As a manifold, the set of Euler angles is $\mathbb{T}^3$, and it has a natural flat Riemannian metric. This makes the Euler angles parametrization very easy to plan over with GgcsTrajOpt. However, this parametrization has degeneracies due to gimbal lock, and the flat metric distorts the true distance.

Another option is to use the approximation strategy described in Subsection 6.4 that can handle arbitrary manifolds, which naturally encompasses SO(3). One can describe SO(3) as the subset of $\mathbb{R}^{3\times3}$ satisfying six nonlinear constraints, but it is more common to work with more compact descriptions. One such description is the unit quaternions, which is geometrically equivalent to the three-sphere $\mathbb{S}^3$. The unit quaternions form a double cover of SO(3), as the unit quaternions v and $-v$ correspond to the same rotation. A naive way to handle this is to simply solve the shortest path problem to the goal and its antipode, and take the shorter path. Another representation of SO(3) is the axis-angle convention, $\mathbb{S}^2 \times \mathbb{S}^1$. Each nonzero rotation corresponds to two axis-angle representations (related by having axis vectors pointing in opposite directions, and opposite angles), except for the identity rotation, which can be represented by any axis and the zero angle. Although the axis-angle convention is less elegant topologically, $\mathbb{S}^1$ is flat, so the approximation is only needed for the manifold $\mathbb{S}^2$.

We empirically compare both of these approximations to the Euler angles parametrization in Subsection 7.5. Based on these comparisons, we recommend using the axis-angle parametrization if optimality is a concern, since it can provide a finer approximation than quaternions at a reduced computational cost. If algorithmic runtime is the primary concern, the Euler angles parametrization will lead to the fastest runtimes, but this comes at the cost of suboptimal solutions.

7. Experiments

We demonstrate our GgcsTrajOpt on various robotic platforms. We present illustrative toy examples of planning for a point robot on a toroidal world and an arm in the plane with multiple continuous revolute joints. We also build plans for a KUKA iiwa arm (with the base joint modified to be continuous revolute) and a PR2 bimanual mobile manipulator, implemented in Drake (Tedrake and the Drake Development Team, 2019). We make interactive recordings of these trajectories available online at our results website. For each experiment, we explicitly state the configuration space, using $\mathcal{I}$ to refer to a general bounded interval in $\mathbb{R}$. Each of the robot experiments in Subsections 7.1 to 7.4 have flat configuration spaces, and we use the canonical metric which equally weights velocities for each degree of freedom. Besides the robot experiments, we compute shortest paths along *SO* (3) with each of the three approximation strategies discussed in Subsection 6.5. Finally, we compute shortest paths for a block, with configuration space SE (3), moving through a maze.

7.1. Point robot

Consider a point robot moving about a toroidal world (configuration space $\mathbb{T}^2$, modeled as a unit square with the edges identified), with convex planar obstacles. It is easy to visualize the obstacles, g-convex sets, and graph edges, as shown in Figure 5. We also show an example of an optimal trajectory produced by GgcsTrajOpt, which “wraps around” the toroidal world. This plan was computed in 0.79 s.

7.2. Planar arm

Consider a robot arm with a fixed base, composed of five continuous revolute joints (configuration space $\mathbb{T}^5$), moving through a planar workspace with convex obstacles. (We assume the arm does not suffer from self-collisions.) We present sample plans produced by GgcsTrajOpt in Figure 6, together with the swept collision volumes. These two plans were found in 5.36 and 4.63 s, respectively. A video of these trajectories is available at our results website.

7.3. Modified KUKA iiwa arm

We also demonstrate that GgcsTrajOpt can be used to plan a series of motions using a KUKA iiwa robot arm. The KUKA iiwa is a 7-DoF robot arm where each joint is a revolute joint with limits; in simulation, we remove the limits on the first joint, so the configuration space is $\mathbb{T}^1 \times \mathcal{I}^6$.

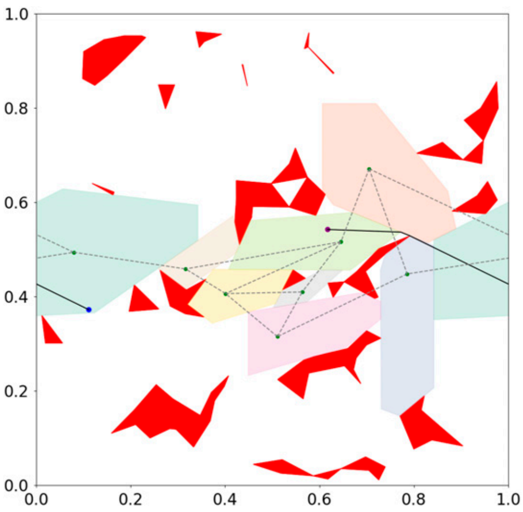

Figure 5. Results for a point robot in a toroidal world, realized as a unit square with opposite edges identified. Obstacles are shown in red, and each IRIS region is given a distinct pastel color. Note that one of the regions "wraps around" along the horizontal dimension, connecting opposite sides of the world. Grey dashed lines indicate which regions overlap. The optimal path between the start and end points is shown in black.

We consider a scenario where the arm is mounted on a table, surrounded by three sets of shelves, with mugs arranged on the shelves. The goal of the task is to sort the mugs onto different shelves, organized by color. We specify the order of motions that are needed to achieve this goal and use GgcsTrajOpt to find the path from start to goal for each motion.

For this experiment, we used a set of 18 convex regions to achieve approximate coverage of the collision-free space. These regions were adjusted as the mugs were moved about the environment and were used to plan the complete motion of the arm—no heuristic motion or "pre-grasp pose" was needed to reach the grasp configuration. Several configurations used to seed the region generation are shown in Figure 7, and the initial and final states are shown in Figure 8. A video and an interactive recording of the plan are available at our results website.

The experiment consisted of 14 motions, which were each planned individually, and we use the region refinement method from Petersen and Tedrake (2023) to account for the current placement of the mugs. This ensured that both the arm *and the grasped mug* were collision-free for the entirety of each trajectory segment. The robot takes full advantage of the base joint's lack of limits—always choosing the shortest path and never needing to unwind any rotations. For each segment, planning a trajectory took an average of 25.75 s (with a range of 4.63 to 50.30 s).

7.4. PR2 bimanual mobile manipulator

In addition to its mobile base, the PR2 has two continuous revolute joints in each arm. We have fixed the wrist rotation and gripper joints, so the configuration space is $\mathrm{SE}(2) \times \mathbb{T}^2 \times \mathcal{I}^{10} \cong \mathbb{T}^3 \times \mathcal{I}^{12}$. We consider a scenario where the robot is driving around a square table that has an outward-pointing stack of three shelves on each side. The robot must reach into the different shelves with both arms. This represents a challenging motion planning scenario for all existing approaches due to the obstacle-rich environment and high dimensionality of the configuration space.

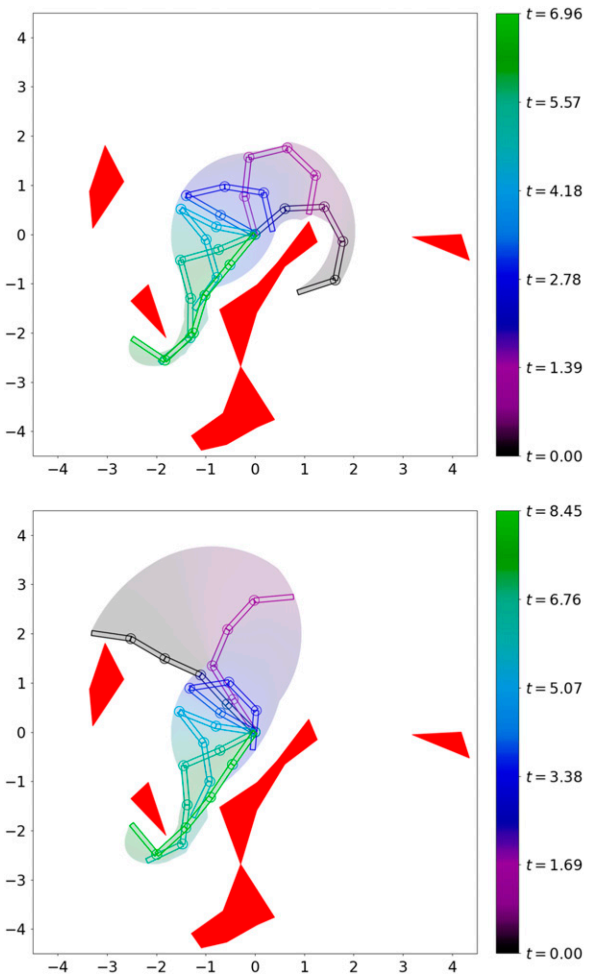

Figure 6. Two plans produced by GgcsTrajOpt for a planar arm around task-space obstacles (shown in red). We display both the swept collision volume and individual poses in the trajectory (colored by time, as indicated by the color bar).

The performance of GgcsTrajOpt is largely driven by the choice of g-convex sets. For each set of shelves, we generate IRIS regions for the robot to reach into the top, middle, and bottom shelves with both arms simultaneously. We also generate two additional regions where the robot reaches into the middle shelf with one arm and the bottom shelf with the other while crossing its arms. Finally, we manually add various intermediate regions to promote graph connectivity and cover more of C-Free. In Figure 9, we show several robot configurations along a trajectory produced by GgcsTrajOpt.

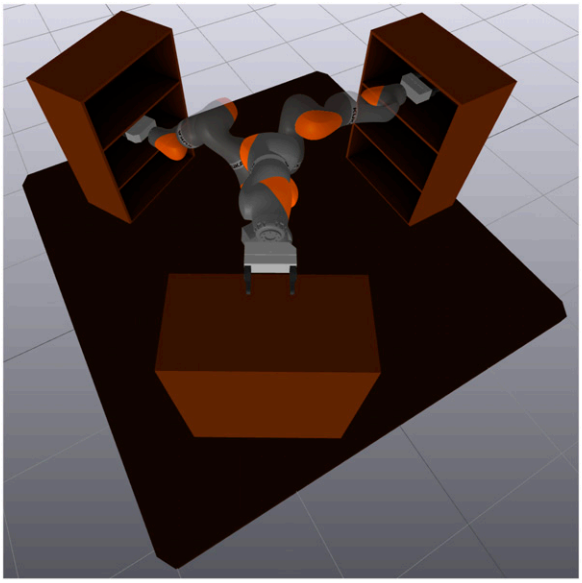

Figure 7. Key configurations (overlaid) used for a mug reorganization demo.

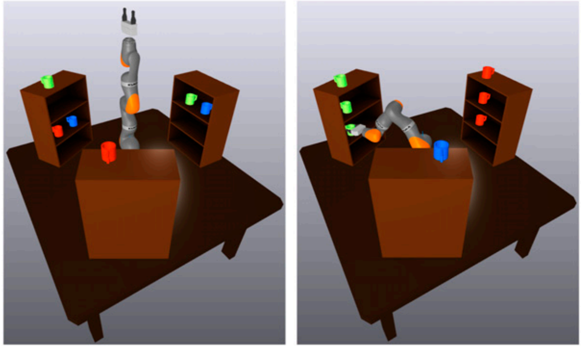

Figure 8. Initial (left) and final (right) states for the mug reorganization demo.

For the planning scenarios considered with the PR2, we compare GgcsTrajOpt to existing approaches. Trajectory lengths are listed in Table 1, plan times are listed in Table 2, and interactive recordings of all trajectories are available online at our results website. We compare our algorithm to kinematic trajectory optimization (Tedrake, 2022, §7.2) (abbreviated as *Drake-TrajOpt*), utilizing the general nonlinear optimizer SNOPT (Gill et al., 2005, 2018). Drake-TrajOpt struggles to figure out how to move the arms into or out of the shelf; we often have to add waypoints to force the robot to back out of the shelf by moving its base.

We also compare it to a sampling-based PRM planner. To mitigate the curse of dimensionality and ensure connectivity between seed points, we initialize the PRM with the skeleton of the GGCS graph: for each pair of overlapping regions, we place a node in the Chebyshev center (Boyd et al., 2004: 148) of their intersection. We then add 100,000 additional samples, drawn uniformly across C-Free (with rejection sampling). This process takes 124.39 s. In comparison, it takes an average of 30.20 s to generate an IRIS region (with a range of 8.56 to 75.42 s). With parallelization, all of the IRIS regions were generated in only 156.63 s.

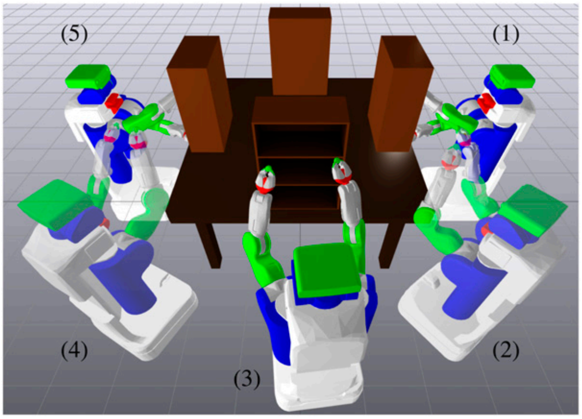

Figure 9. Individual poses along a trajectory produced by GgcsTrajOpt for the PR2 robot, labeled with their order in the plan.

The plans produced by the PRM are significantly longer than those from the GgcsTrajOpt, so we also examine using the output of the PRM planner as the initial guess for the trajectory optimization. (In principle, this should help prevent Drake-TrajOpt from getting stuck in local minima.) When post-processing PRM plans with Drake-TrajOpt, it sometimes produces shorter final trajectories than GgcsTrajOpt, at the expense of colliding slightly with the environment (an example is shown in Figure 10). This is likely due to the challenge of balancing the collision-free constraint with the minimum distance objective (and because collision-free constraints can only be applied at discrete points).

Finally, we compare our GgcsTrajOpt to two workarounds for applying ordinary GCS to non-Euclidean motion planning. One could add artificial joint limits to prevent the wraparound, but placing the joint limits incorrectly could make the optimal path infeasible. The planar arm experiment clearly demonstrates this problem; during the second trajectory in Figure 6, the middle joint traverses more than 360° in the course of the plan. Thus, the optimal trajectory is infeasible for *every* possible choice of joint limits.

Another option is treating the angles as real numbers with no bound (and ignoring the fact that $0° \equiv 360°$). But in this case, the correct joint angle modulo 360° must be chosen to get the optimal path. Furthermore, many copies of each convex set must be made to account for each possible choice of angle modulo 360°, increasing the size of the optimization problem.

With both workarounds, a priori knowledge about the solution is required to guarantee that it is found, so in each comparison, we separately consider the best and worst cases. We use the same seed points across GgcsTrajOpt and both GcsTrajOpt workarounds.

Table 1. A comparison of trajectory lengths (in configuration space) for each PR2 experiment across different methods. Experiments are titled by the start and goal configurations. The configuration names indicate the shelf positions on the table (1 through 4), followed by the position of the grippers. T, M, B, CL, and CR stand for top, middle, bottom, cross left over right, and cross right over left (respectively). Table cells that are struck through indicate that the trajectory is not collision-free, and the time stamps below the trajectory length indicate when the collisions occurred. The shortest collision-free trajectory for each experiment is bolded. For both GcsTrajOpt workarounds, we include the best- and worst-case results (in general, achieving the best-case results requires a priori knowledge of the optimal plan). Interactive recordings of each trajectory are available online at our results website, https://ggcs-anonymous-submission.github.io/.

Experiment	GgcsTrajOpt	Drake-TrajOpt	PRM	PRM + Drake-TrajOpt	GcsTrajOpt (joint limits)	GcsTrajOpt (no joint limits)
1T to 1B	1.829	**1.803**	4.359	1.808	1.826	1.839
1CL to 1CR	2.255	2.204	9.219	**2.182**	2.239	2.247
1M to 4M	**3.875**	~~6.938~~ $t = 0.275$, 5.272	14.554	~~5.874~~ $t = 0.714$, 4.381	6.482 / 10.478	3.990 / 12.589
1CL to 2CR	4.473	~~5.409~~ $t = 2.155$	12.110	~~4.108~~ $t = 0.49$	**4.441** / 13.815	4.640 / 13.233
1CL to 3CR	8.182	10.263	15.250	~~7.166~~ $t = 0.7$, 1.87, 2.02,2.77	**7.820** / 12.125	8.501 / 12.125
1CL to 4CR	**4.382**	7.583	17.459	~~6.088~~ $t = 0.27$, 0.555,4.39	4.728 / 9.961	4.559 / 12.418
1T to 4B	**4.538**	8.781	12.351	~~5.949~~ $t = 0.34$, 0.68	5.320 / 14.928	5.473 / 14.160

Table 2. A comparison of online planning times (in seconds) for each PR2 experiment across different methods. (We omit the GCS workaround comparisons, as they are indistinguishable from the corresponding GgcsTrajOpt runtimes.) Experiment names match Table 1.

Experiment	GgcsTrajOpt	Drake-TrajOpt	PRM	PRM + Drake-TrajOpt
1T to 1B	25.51	12.69	0.49	11.61
1CL to 1CR	39.42	15.23	0.49	16.11
1M to 4M	46.61	2.26	0.53	25.51
1CL to 2CR	62.87	9.74	0.54	21.48
1CL to 3CR	58.60	7.82	0.52	27.30
1CL to 4CR	66.15	4.32	0.54	40.10
1T to 4B	29.89	10.92	0.54	15.36

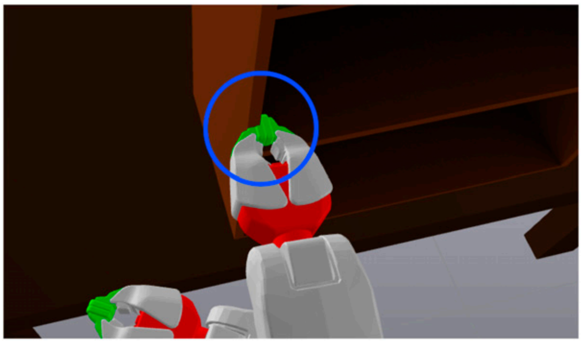

Figure 10. An example of the slight collisions typical of the trajectories produced by Drake-TrajOpt. (The blue circle highlights the point of collision).

An interesting result is that the best case for the GcsTrajOpt workarounds is sometimes slightly better than GgcsTrajOpt. This is because the sets are not bounded by the convexity radius, so they can grow larger (and cover more of C-Free) with the same seed points. If the workarounds are restricted to using the same regions as GgcsTrajOpt, then, in the best case, their performance is indistinguishable.

7.5. Planning over SO(3)

As discussed in Subsection 6.5, it is necessary to use an approximation strategy to plan over SO(3) with our methodology. To compare the efficacy of the approximation strategies, we consider an abstract planning problem, where we have to find the shortest path (with respect to the canonical bi-invariant Riemannian metric) between two configurations in SO(3). Since there are no obstacles, we can compare the solution from each approximation to the closed-form solution, obtained from spherical linear interpolation (SLERP) (Dam et al., 1998). For numerical purposes, we slightly expand the charts throughout these experiments, so as to achieve a positive-measure overlap.

The Euler angles description is equivalent to $\mathbb{T}^3$, which we realize as a cube with opposing faces identified. We use a flat metric on $\mathbb{T}^3$, where the velocity along each component is weighted equally. We use sets of the form $[x, x + 2\pi/3] \times [y, y + 2\pi/3] \times [z, z + 2\pi/3]$ for $x, y, z \in \{0, 2\pi/3, 4\pi/3\}$. In our experiments, we have a GGCS with 27 sets and 702 directed edges, as the graph is fully connected.

The axis-angle description is equivalent to $\mathbb{S}^2 \times \mathbb{S}^1$, where the first, two-sphere-valued component denotes the axis of rotation and the second, circle-valued component denotes the angle rotated about that axis. We use a product metric on $\mathbb{S}^2 \times \mathbb{S}^1$, where we evenly weight the canonical flat metric on $\mathbb{S}^1$ and the metric induced on $\mathbb{S}^2$ as a submanifold of $\mathbb{R}^3$ with the standard Euclidean metric. We use an icosahedron as our PL approximation of the unit sphere, although higher resolution approximations can easily be constructed by subdividing the faces (Dahl et al., 2014, §2.B.1). Sets are of the form $\mathcal{P}_v \times [\theta, \theta + 2\pi/3]$ for a face $\mathcal{P}_v$ of the PL approximation and $\theta \in \{0, 2\pi/3, 4\pi/3\}$. In our experiments, we have a GGCS with 60 charts and 660 directed edges. Because the axis-angle representation almost perfectly double-covers SO(3) (outside of the identity configuration), we solve two planning problems, to ensure we plan to the closest representative of the orientation.

The quaternion description is equivalent to $\mathbb{S}^3$, and we use the metric induced as a submanifold of $\mathbb{R}^4$ with the standard Euclidean metric. We construct a tiling with respect to the hyperspherical coordinate system (Blumenson, 1960). We evenly tile the angular variables $\psi_1, \psi_2, \psi_3 \in [0, 2\pi]$ to a desired resolution, and then map the corners onto $\mathbb{S}^3$ (by taking the radius of each point to be 1). The polytopes $\mathcal{P}_v$ are then taken to be the convex hulls of the corners of each tile. If we subdivide each angular dimension into three pieces, we have 27 charts and 390 directed edges. Subdividing into four pieces yields 64 charts and 2240 directed edges. Similarly to axis-angle representation, $\mathbb{S}^3$ double-covers SO(3), so we solve two planning problems, to ensure we plan to the closest representative of the orientation.

To compare these approximations, we uniformly sampled random start and goal orientations, and computed the shortest path between them for each approximation strategy. We measure their length according to the geodesic distance between each successive control point of the path, and compare to the ground truth distance between the start and goal. Ground truth for this distance metric can be obtained in closed form with spherical linear interpolation (Dam et al., 1998), allowing us to measure the approximation error for each method. In Figure 11, we plot the distribution of the relative errors for each of the three methods. The mean relative errors for each representation are listed in Table 3.

The distance in the Euler angles representation is known to distort the true distance between orientations, so it is unsurprising that this choice of approximation has higher error. If algorithmic runtime is the primary concern, the Euler angles parametrization will lead to the fastest runtimes, but this comes at the cost of suboptimal solutions. When the higher-resolution approximation of the quaternionic sphere is used, its relative error is roughly equivalent to that of the axis-angle approximation. However, this requires a much larger graph (and many more edges), yielding a more computationally costly optimization program. Thus, we recommend using the axis-angle parametrization, as it strikes the best balance between accuracy and computational efficiency.

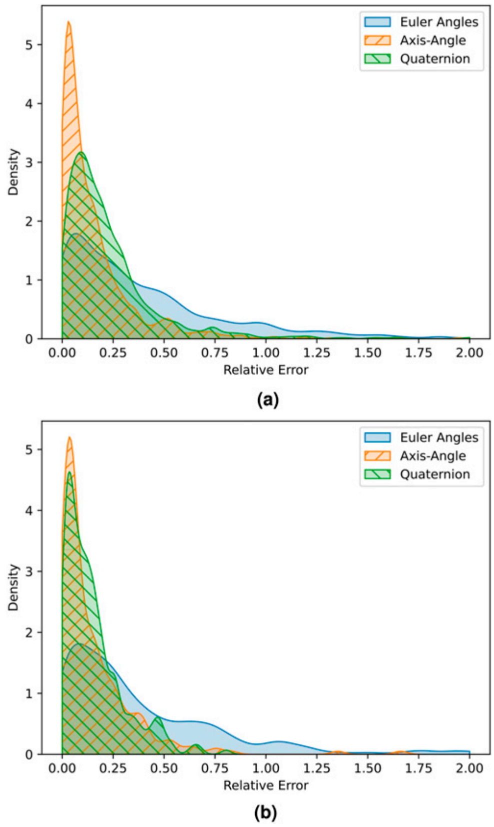

Figure 11. The distribution of relative error across many sampled start and goal configurations when planning using various *SO* (3) approximations strategies. (a) Compared across 1000 random samples, using the lower-resolution approximation of quaternionic sphere. (b) Compared across 250 random samples, using the higher-resolution approximation of quaternionic sphere.

Table 3. A comparison of the mean relative error for each of the four choices of *SO* (3) representations used with GGCS. All statistics are taken from the experiment with 1000 random samples, except for the high resolution discretization of the quaternion description, which used 250 samples.

Parametrization	Mean relative error (%)
Euler angles	41.2
Axis-angle	15.4
Quaternion (low resolution)	22.9
Quaternion (high resolution)	15.2

7.6. Planning over SE(3)

In this experiment, we use GgcsTrajOpt to produce plans for a block moving freely throughout a maze. The maze has a

2D layout, similar to the maze shown in Figure 2 of (Marcucci et al., 2023), but the walls of the maze enclose a 3D world. The block is allowed to move freely through the maze—its configuration space is SE(3). However, it is not allowed to leave the maze along the vertical direction; the center of the block is constrained to lie between the top and bottom of the maze walls. We produce plans between nine different key poses of the block, shown within the maze in Figure 12. Above each of these poses, the orientation of the block is annotated with a coordinate frame. These various orientations differ by rotations of up to 180° along all three axes. Thus the planner must carefully reason about which way the block should rotate around any given axis over the course of the trajectory.

We use Euler angles in our representation of SE(3) because we do not yet know how to get collision free regions with PL approximations. We treat the three Euler angles as if they are three successive continuous revolute joints. So we allow the regions to grow up to $\pi - \epsilon$ in diameter along each revolute axis. We construct a coarse grid roadmap of the maze and grow regions to cover a subset of the edges so as to connect the key poses. This results in a GGCS with 200 sets and 4744 (directed) edges. The polytopes were described by an average of 26.095 halfspaces (minimum 21, maximum 35) and took 65.434 s to compute (without leveraging any parallelization).

Given this GGCS, we solve the shortest path problem for every choice of start and goal. Due to the large size of this problem instance, the memory and runtime requirements to solve the convex relaxation are prohibitive. As a mitigation strategy, we replace the 2-norm in the objective of (20) with the 1-norm when solving the convex relaxation. This yields a linear program, which can be solved much more efficiently than the second-order cone program that would result from the 2-norm objective. We observe that in this problem instance, the objective functions are similar enough that we can still use the downstream rounding process to produce high-quality trajectories. We still use the original 2-norm objective to minimize path length in the rounding stage. This can be seen as an instance of the technique, introduced in von Wrangel (2024), of using different costs and constraints for the relaxation and rounding stages.

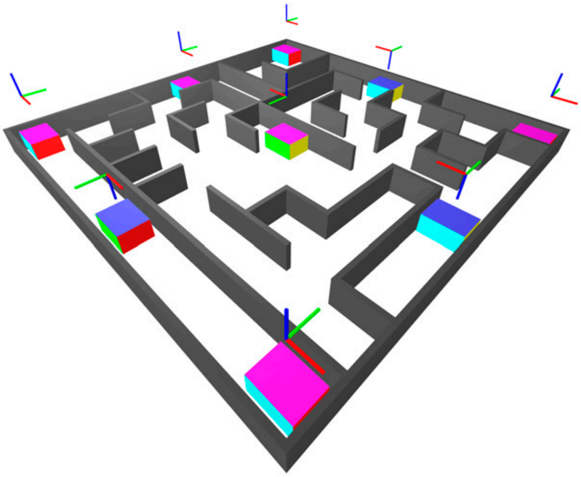

Figure 12. The maze environment used for an SE (3) planning problem. The block is allowed to move freely within the maze, including arbitrary rotations and translations in 3D space. The nine configurations shown have various orientations, which are indicated by the coordinate frames superimposed above each configuration. Note that some of these orientations include requiring a complete flip of 180° about a horizontal axis (e.g., between the top-right and top-center configurations).

We also use relatively loose tolerances when solving the relaxation to further improve runtimes. (In particular, we use 10^{-3} as the termination tolerance for primal feasibility, dual feasibility, and relative gap.) This is acceptable because the downstream rounding process is robust to such inaccuracies; the flows are known to only heuristically indicate the shortest path, even when the convex relaxation is solved to an arbitrarily tight numerical tolerance. In the rounding stage, each individual optimization problem is much smaller, so we can use a tight tolerance to ensure constraints on the trajectory are properly enforced. The average solve time with this approach was 354.98 s, and we visualize a planned trajectory in Figure 13.

An alternative approach to solving GCS problems with excessively large graphs is the GCS* algorithm (Chew Chia et al., 2024). GCS* solves a shortest path problem on a GCS using graph search techniques, similar to the famous A* search algorithm (Hart et al., 1968). The average solve time with GCS* was only 46.71 s—13.5% of the runtime of the relax-and-round solution method. And the paths produced by GCS* were only 4.4% longer than those found with the relax-and-round strategy.

8. Discussion

In this paper, we have formulated the general problem of motion planning around obstacles on Riemannian manifolds as a shortest path problem in a graph of geodesically convex sets, and we have presented sufficient conditions

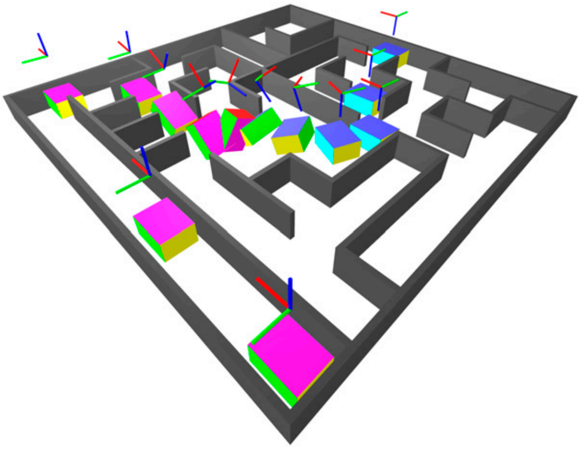

Figure 13. Solution path for a planning problem in the maze environment. Paths were obtained with the relax-and-round strategy.

under which this formulation inherits the same guarantees as in the ordinary Euclidean case. We describe how these theoretical developments inform simple and elegant modifications to the original GcsTrajOpt, in order to handle robots with mobile bases and continuous revolute joints. This enables us to solve motion planning problems on such robotic platforms to global optimality and guarantee that the trajectory is collision-free at every point in time. Approximate solving techniques still guarantee that trajectories are collision-free, and empirically, such trajectories are very close to optimal.

The PL approximation strategy has the benefit of being applicable to arbitrary manifolds, and can produce arbitrarily fine approximations given sufficient computation time. However, it remains to be seen whether that runtime will become prohibitive if this strategy is used for more complex robot configuration spaces. The use of heuristic strategies to handle large problems, such as GCS* (Chew Chia et al., 2024) is a promising direction for keeping computational costs low even as the problems grow in complexity. Another limitation of the PL approximation strategy is the lack of a general approach for producing collision-free regions. For now, we must rely on workarounds such as flat parametrizations and accept the resulting distortion of the objective function.

We have demonstrated that GgcsTrajOpt is a powerful tool for robot motion planning. It is capable of producing plans for high degree-of-freedom systems operating in obstacle-dense configuration spaces, such as a PR2 bimanual mobile manipulator reaching into and out of shelves. Although the planning and optimization frameworks used in GgcsTrajOpt are still in their infancy, they are already capable of producing high-quality results that are competitive with existing methods. As further research and technical improvements are made, its performance will continue to improve.

Acknowledgements

The authors would like to thank Tobia Marcucci and Seiji Shaw for their valuable suggestions throughout the course of this work. The authors would also like to thank David von Wrangel for his assistance with the implementation of the PRM comparisons and Shao Yuan Chew Chia for applying the GCS* algorithm to our problem domain.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by (in alphabetical order) Amazon.com, PO No. 2D-06310236 and the Frederick and Barbara Cronin Fellowship.

ORCID iD

Thomas Cohn https://orcid.org/0000-0002-5411-0710

References

Aceituno-Cabezas B, Mastalli C, Dai H, et al. (2017) Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization. *IEEE Robotics and Automation Letters* 3(3): 2531–2538.

Adu-Bredu A, Devraj N and Jenkins OC (2022) Optimal constrained task planning as mixed integer programming. ArXiv preprint arXiv:2211.09632.

Amice A, Dai H, Werner P, et al. (2023) Finding and optimizing certified, collision-free regions in configuration space for robot manipulators. In: International workshop on the algorithmic foundations of robotics, College Park, MD, 22–24 June 2022, 328–348. Springer.

Atceken M and Keles S (2003) On the product Riemannian manifolds. *Differential Geometry - Dynamical Systems* 5(1): 1–7.

Bacak M (2014) *Convex Analysis and Optimization in Hadamard Spaces*. Berlin, Germany: Walter de Gruyter GmbH & Co KG, Vol. 22.

Blumenson L (1960) A derivation of n-dimensional spherical coordinates. *The American Mathematical Monthly* 67(1): 63–66.

Boumal N (2022) *An Introduction to Optimization on Smooth Manifolds*. Cambridge, UK: Cambridge University Press. URL: https://www.nicolasboumal.net/book.

Boyd S, Boyd SP and Vandenberghe L (2004) *Convex Optimization*. Cambridge, UK: Cambridge University Press. URL: https://web.stanford.edu/boyd/cvxbook/.

Burago D, Burago Y and Ivanov S (2001) *A Course in Metric Geometry*. Providence, RI: American Mathematical Society, Vol. 33.

Bylard A, Bonalli R and Pavone M (2021) Composable geometric motion policies using multi-task pullback bundle dynamical systems. In: 2021 IEEE international conference on robotics and automation (ICRA), Xi'an, China, 30 May–5 June 2021, 7464–7470. IEEE.

Cartan É (1983) Geometry of Riemannian spaces: lie groups; history. In: *Frontiers and Applications Series*. Berkeley CA: Math Science Press, Vol. 13.

Cheeger J, Ebin DG and Ebin DG (1975) *Comparison Theorems in Riemannian Geometry*. Amsterdam, the Netherlands: North-Holland publishing company, Vol. 9.

Chen J, Li J, Huang Y, et al. (2022) Cooperative task and motion planning for multi-arm assembly systems. ArXiv preprint arXiv:2203.02475.

Cheng CA, Mukadam M, Issac J, et al. (2018) RMPflow: a computational graph for automatic motion policy generation. In: International workshop on the algorithmic foundations of robotics, Merida, Mexico, 9–11 December 2018, 441–457. Springer.

Chew Chia SY, Jiang RH, Graesdal BP, et al. (2024) GCS*: forward heuristic search on implicit graphs of convex sets. In: Proceedings of the 16th international workshop on the algorithmic foundations of robotics (WAFR), Chicago, IL, 7 October 2024 (To appear).

Ciarlet PG (1988) Mathematical elasticity volume I: three-dimensional elasticity. In: *Studies in Mathematics and Its Applications*. London, UK: Elsevier Science.

Cignoni P, Rocchini C and Scopigno R (1998) *Metro: Measuring Error on Simplified Surfaces*. Hoboken, NJ: Wiley Online Library, Vol. 17, pp. 167–174.

Cohn T, Devraj N and Jenkins OC (2022) Topologically-informed atlas learning. In: 2022 international conference on robotics and automation (ICRA), Philadelphia, PA, 23–27 May 2022, 3598–3604. IEEE.

Cohn T, Petersen M, Simchowitz M, et al. (2023) Non-Euclidean motion planning with graphs of geodesically-convex sets. In: Proceedings of robotics: science and systems, Daegu, Republic of Korea, 15–19 July 2023. DOI: 10.15607/RSS.2023.XIX.057.

Cohn T, Shaw S, Simchowitz M, et al. (2024) Constrained bimanual planning with analytic inverse kinematics. In: 2024 IEEE international conference on robotics and automation (ICRA), Yokohama, Japan, 13–18 May 2024, 6935–6942. IEEE.

Dahl VA, Dahl AB and Larsen R (2014) Surface detection using round cut. In: 2014 2nd international conference on 3D vision, Tokyo, Japan, 8–11 December 2014, 82–89. IEEE.

Dai H, Izatt G and Tedrake R (2019) Global inverse kinematics via mixed-integer convex optimization. *The International Journal of Robotics Research* 38(12–13): 1420–1441.

Dam EB, Koch M and Lillholm M (1998) *Quaternions, Interpolation and Animation*. Princeton, NJ: Citeseer, Vol. 2.

Dankowicz H and Schilder F (2013) *Recipes for Continuation*. New Delhi, India: SIAM.

Deits R and Tedrake R (2014) Footstep planning on uneven terrain with mixed-integer convex optimization. In: 2014 IEEE-RAS international conference on humanoid robots, Atlanta, GA, 18–20 November 2014, 279–286. IEEE.

Deits R and Tedrake R (2015) Computing large convex regions of obstacle-free space through semidefinite programming. In: *Algorithmic Foundations of Robotics XI*. Berlin, Germany: Springer, 109–124.

Do Carmo MP (1992) *Riemannian Geometry*. Berlin, Germany: Springer, Vol. 6.

Ghosh A (2012) *Piecewise Linear Reconstruction and Meshing of Submanifolds of Euclidean Space*. PhD Thesis, Université de Nice Sophia Antipolis, Nice, France.

Gill PE, Murray W and Saunders MA (2005) SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Review* 47: 99–131.

Gill PE, Murray W, Saunders MA, et al. (2018) *User's Guide for SNOPT 7.7: Software for Large-Scale Nonlinear Programming. Center for Computational Mathematics Report CCoM 18-1, Department of Mathematics*. San Diego, CA: University of California.

Gurobi Optimization LLC (2023) *Gurobi Optimizer Reference Manual*. Beaverton, OR: Gurobi Optimization LLC.

Hart PE, Nilsson NJ and Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2): 100–107.

Henderson ME (2002) Multiple parameter continuation: computing implicitly defined k-manifolds. *International Journal of Bifurcation and Chaos* 12(03): 451–476.

Hillier FS and Lieberman GJ (2015) *Introduction to Operations Research*. New York, NY: McGraw-Hill.

Ioan D, Prodan I, Olaru S, et al. (2021) Mixed-integer programming in motion planning. *Annual Reviews in Control* 51: 65–87.

Jaillet L and Porta JM (2013) Path planning under kinematic constraints by rapidly exploring manifolds. *IEEE Transactions on Robotics* 29(1): 105–117.

Kalakrishnan M, Chitta S, Theodorou E, et al. (2011) STOMP: stochastic trajectory optimization for motion planning. In: 2011 IEEE international conference on robotics and automation, Shanghai, China, 9–13 May 2011, 4569–4574. IEEE.

Kavraki LE, Svestka P, Latombe JC, et al. (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12(4): 566–580.

Klein H, Jaquier N, Meixner A, et al. (2022) A Riemannian take on human motion analysis and retargeting. In: 2022 IEEE/RSJ international conference on intelligent robots and systems (IROS), Kyoto, Japan, 23–27 October 2022, 5210–5217. IEEE.

Knyazev AV and Zhu P (2012) Principal angles between subspaces and their tangents. ArXiv preprint arXiv:1209.0523.

Landry B, Deits R, Florence PR, et al. (2016) Aggressive quadrotor flight through cluttered environments using mixed integer programming. In: 2016 IEEE international conference on robotics and automation (ICRA), Stockholm, Sweden, 16–21 May 2016, 1469–1475. IEEE.

LaValle SM (1998) Rapidly-exploring random trees: a new tool for path planning. *The Annual Research Report*.

Lee JM (2013) Smooth manifolds. In: *Introduction to Smooth Manifolds*. Berlin, Germany: Springer.

Lee JM (2018) *Introduction to Riemannian Manifolds*. Berlin, Germany: Springer, Vol. 176.

Liverani C (2019) Implicit function theorem 1 (a quantitative version). retrieved January 13. https://www.mat.uniroma2.it/˜liverani/Calcolo1-2016/implicit.pdf.

Marchand H, Martin A, Weismantel R, et al. (2002) Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics* 123(1–3): 397–446.

Marcucci T, Umenberger J, Parrilo PA, et al. (2021) Shortest paths in graphs of convex sets. ArXiv preprint arXiv:2101.11565.

Marcucci T, Petersen M, von Wrangel D, et al. (2023) Motion planning around obstacles with convex optimization. *Science Robotics* 8(84): eadf7843.

MOSEK ApS (2019) MOSEK optimization suite.

Myers SB and Steenrod NE (1939) The group of isometries of a Riemannian manifold. *Annals of Mathematics* 40: 400–416.

Park FC and Ravani B (1995) Be´zier curves on Riemannian manifolds and Lie groups with kinematics applications. *Journal of Mechanical Design* 117(1): 36–40.

Petersen M and Tedrake R (2023) Growing convex collision-free regions in configuration space using nonlinear programming. ArXiv preprint arXiv:2303.14737.

Phillips-Grafflin C (2023) Common robotics Utilities.

Popiel T and Noakes L (2007) Bézier curves and C^2 interpolation in Riemannian manifolds. *Journal of Approximation Theory* 148(2): 111–127.

Porta JM and Jaillet L (2010) Path planning on manifolds using randomized higher-dimensional continuation. In: Algorithmic foundations of robotics IX: selected contributions of the ninth international workshop on the algorithmic foundations of robotics, Singapore, 13–15 December 2010, 337–353. Springer.

Rana MA, Li A, Fox D, et al. (2021) Towards coordinated robot motions: end-to-end learning of motion policies on transform trees. In: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), Prague, Czech Republic, 27 September–1 October 2021, 7792–7799. IEEE.

Ratliff ND, Issac J, Kappler D, et al. (2018) Riemannian motion policies. ArXiv preprint arXiv:1801.02854.

Saha S and Julius AA (2017) Task and motion planning for manipulator arms with metric temporal logic specifications. *IEEE Robotics and Automation Letters* 3(1): 379–386.

Suh HT, Xiong X, Singletary A, et al. (2020) Energy-efficient motion planning for multi-modal hybrid locomotion. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), Las Vegas, NV, 25–29 October 2020, 7027–7033. IEEE.

Tedrake R (2022) Robotic manipulation. URL: https://manipulation.mit.edu.

Tedrake R, the Drake Development Team R (2019) Drake: model-based design and verification for robotics. URL: https://drake.mit.edu.

Tika A, Gashi F and Bajcinca N (2022) Robot online task and trajectory planning using mixed-integer model predictive control. In: 2022 European control conference (ECC), London, UK, 12–15 July 2022, 2005–2011. IEEE.

Toussaint M (2014) Newton methods for k-order Markov constrained motion problems. ArXiv preprint arXiv:1407.0414.

Väisälä J (2003) A proof of the Mazur-Ulam theorem. *The American Mathematical Monthly* 110(7): 633–635.

Valenzuela AK (2016) *Mixed-integer Convex Optimization for Planning Aggressive Motions of Legged Robots over Rough Terrain*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Van Wyk K, Xie M, Li A, et al. (2022) Geometric fabrics: generalizing classical mechanics to capture the physics of behavior. *IEEE Robotics and Automation Letters* 7(2): 3202–3209.

Vishnoi NK (2018) Geodesic convex optimization: differentiation on manifolds, geodesics, and convexity. ArXiv preprint arXiv:1806.06373.

von Wrangel D (2024) *Guiding Nonconvex Trajectory Optimization with Hierarchical Graphs of Convex Sets*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Werner P, Amice A, Marcucci T, et al. (2024) Approximating robot configuration spaces with few convex sets using clique covers of visibility graphs. In: 2024 IEEE international conference on robotics and automation (ICRA), Yokohama, Japan, 13–18 May 2024, 10359–10365. IEEE.

Wu Y, Spasojevic I, Chaudhari P, et al. (2024) Optimal convex cover as collision-free space approximation for trajectory generation. ArXiv preprint arXiv:2406.09631.

Yang L and Lavalle SM (2004) The sampling-based neighborhood graph: an approach to computing and executing feedback motion strategies. *IEEE Transactions on Robotics and Automation* 20(3): 419–432.

Yi JS, Luong TA, Chae H, et al. (2022) An online task-planning framework using mixed integer programming for multiple cooking tasks using a dual-arm robot. *Applied Sciences* 12(8): 4018.

Zhang H and Sra S (2016) First-order methods for geodesically convex optimization. In: Conference on learning theory, New York, NY, 23–26 June 2016, 1617–1638. PMLR.

Zucker M, Ratliff N, Dragan AD, et al. (2013) Chomp: covariant Hamiltonian optimization for motion planning. *The International Journal of Robotics Research* 32(9–10): 1164–1193.

Appendix

A. Proofs

A.1. Proof of Theorem 1.

Lemma 3. *For any $p, q \in \overline{\mathcal{M}}$, there is a piecewise-smooth path connecting p and q.*

Proof. Because $\mathcal{M}$ is path connected, there is a continuous curve $\gamma : [a, b] \rightarrow \overline{\mathcal{M}}$ joining them. Let (U_1, ψ_1), …, (U_n, ψ_n) be a series of charts of $\mathcal{Q}$ that cover the image of γ, with $p \in U_1$, $q \in U_n$, and $U_i \cap U_{i+1} \cap \overline{\mathcal{M}} \neq \emptyset$ for each i. (Such a finite covering exists because the image of γ is compact.) Let $t_0, \ldots, t_n \in [a, b]$ such that $t_0 = a$, $t_n = b$, and for each $i = 1, \ldots, n-1$, $\gamma(t_i) \in U_i \cap U_{i+1}$. For each $i = 1, \ldots, n-1$, let $\widetilde{\gamma}_i$ be a smooth curve joining $\psi_i(\gamma(t_i))$ to $\psi_{i+1}(\gamma(t_{i+1}))$ that is contained within $\psi_i(U_i \cap \overline{\mathcal{M}})$. Let $\widetilde{\gamma}_0$ join $\psi_1(\gamma(t_0))$ to $\psi_1(\gamma(t_1))$ and be contained within $\psi_1(U_1 \cap \overline{\mathcal{M}})$, and let $\widetilde{\gamma}_n$ join ψ_n $(\gamma(t_{n-1}))$ to $\psi_n(q)$ and be contained within $\psi_n(U_n \cap \overline{\mathcal{M}})$. Then by lifting each of these curves to $\overline{\mathcal{M}}$, and concatenating them, we obtain a piecewise-smooth curve connecting p and q.

Proof of Theorem 1. The proof follows by verifying that $\overline{\mathcal{M}}$ is a complete, locally compact length space, so that we can apply Theorem 2.5.23 of Burago et al. (2001: 50). A length space is a metric space in which the distance between any two points is given by the infimum of the arc lengths of all paths connecting those two points. A length space is complete if the distance between any two points is finite. Thus, $\overline{\mathcal{M}}$ inherits a length structure from $\mathcal{Q}$ (with the restriction to curves that are entirely contained in $\overline{\mathcal{M}}$). All topological manifolds are locally compact. To check that $\overline{\mathcal{M}}$ is complete, let $p, q \in \overline{\mathcal{M}}$. By Lemma 3, there is a piecewise-smooth curve connecting p and q, so the set of arc lengths of curves connecting p and q is nonempty. It is also bounded below, so its infimum is finite, and thus $\mathrm{d}(p, q)$ exists. We conclude that $\overline{\mathcal{M}}$ is a complete, locally compact length space.

A.2. Proof of Theorem 4.

We first prove some facts about locally isometric coordinate charts: the transition map between two locally isometric coordinate charts is a Euclidean isometry (Lemma 4), and the preimage under such a chart of a convex set is g-convex (Lemma 5). To prove Theorem 4, we construct a Riemannian normal coordinate system covering $\overline{\mathcal{Y}}$, and leverage the fact that such a chart is a local isometry for flat manifolds.

Lemma 4. *Let $(\mathcal{Z}_1, \psi_1)$ and $(\mathcal{Z}_2, \psi_2)$ be coordinate charts of $\mathcal{M}$, with ψ_1 and ψ_2 local isometries. If $\mathcal{Z}_1 \cap \mathcal{Z}_2$ is connected, then there is a Euclidean isometry ξ such that $\forall p \in \mathcal{Z}_1 \cap \mathcal{Z}_2$, $\psi_1(p) = (\xi \circ \psi_2)(p)$.*

Proof. $\psi_1 \circ \psi_2^{-1}$ is a local isometry between two connected open subsets of Euclidean space, so $(\psi_1 \circ \psi_2^{-1})_{*,p}$ is an orthogonal matrix for any p. Thus, we can apply Theorem 1.8–1 of Ciarlet (1988: 44).

Lemma 5. *Consider $\overline{\mathcal{Y}} \subseteq \mathcal{Z} \subseteq \mathcal{M}$, where $\mathcal{Z}$ is g-convex, and we have a coordinate chart $(\mathcal{Z}, \psi)$ such that ψ is a local isometry. If $\psi(\overline{\mathcal{Y}})$ is convex, then $\overline{\mathcal{Y}}$ is g-convex.*

Proof. Fix $p, q \in \overline{\mathcal{Y}}$. Then there is a unique minimizing geodesic γ connecting p to q, and γ is contained in $\mathcal{Z}$. Because ψ is a local isometry, it maps γ to a line segment in $\psi(\mathcal{Z})$. $\psi(p), \psi(q) \in \psi(\overline{\mathcal{Y}})$, so by convexity of $\psi(\overline{\mathcal{Y}})$, $\psi \circ \gamma$ is contained in $\psi(\overline{\mathcal{Y}})$. Thus, γ is contained in $\overline{\mathcal{Y}}$, so $\overline{\mathcal{Y}}$ is g-convex.

Proof of Theorem 4. For each $i \in [m]$, we can construct a ball $B_{s_i}(c_i) \supseteq \mathrm{proj}_{\mathcal{Q}_i}(\overline{\mathcal{Y}})$, with $s_i < r_i$. Define $\mathcal{Z} = \prod_{i \in [m]} B_{s_i}(c_i)$, a g-convex set. Consider the Riemannian normal coordinates of $\mathcal{Q}$ at $(c_1, \ldots, c_m)$. This coordinate system, restricted to $\mathcal{Z}$, induces a coordinate chart φ. Because $\mathcal{Q}$ is flat, φ is a local isometry, so by Lemma 4 there is a Euclidean isometry ξ such that $\varphi(\overline{\mathcal{Y}}) = \xi(\psi(\overline{\mathcal{Y}}))$, so $\varphi(\overline{\mathcal{Y}})$ is convex. Thus, by Lemma 5, $\overline{\mathcal{Y}}$ is g-convex.

A.3. Proofs of Lemmata 1 and 2. **Proof of Lemma 1.** Define $\gamma(t) = tx_0 + (1-t)x_1$. Then $(\Psi_i \circ \gamma)(0) = y_0$ and $(\Psi \circ \gamma)(1) = y_1$, so $\mathsf{d}(y_0, y_1) \leq L(\Psi \circ \gamma)$. Let $\Delta x = x_1 - x_0$. Well,

$$\begin{aligned} \|(\Psi \circ \gamma)'(t)\|_2 &= \|D\Psi(t)\gamma'(t)\|_2 = \|D\Psi(t)\Delta x\|_2 \\ &= \|\Delta x + (D\Psi_i(t) - I)\Delta x\|_2 \\ &\leq \|\Delta x\|_2 + \|(D\Psi(t) - I)\Delta x\|_2 \\ &\leq \|\Delta x\|_2 + \|D\Psi(t) - I\|_{\mathrm{op}} \|\Delta x\|_2, \end{aligned}$$

where we have leveraged the triangle and operator norm inequalities. By considering the largest possible operator norm over $\psi(\overline{\mathcal{Y}}_i)$, we obtain

$$\begin{aligned} L(\Psi \circ \gamma) &\leq \int_0^1 \|\Delta x\|_2 + \|D\Psi(t) - I\|_{\mathrm{op}} \|\Delta x\|_2 \mathrm{d}t \\ &\leq \int_0^1 \|\Delta x\|_2 \left(1 + \max_{x \in \psi(\overline{\mathcal{Y}}_i)} \|D\Psi(x) - I\|_{\mathrm{op}}\right) \mathrm{d}t \\ &= \|\Delta x\|_2 \left(1 + \max_{x \in \psi(\overline{\mathcal{Y}}_i)} \|D\Psi(x) - I\|_{\mathrm{op}}\right), \end{aligned}$$

which is the desired upper bound.

Proof of Lemma 2. Construct a triangle whose three vertices are $x_{i,1} = x_{i+1,0}$, $y_{i,1}$, and $\psi_{i+1}(y_{i,1})$. We note that $\angle x_{i,1} y_{i,1} \psi_{i+1}(y_{i,1})$ is equal to the angle between $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ along the plane through those three points (which we call α), and hence is bounded above by $\alpha_{\max}$.

$$\begin{aligned} \|x_{i+1,0} - \psi_{i+1}(y_{i,1})\|_2 &= \|x_{i+1,0} - y_{i,1}\|_2 \sin \alpha \\ &\leq \epsilon_H \sin \alpha_{\max} \end{aligned}$$

Applying Lemma 1 to $x_{i+1,0}$ and $\psi_{i+1}(y_{i,1})$ completes the proof.

A.4. Proof of Theorem 6. **Proof of Theorem 6.** Fix a neighborhood $\mathcal{U}$ of A_1. For the remainder of the proof, all Riemannian norms and inner products are acting on tangent vectors in $T_{A_1}\mathcal{M}$, so we omit the subscripts for notational brevity. Since the sectional curvature is invariant with respect to a change of basis, suppose without loss of generality that $\|u\|_{A_1} = \|v\|_{A_1} = 1$ and $\langle u, v \rangle_{A_1} = 0$. To prove this result, we will construct a geodesic γ on $\mathcal{U}^2$ such that $\mathsf{d} \circ \gamma$ achieves smaller values on its endpoints than at its center. This relies on the properties of specially constructed Levi-Civita parallelogramoids on $\mathcal{U}$.

Let $\epsilon > 0$ be small, such that $\exp_p(\overline{B_{2\epsilon}(0)}) \subseteq \mathcal{U}$. Let $\alpha : t \mapsto \exp_p(tv)$, let $B_1 = \alpha(\epsilon)$, and let $u' = \Gamma(\alpha)_0^{\epsilon}(u)$ be u parallel transported from A_1 to B_1. Let $\gamma_A : t \mapsto \exp_{A_1}(tu)$ and $\gamma_B : t \mapsto \exp_{B_1}(tu')$, with domain $[-\epsilon, \epsilon]$. Then $\gamma = (\gamma_A, \gamma_B)$ is a geodesic of $\mathcal{U}^2$. Define $A_0 = \gamma_A(-\epsilon)$, $B_0 = \gamma_B(-\epsilon)$, $A_2 = \gamma_A(\epsilon)$, and $B_2 = \gamma_B(\epsilon)$. This construction is visualized in Figure 14. We want to show that $\mathsf{d}(A_0, B_0) < \mathsf{d}(A_1, B_1)$ and $\mathsf{d}(A_2, B_2) < \mathsf{d}(A_1, B_1)$.

The points A_1, B_1, A_2, and B_2 describe a Levi-Civita parallelogramoid, with base A_1B_1 and suprabase A_2B_2. Thus, we can relate the length of the base and suprabase via the formula of Cartan (1983: 244):

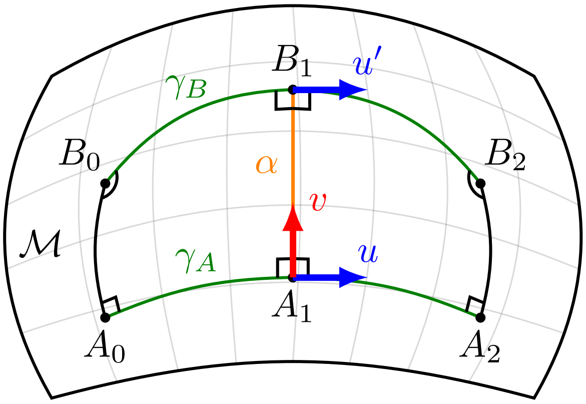

Figure 14. The construction of two Levi-Civita parallelogramoids used in the proof of Theorem 6. $\mathsf{d}(A_0, B_0) < \mathsf{d}(A_1, B_1)$ and $\mathsf{d}(A_2, B_2) < \mathsf{d}(A_1, B_1)$, which demonstrates the nonconvexity of d around A_1. $\angle A_0B_0B_1$ and $\angle A_2B_2B_1$ are obtuse.

$$\mathsf{d}(A_2,B_2)^2 = \mathsf{d}(A_1,B_1)^2 + \frac{8}{3}\langle\mathcal{R}(\epsilon u,\epsilon v)\epsilon u,\epsilon v\rangle + O(\epsilon^5).$$

Because $||u|| = ||v|| = 1$ and $\langle u,v\rangle = 0$,

$$\langle\mathcal{R}(\epsilon u,\epsilon v)\epsilon u,\epsilon v\rangle = -\epsilon^4\langle\mathcal{R}(u,v)v,u\rangle = -\epsilon^4 K(u,v) < 0.$$

So as ϵ is decreased toward 0, the fifth and higher order terms vanish, and $\mathsf{d}(A_2,B_2) < \mathsf{d}(A_1,B_1)$. A similar calculation shows that $\mathsf{d}(A_0,B_0) < \mathsf{d}(A_1,B_1)$. Thus, $d \circ \gamma$ has a local minimum, so we conclude that d is nonconvex on $\mathcal{U}$.

B. Experiment implementation details

In this appendix, we present further details about the setup of our experiments and demonstrations. Throughout the robot experiments, we use the sum of the trajectory length and duration as our objective, as was done for the quadrotor and bimanual motion planning experiments in Marcucci et al. (2023). Given the presence of joint velocity and acceleration limits, adding a trajectory duration cost biases the planner toward producing a path that admits a faster traversal.

B.1. Planar arm. The trajectories shown in Subsection 7.2 were generated with a GGCS that had 19 sets. We generated IRIS regions for the start and goal configurations, and hand-picked several seed points along the narrow gap between the two lower obstacles to help ensure connectivity between the start and goal. We then generated the remaining IRIS regions with random seed points (chosen uniformly from C-Free with rejection sampling).

The GgcsTrajOpt results shown used the sum of the path length and path duration as the objective. We used the relax-and-round approximation strategy to produce the trajectories shown in the paper. The first trajectory had a path length of 7.749, and the second had a path length of 8.448. When solving the integer program with branch-and-bound, the first trajectory had a path length of 7.274, and the second had a path length of 8.008. (Note that the optimal solution for the latter trajectory still had the middle joint of the arm traverse more than 360°).

B.2. KUKA iiwa arm. The motions shown in Subsection 7.3 used regions generated from 18 seed points. The seeds consisted of one seed for each middle and top shelf (the bottom shelves are excluded because they are kinematically unreachable), one seed above each shelf, one seed directly between each shelf, and two seeds per shelf to aid moving between the top and middle shelves. Regions were generated for each seed with both an empty hand and a mug in the hand to aid both types of trajectory planning. Regions were post-processed to remove redundant hyperplanes with the `ReduceInequalities` algorithm from Drake.

GgcsTrajOpt minimized both time and path length of the trajectory while ensuring continuity of the path, velocity, and acceleration. For velocity limits, the real velocity limits of the KUKA iiwa hardware were used. Trajectories were computed using the relax-and-round approximation strategy.

B.3. PR2 bimanual mobile manipulator. To model the PR2 robot, we use the URDF file and object meshes included with Drake. For each link, we take the convex hull of the mesh and use that as the collision geometry. (Collisions annotated in Table 1 are determined based on the true collision geometry, not the convex hulls). The plans we produce take into account the robot's base joint, torso lift joint, and all arm joints (up to the final wrist rotation joint and gripper joints). All other joints are fixed.

For the experiments demonstrated in Subsection 7.4, we first constructed IRIS regions for each of the possible goals: reaching into each of the three shelves in a set with both arms, crossing right-over-left on the middle and bottom shelves, and crossing left-over-right. (See Figure 15 for a visualization of these cross-over poses). We then hand select a few intermediate seed points; the regions around these points are used to promote connectivity among the various shelf-reaching regions. We construct these regions for each set of shelves, except for the experiments where the start and goal are within the same set of shelves.

We take several actions to improve the efficiency of GgcsTrajOpt. To reduce the number of constraints needed, we simplify the IRIS regions by removing redundant halfspaces from their polyhedral representation, using the `ReduceInequalities` algorithm in Drake. We also only include shelf-reaching regions if they are the start or goal of the plan. This greatly reduces the size of the optimization problem, promoting faster solve times. Empirically, it also leads to a shorter trajectory, likely due to a tightening of the convex relaxation. For GgcsTrajOpt, we use the same objective as the planar arm experiments (the sum of the trajectory length and duration), and we use the relax-and-round strategy.

For the comparison to kinematic trajectory optimization (Drake-TrajOpt), we use the same objective as GgcsTrajOpt: the sum of the trajectory duration and length. However, the trajectories are parametrized as B-splines instead of linear segments (or Bézier curves if the extensions in Subsection 6.3 are utilized). The `KinematicTrajectoryOptimization`

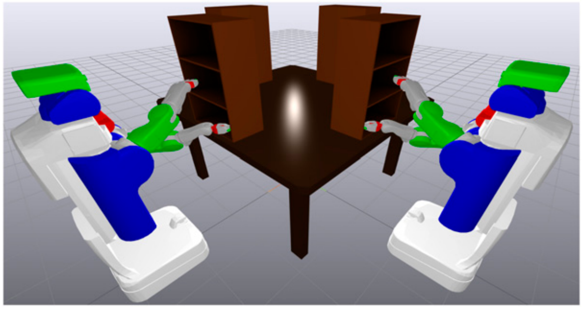

Figure 15. The start and goal pose for one of our motion planning experiments, using the PR2 bimanual mobile manipulator.

can automatically construct the nonlinear optimization problem for a given scenario, which we then solve with SNOPT. We first solve the problem without collision-free constraints. The output of this initial problem is used as the initial guess for the full problem (i.e., including collision-free constraints). The collision-free constraint is encoded with the `MinimumDistanceConstraint` class. We set a minimum distance of 1 mm and begin applying a penalty at 1 cm. This constraint is applied to 50 points along the trajectory. (Such a constraint can only be evaluated pointwise). For motion planning tasks where the robot had to move between shelves, Drake-TrajOpt was unable to produce a collision-free trajectory. Thus, we added waypoints near the beginning and end of the trajectory, in which the robot was in the same configuration as the start and goal (respectively), but the base was moved away from the shelf. This was only sometimes effective at finding collision-free trajectories.

As in Marcucci et al. (2023), we use the PRM planner from the Common Robotics Utilities library (Phillips-Grafflin, 2023), with the modifications described in Subsection 7.4. Given a piecewise-linear trajectory from the PRM, we construct a B-spline that passes through the nodes on this trajectory for use as an initial guess for Drake-TrajOpt. In this case, when solving the optimization problem, we begin applying a distance penalty at 1 m and perform collision checking at 100 points along the trajectory.