GCS Path Planning: SDP Relaxation and Formulation

Antigravity

November 18, 2025

1 Introduction

This document details the mathematical formulation for the Graph of Convex Sets (GCS) path planning algorithm, specifically focusing on the Semidefinite Programming (SDP) relaxation used to minimize the L2 norm of the trajectory velocity.

2 Convex Decomposition

The goal is to decompose the free space $\mathcal{F} = \mathcal{W} \setminus \bigcup_{j=1}^{M} \mathcal{O}_j$, where $\mathcal{W}$ is the rectangular domain and $\mathcal{O}_j$ are convex obstacles, into a set of convex polygons $\{R_1, \ldots, R_N\}$.

2.1 Hole Integration

Since the free space is multiply-connected (contains holes), we first transform it into a simply-connected polygon P.

1. Let P_0 be the boundary of $\mathcal{W}$ (ordered Counter-Clockwise).
2. For each obstacle $\mathcal{O}_j$ (ordered Clockwise):
    - Find a "bridge" connecting a vertex $v \in \mathcal{O}_j$ to a visible vertex $u \in P_{current}$.
    - Insert the sequence of $\mathcal{O}_j$ vertices into $P_{current}$ at u, effectively merging the hole into the boundary.
3. The result is a single polygon P (possibly with self-touching edges) that represents $\mathcal{F}$.

2.2 Partitioning

We decompose P into convex regions using a heuristic approach based on resolving reflex vertices (vertices with internal angle $> 180°$).

Algorithm: Convex Decomposition

1. Initialize $Regions \leftarrow \emptyset$.
2. While P is not convex:
    (a) Find a diagonal $d = (v_i, v_k)$ such that:
        - The polygon formed by $v_i, \ldots, v_k$ is convex.
        - The diagonal d does not intersect any other edges of P.
        - No other vertices of P lie inside the formed polygon.

    (b) Let $C = \{v_i, \ldots, v_k\}$.

    (c) $Regions \leftarrow Regions \cup \{C\}$.

    (d) $P \leftarrow P \setminus C$ (Update boundary of P).
3. $Regions \leftarrow Regions \cup \{P\}$.

3 Trajectory Optimization (SDP Relaxation)

We solve a relaxed version of the GCS problem using Semidefinite Programming (SDP). The binary variables are relaxed to continuous variables, resulting in a "flow" solution rather than a discrete path.

3.1 Sets and Indices

- V: Set of convex regions (polygons), indexed by i.
- E: Set of edges (i, j) representing adjacency between region i and j.
- K: Degree of the Bézier curve (we use $K = 2$ for quadratic).
- d: Dimension ($d = 2$).

3.2 Variables

- $y_i \in [0, 1]$: Continuous variable representing the activation of region i.
- $z_{ij} \in [0, 1]$: Continuous variable representing the flow from region i to j.
- $x_{i,k} \in \mathbb{R}^2$: Control point k ($k \in \{0, \ldots, K\}$) for the Bézier curve in region i.
- $t_{i,k} \in \mathbb{R}$: Slack variable for the L2 norm of the velocity vector.

3.3 Optimization Problem

The objective is to minimize the sum of the L2 norms of the velocity vectors (path length approximation).

$$\min \sum_{i \in V} \sum_{k=1}^{K} t_{i,k}$$

Subject to:

1. Flow Conservation

Standard network flow constraints adapted for the relaxed variables:

$$\begin{aligned}
&\sum_{j:(s,j)\in E} z_{sj} - \sum_{j:(j,s)\in E} z_{js} = 1 \quad \text{(Start Node } s\text{)} \\
&\sum_{j:(g,j)\in E} z_{gj} - \sum_{j:(j,g)\in E} z_{jg} = -1 \quad \text{(Goal Node } g\text{)} \\
&\sum_{j:(i,j)\in E} z_{ij} - \sum_{j:(j,i)\in E} z_{ji} = 0 \quad \forall i \in V \setminus \{s, g\} \\
&y_i \geq \sum_j z_{ij}, \quad y_i \geq \sum_j z_{ji}
\end{aligned}$$

2. L2 Norm via Schur Complement (SDP)

We want to enforce $t_{i,k} \geq \|v_{i,k}\|_2$, where $v_{i,k} = x_{i,k} - x_{i,k-1}$. This is equivalent to $t_{i,k}^2 \geq v_{i,k}^T v_{i,k}$ (for $t_{i,k} \geq 0$). Using the Schur Complement, this can be written as a Linear Matrix Inequality (LMI):

$$\begin{pmatrix} t_{i,k} I_d & v_{i,k} \\ v_{i,k}^T & t_{i,k} \end{pmatrix} \succeq 0$$

where I_d is the $d \times d$ identity matrix.

3. Containment

$$\begin{aligned}
&A_i x_{i,k} \leq b_i + M(1 - y_i) \quad \forall i \in V, k \in \{0, \ldots, K\} \\
&-My_i \leq x_{i,k} \leq My_i \quad \text{(Force } x_{i,k} = 0 \text{ if } y_i = 0\text{)}
\end{aligned}$$

4. Continuity (C^0)

$$\|x_{i,K} - x_{j,0}\|_\infty \leq M(1 - z_{ij}) \quad \forall (i,j) \in E$$

5. Heading Consistency (C^1)

$$\|(x_{i,K} - x_{i,K-1}) - (x_{j,1} - x_{j,0})\|_\infty \leq M(1 - z_{ij}) \quad \forall (i,j) \in E$$

6. Boundary Conditions

$$x_{s,0} = p_{\text{start}}, \quad x_{g,K} = p_{\text{goal}}$$

Note: M is a sufficiently large constant ("Big-M"). The relaxation allows the problem to be solved by convex solvers like SCS, which support PSD cones but not integer variables. The result is a flow indicating the optimal path.