Conical Routing For Lookahead

Gaurish Telang

gaurish108@gmail.com

March 6, 2022
6:17am

Contents

Page

1 Introduction . . . 1

2 Basic scheme . . . 2

1 Introduction

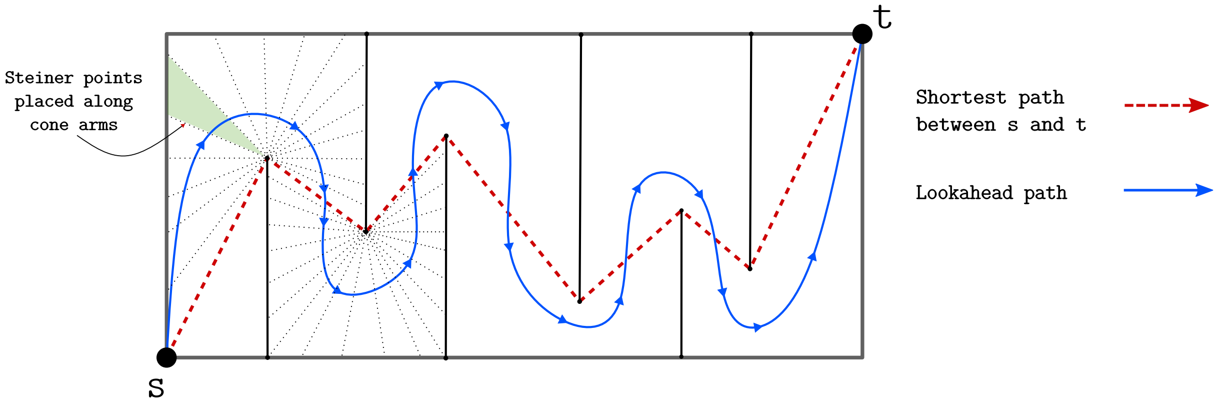
t
Steiner points placed along cone arms
s
Shortest path between s and t
Lookahead path

Figure 1: Conical Routing between start s and destination t

We report some results on schemes in computing paths with lookahead in the stalactite and stalagmite setting. In a path with lookahead any point on the path can see a large chunk of the path immediately ahead of it. [1] More precisely, given a parameter $l \in \mathbb{R}^+$ we want to compute a shortest path between s and t such that point on the computed path satisfies one of either of the two following properties

- The point can see at least a subpath of length $\geq l$ immediately ahead of it
- The part of the subpath that the point can see ahead of it contains t

We will give a set of schemes for doing this in the alternating *stalactites and stalagmites* — henceforth shortened to "*ites" — such that the shortest path between two given points s and t is taut against the tips of the "*ites".

[1] we are using euclidean vision where two points A and B see one another iff the straight line segment AB is contained in the closure of the free space.

2 Basic scheme

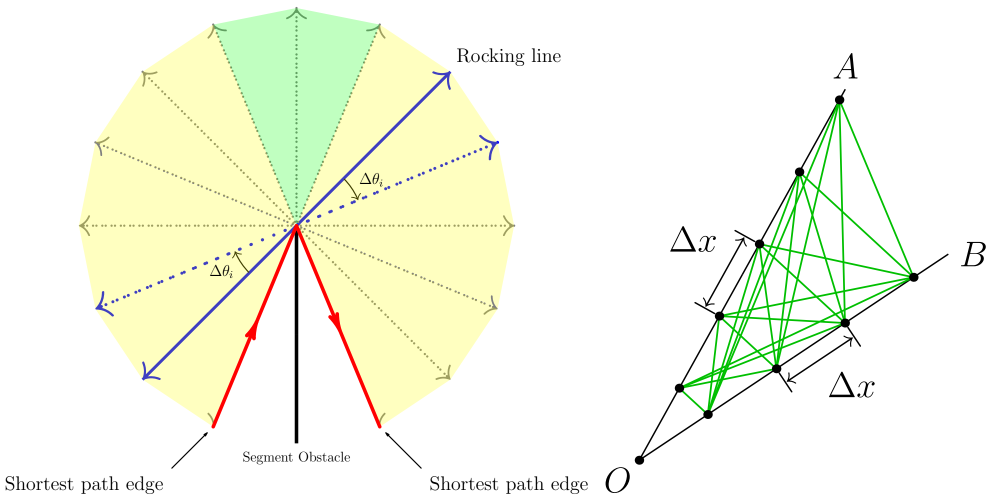

Figure 2: (Left) A rocking line (blue) creates a sequence of cones of angles $\Delta\theta_i$ between two successive shortest path edges. (Right) Complete bipartite graph (green) between points on two arms of a cone on the Steiner points. Distance between two consecutive Steiner points along each arm is Δx. O is the tip of the stalactite/stalagmite.

- At each tip T of the *ites, consider the two shortest path edge between s and t incident at that point. Divide the angle between them into yellow and green wedges as shown in the left hand side Figure 2. Each yellow wedge is divided into cones of small angle $\Delta\theta$ whereas the green wedge is divided into small cones again of size $\Delta\theta$, allowing for one of the cones to have angle $\leq \Delta\theta$
- Place Steiner points separated by a small distance Δx starting from the obstacle segment tip along each arm of every cone, for as long as the cone arm lies inside free space. Also add the endpoints of the segments as Steiner points of those respective segments. [2]
- Draw the complete bipartite graph between the Steiner points of each cone as shown in Figure 2
- Solve the following integer program.

$$\min \sum_{e \in E} c_e x_e = 2 \tag{1}$$

such that

$$x_e \in \{0, 1\} \tag{2}$$

$$\alpha + \beta = \gamma \tag{3}$$

This scheme has been implemented in Python 2.7.12 using the Python bindings of CGAL (especially those of its kernel) on a Thinkpad laptop running Linux Mint.

[2] computing these is just a simple matter of ray-shooting.