Tracking the trajectory of a Parade

Scribe: Shikha Singh

March 6, 2014

1 Introduction

Suppose you have a network of paths and a parade passes through it. The parade has a fixed trajectory that is known, starting and ending at some fixed points. The observer A wants to view the parade as much as possible, by some metric which we define later. If she was a superheroine she could always just catch up to the parade and stay with it, rendering the problem trivial. So we would assume that the observer's speed is less than or equal to that of the parade. Given a starting location, different from that of the parade, A's task is to navigate through the network in a way as to maximize its view of the parade.

Here we assume that the trajectory of the parade is fixed and known to the observer. If we relax this assumption, we can model problems like that of the police chasing after a crimilar through a network, where the path the criminal chooses may vary and is not known to the police.

2 Objective

We could have several objectives, two of which we discussed are:

1. **Maximize number of viewings** – What counts as a *viewing* depends on how we define the discrete model of the problem.
2. **Maximise the amount of time the observer sees the parade**.

3 Notations and Model

Let the network of paths be a graph with vertices which are discrete viewpoints. Let $\tilde{\tau}$ be the trajectory of the parade at time t, which is the input to the problem. We say A *views* the parade at time t if A is at a vertex adjacent to that of the parade. Let S be the starting point of the observer A. Let s_P and s_B be the speed of the parade and the observer respectively. We assume that $s_P \leq s_B$.

We model the above problem as that of longest path in a DAG. We define the DAG as follows. The vertices of the DAG are 2-tuples constituting of the position of the observer and the position of the parade. We have an edge between (v_i, p_k) and (v_j, p_ℓ) iff node v_i *views* parade p_k and v_j views p_ℓ, and $d(v_i, v_j) \leq t_l - t_k$. In other words the vertices of the graph are only those nodes which count as *views*, and there is an edge between them iff the observer can make it from one point to another in time. Note that the monotonicity of time/parade ensures that the above is in fact a directed acyclic graph.

4 A Solution and Open Problems

The above model solves the discrete version of the problem by finding the longest path through this DAG from the starting node S to the ending node T. But how efficient is our method? What is the size of our resulting DAG and thus the running time of our algorithm?

Other open problems raised were as follows.

- What if there are two observers? The DAG would then have $(v_i, v_{i'}, p_k)$ adjacent to $(v_j, v_{j'}, p_\ell)$ iff at least one of the observer's sees p_k at time t_k and at least on of them sees p_ℓ at time t_ℓ, such that $d(v_i, v_j) \leq t_l - t_k$ and $d(v_{i'}, v_{j,}) \leq t_l - t_k$.
- What about N observers? We believe that it is hard for k observers. Can we prove it?
- *Uncertainity* – Suppose that the parade P arrives at p_k during the time window $[t_k, t_\ell]$.
- What is the maximum amount of time the parade is seen when we give credit for being on an edge too, i.e., if we give credit to the observer for being on the edge (p_k, p_{k+1}) while the parade is on it.