

Offline Trajectory Optimization of Autonomous Vehicles in a Network of Intersections

Jeroen van Riel

March 2025

Contents

1 Trajectories in networks	1
1.1 General decomposition	2
1.2 Decomposition for delay objective	3

1 Trajectories in networks

We now extend the single intersection model to a network of intersections without turning routes, illustrated in Figure 1. We define a directed graph (V, E) with nodes V and arcs E , representing the possible paths that vehicles can follow. Nodes of in-degree at least two are called *intersections*. Nodes with only outgoing arcs are *entrypoints* and nodes with only incoming arcs are *exitpoints*. Let $d(v, w)$ denote the distance between nodes v and w . For each route index $r \in \mathcal{R}$, we let

$$\bar{V}_r = (v_r(0), v_r(1), \dots, v_r(m_r), v_r(m_{r+1}))$$

be the path that vehicles $i \in \mathcal{N}_r$ follow through the network. We require that the first node $v_r(0)$ is an entrypoint and that the last node $v_r(m_{r+1})$ is an exitpoint and we write

$$V_r = \bar{V}_r \setminus \{v_r(0), v_r(m_{r+1})\}$$

to denote the path restricted to intersections. We say that some $(v, w) \in E$ is on path V_r whenever v and w are two consecutive nodes on the path and we write E_r to denote the set of all these edges. We require that routes can only overlap at nodes by making the following assumption.

Assumption 1.1. *Every arc $(v, w) \in E$ is part of at most one route V_r .*

We start by considering networks in which all roads are axis-aligned such that intersections always involve perpendicular lanes and where routes are such that no turning is required. For each $v \in V_r$ define the conflict zone $\mathcal{E}_r(v) = (b_r(v), e_r(v))$ and consider the union

$$\mathcal{E}_r = \bigcup_{v \in V_r} \mathcal{E}_r(v)$$

corresponding to the positions of vehicles $i \in \mathcal{N}_r$ for which it occupies an intersection on its path V_r . By reading $\mathcal{E}_i \equiv \mathcal{E}_r$ for $r(i) = r$, the single intersection problem naturally extends to the network case. Like before, the resulting problem can be numerically solved by a direct transcription method.

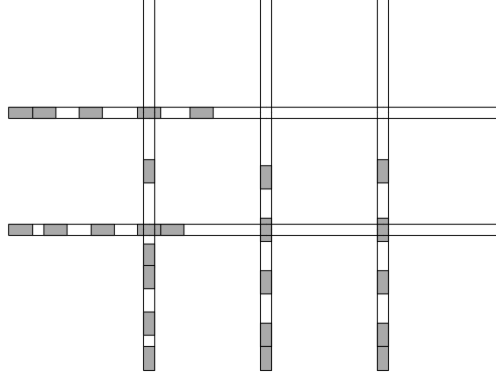


Figure 1: Illustration of some grid-like network of intersections with vehicles drawn as grey rectangles. There are five vehicle routes: two from east to west and three from south to north. Turning at intersections is not allowed.

1.1 General decomposition

The general two-stage decomposition for the single intersection extends rather naturally to the present model. Let for each pair (i, v) of some vehicle $i \in \mathcal{N}$ and an intersection $v \in V_{r(i)}$ along its route, let

$$\inf\{t : x_i(t) \in \mathcal{E}_r(v)\} \quad \text{and} \quad \sup\{t : x_i(t) \in \mathcal{E}_r(v)\}$$

be the crossing time and exit time, which we denote by $y(i, v)$ and $y(i, v) + \sigma(i, v)$, respectively. Instead of a single set of conflicts, we now define for each intersection $v \in V$ in the network the set of conflict pairs

$$\mathcal{D}^v = \{\{i, j\} \subset \mathcal{N} : r(i) \neq r(j), v \in V_{r(i)} \cap V_{r(j)}\}.$$

Now the two-stage approach is to solve

$$\begin{aligned} \min_{y, \sigma} \quad & \sum_{r \in \mathcal{R}} F(y_r, \sigma_r) \\ \text{s.t.} \quad & y(i, v) + \sigma(i, v) \leq y(j, v) \quad \text{or} \\ & y(j, v) + \sigma(j, v) \leq y(i, v), \quad \text{for all } \{i, j\} \in \mathcal{D}^v \text{ and } v \in V, \\ & (y_r, \sigma_r) \in \mathcal{S}_r, \quad \text{for all } r \in \mathcal{R}, \end{aligned}$$

where $F(y_r, \sigma_r)$ and \mathcal{S}_r are the value function and set of feasible parameters, respectively, of the parametric trajectory optimization problems

$$\begin{aligned} F(y_r, \sigma_r) = \min_{x_r} \quad & \sum_{i \in \mathcal{N}_r} J(x_i) \\ \text{s.t.} \quad & x_i(t) \in D_i(s_{i,0}), \quad \text{for } i \in \mathcal{N}_r, \\ & x_i(y(i, v)) = b_r(v), \quad \text{for } v \in V_r, i \in \mathcal{N}_r, \\ & x_i(y(i, v) + \sigma(i, v)) = e_r(v), \quad \text{for } v \in V_r, i \in \mathcal{N}_r, \\ & x_i(t) - x_j(t) \geq L, \quad \text{for } (i, j) \in \mathcal{C} \cap \mathcal{N}_r, \end{aligned}$$

where we again use subscript r to group variables according to their associated route.

1.2 Decomposition for delay objective

Suppose we use the crossing at the last intersection as performance measure, by defining the objective function as

$$J(x_i) = \inf\{t : x_i(t) \in \mathcal{E}_r(v_r(m_r))\}.$$

We show how to reduce the resulting problem to a scheduling problem, like we did in the single intersection case. It is not clear whether vehicles will always cross intersections at full speed, but we will simply require vehicles to do so from here on. Furthermore, we will again assume that all vehicles share the same geometry. Hence, the occupation time $\sigma \equiv \sigma(i, v)$ is the same for all vehicles and intersections. For this reason, we will write the shorthand $y_r \in \mathcal{S}_r$, because σ_r is no longer a free variable.

Assumption 1.2. *Vehicles must drive at full speed while occupying an intersection.*

Assumption 1.3. *All vehicles have the same length $L_i = L$ and width $W_i = W$.*

As a consequence of Assumption 1.2 and Assumption 1.3, each lower-level trajectory optimization problem for a given route $r \in \mathcal{R}$ decomposes into a sequence of problems, each corresponding to two consecutive intersection along V_r . This means that $y_r \in \mathcal{S}_r$ is equivalent to $y_{(v,w)} \in \mathcal{S}_{(v,w)}$ for each $(v,w) \in E_r$, where $y_{(v,w)}$ denotes the vector of all variables $y(i, v)$ and $y(i, w)$ for all $i \in \mathcal{N}_r$ and $\mathcal{S}_{(v,w)}$ denotes the set of values of $y_{(v,w)}$ for which a feasible trajectory part can be found.

We will first investigate necessary conditions $y_{(v,w)} \in \mathcal{S}_{(v,w)}$.

References