Online control with re-optimization

The model assumes that routes are edge-disjoint and that vehicles drive at full speed over the intersection area. We add vehicles to the network in order of increasing arrival time. After a new arrival happened, we calculate the current optimal set of crossing times relative to the current time. Next, we simulate this schedule forward in time until the next arrival. The re-optimization problem is obtained by generalizing the network MILP to incorporate vehicle positions at a certain time.

Instantaneous acceleration

We only include a crossing time variable $y(i, v)$ in the MILP when vehicle i still needs to cross intersection v. Let $x(i)$ denote the current position of vehicle i, relative to the first node of its route. Let $(u(i), v(i))$ denote the current edge of vehicle i. Let $p(i)$ denote the *edge position*, which is the position on the current edge, so relative to $u(i)$.

For conjunctions, we need to consider three cases:

- both vehicles still need to cross the intersection
- vehicle 2 is occupies the intersection
- both vehicles crossed the intersection

For disjunctions, we need to consider four cases:

- both vehicles still need to cross the intersection
- vehicle 1 still occupies the intersection
- vehicle 2 still occupies the intersection
- both vehicles crossed the intersection

Distance constraints:

- current edge, take into account current edge position
- unvisited edges, use edge length

Capacity constraints:

- not yet crossed
- crossed

Bounded acceleration

This problem is more restrictive, because some decisions are fixed due to the inability of vehicles to stop immediately.