Finite Buffers

Jeroen van Riel

January 2024

1 Model definition

Up to this point, we have not taken into account the fact that lanes between intersection have finite capacity. We need to incorporate this aspect in order to develop a model that could be used for practical applications. Under high traffic loads, lanes with finite buffer capacity can give rise to *blocking* of upstream intersections. Therefore, the traffic controller needs to take into account these additional dynamics.

We start with a very simple extension of the single intersection scheduling model that incorporates additional *locations* on each lane. Instead of only determining the arrival times of vehicles at intersections, the scheduler has to determine the arrival and departure time of vehicles at every location. We assume that it takes constant time Δt for a vehicle to travel to the next location.

For each lane $k \in \{1, \ldots, K\}$, let m_k denote the number of locations, excluding the entrypoint. Let the locations of lane k be identified by increasing numbers $\mathcal{L}(k) = (1, \ldots, m_k)$, where the last one corresponds to the final intersection. In the following variable definitions, we will not explicitly mention the dependency on the lane to keep notation simple. Let y_{ij} denote the time at which vehicle j departs from location $i \in \{0, \ldots, m_k\}$, then $y_{ij} + \Delta t$ is the arrival time of vehicle j at location $i + 1$. To simplify notation, we define $\bar{y}_{ij} = y_{i-1,j} + \Delta t$ to be the arrival time of vehicle j at location i. For every location i and vehicle j, we require

$$\bar{y}_{ij} \leq y_{ij}. \tag{1}$$

For each pair of consecutive vehicles on the same lane k with precedence constraint $j \to l \in \mathcal{C}_k$, we have the inequalities

$$y_{ij} + p \leq \bar{y}_{il}, \tag{2}$$

for every location i along their route. Furthermore, we assume that for every vehicle j, the initial departure time from the entrypoint of vehicle j satisfies

$$r_j \leq y_{0j} \tag{3}$$

in order to model the *arrival time* of the vehicle.

Finally, we need to model the safety constraints involving vehicles from different lanes that cross the intersection. Let j be some vehicle on lane $k(j)$, then let let y_j denote the departure time from the intersection, so we have $y_j = y_{ij}$ with $i = m_{k(j)}$. Similarly, let $\bar{y}_j$ denote the arrival time of j at the intersection, so we have $\bar{y}_j = y_{i-1,j} + \Delta t$ with $i = m_{k(j)}$. From constraints (2), we see that it makes sense to say that vehicle j occupies the intersection during the interval

$$[\bar{y}_j, y_j + p]. \tag{4}$$

Like in the single intersection scheduling problem, we require an additional *switch-over time* s whenever the next vehicle to occupy the intersection comes from a different lane. Therefore, we obtain a similar set of constraints for the intersection. Like before, let

$$\mathcal{D} = \{\{j, l\} : j \in F_{k_1}, l \in F_{k_2}, k_1 \neq k_2\}, \tag{5}$$

denote the set of *conflict* pairs. The additional disjunctive constraints are

$$y_j + p + s \leq \bar{y}_l \quad \text{or} \quad y_l + p + s \leq \bar{y}_j, \tag{6}$$

for all $j, l \in \mathcal{D}$.

Let $\mathcal{J}_k$ denote the set of all vehicles that approach the intersection from lane k. Suppose we want to minimize total delay at the intersection among all vehicles, then the extension of the single intersection scheduling problem can be stated as

$$\text{minimize} \sum_j y_j \tag{7a}$$

$$\text{s.t. } r_j \leq y_{0j} \qquad \text{for all } j, \tag{7b}$$

$$\bar{y}_{ij} \leq y_{ij} \qquad \text{for all } j \in \mathcal{J}_k \text{ and } i \in \mathcal{L}_k, \tag{7c}$$

$$y_{ij} + p \leq \bar{y}_{il} \qquad \text{for all } (j, l) \in \mathcal{C}_k \text{ and } i \in \mathcal{L}_k, \tag{7d}$$

$$\text{either } \begin{cases} y_j + p + s \leq \bar{y}_l \\ y_l + p + s \leq \bar{y}_j \end{cases} \qquad \text{for all } \{j, l\} \in \mathcal{D}, \tag{7e}$$

where we let $k \in \{1, \ldots, K\}$ implicitly to simplify notation.

Definition 1.1. *The problem defined be* (7a) *is called the* single intersection problem with finite buffers.

1.1 Alternative formulation

We obtain a different perspective on the problem when we formulate it in terms of delay at locations instead of arrival and departure times. For a vehicle j, we let

$$d_{0j} = y_{0j} - r_j$$

be the delay at its entrypoint. For the remaining locations, we simply define

$$d_{ij} = y_{ij} - \bar{y}_{ij}.$$

It is not difficult to see that this is completely equivalent, based on the relationship

$$y_{ij} = r_j + i \cdot \Delta t + \sum_{n=0}^{i} d_{ij}. \tag{8}$$

When defining a reinforcement learning agent for generating trajectories, it might be more sensible to generate d instead of y.

2 Solution

We can solve (7a) by formulating it as a MILP using the big-M method and solve it using standard software. To illustrate the model, let us consider a small example.

Example 2.1. Let $\Delta t = 1$, $p = 1$ and $s = 1$ and consider two lanes with $m_1 = m_2 = 5$. Assume we have the following arriving vehicles

$$F_1 = (1, 2), F_2 = (3, 4, 5)$$

with arrival times

$$r = (0, 1, 2, 3, 5).$$

The optimal schedule has objective todo and is illustrated in Figure 1. □

2.1 Reduction to single intersection scheduling problem

Recall the definition of a semi-active schedule. It turns out that there is always a schedule y that is *semi-active* at all locations except the intersection. <- This was my first thought, but it is not correct.

We will show now that the problem above can be reduced to the single intersection scheduling problem that we already discussed. It turns out that there is always a schedule y in which each vehicle always stays as close to the next intersection as possible. -> find a definition for this! This means that we can solve the single intersection problem to obtain y_j and then derive the rest of y_{ij} from a simple recursive calculation.

Proposition 2.1. *The single intersection problem with finite buffers has an optimal schedule that is [insert definition here].*

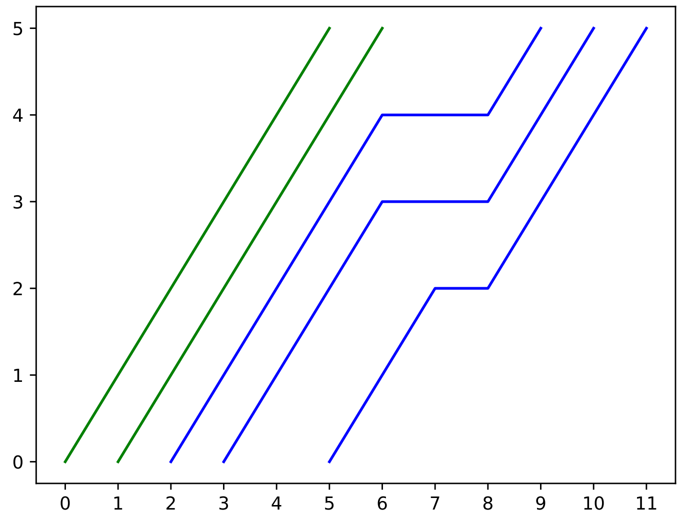

Figure 1: Optimal schedule for Example 2.1. The x-axis represents time and the location numbers are on the y-axis.

Proof. Suppose y is some optimal schedule. We show how to derive from this a schedule y' that is semi-active at all locations other than the intersection without increasing the objective.

We set $y'_j = y_j$ for all j, so the objective does not change. Now for each lane k, we derive the remaining y_{ij} for the vehicles

$$j_1 \to j_2 \to \cdots \to j_{n_k}.$$

on this lane. Let $\mathcal{C}_k$ denote the set of all precedence constraints between vehicles of this lane.

Since the intersection is the last location, a vehicle can always continue immediately after reaching it. Therefore, we do not have to wait at the intersection, so we set $y'_{m(k),j} = \bar{y}'_{m(k),j}$ for all $j_1, \dots, j_{n_k}$, so that we have $y'_{m(k)-1,j} = y'_{m(k),j} - \Delta t = y_j - \Delta t$.

For the first vehicle j_1, we can simply set

$$y'_{ij_1} = y'_{i+1,j_1} - \Delta t,$$

for $i \in \{1, \dots, m(k) - 1\}$. For every pair of consecutive vehicles $j \to l \in \mathcal{C}_k$, we set

$$y'_{il} = \min\{y'_{i+1,l} - \Delta t, y'_{i+1,j} + p - \Delta t\},$$

for $i \in \{1, \dots, m(k)-1\}$

From these recursive equations, all values of y' can be determined. It is not difficult to see that y'_{il} cannot be made smaller without increasing another entry of y', because it needs to satisfy $y'_{il} + \Delta t = \bar{y}'_{i+1,l} \leq y'_{i+1,l}$ and $y'_{il} + p \leq \bar{y}_{il}$. $\square$

3 Extending to network

It is not a surprise that we can reduce the problem back to the variant with infinite buffer space, because we essentially still have infinite buffer space at the entrypoints. The only difference is that the movement is now somewhat more explicitly modelled. As we move on to consider more than one intersections, we will see that the model is really different.

For every lane (x_0, x_1), let $m(x_0, x_1)$ denote the number of locations between both endpoints. Consider vehicle j with route R_j. Let y_{ij} denote the time at which vehicle j departs from location $i = R_j(k)$ for some k, then $y_{ij} + \Delta t$ is the arrival time of vehicle j at location $R_j(k+1)$. To simplify notation, we define

$$\bar{y}_{ij} = y_{R_j(k-1),j} + \Delta t \tag{9}$$

to be the arrival time of vehicle j at location $i = R_j(k)$. For every location i and vehicle j, we require

$$\bar{y}_{ij} \leq y_{ij}. \tag{10}$$

For each pair of consecutive vehicles on the same lane with precedence constraint $j \to l$, we have the inequalities

$$y_{ij} + p \leq \bar{y}_{il}, \tag{11}$$

for every location i along the shared parts of their routes. Furthermore, we assume that for every vehicle j, the initial departure time from the first location $i_0 = R_j(0)$ on the route of vehicle j satisfies

$$y_{i_0,j} \geq r_{i_0,j} \tag{12}$$

in order to model the *release date*.

4 Second order constraints

Our initial formulation is rather unrealistic, because it allows, in some sense, infinite acceleration. In order to model bounded acceleration, we will consider additional constraints on the difference between delay at consecutive locations. More specifically, we require

$$d_{i-1,j} - d_{ij} \leq a,$$

for all vehicles j and locations $i \in \mathcal{L}(k) \setminus \{1\}$. In terms of y, we obtain the usual finite differences

$$y_{ij} - 2y_{i-1,j} + y_{i-2,j} \leq a.$$

often seen in numerical analysis.