# Deliveries, Feasibility and Route Indexation

This document is about the routes abstraction that we use to track vehicle assignments made while analysing and scoring timeslots.

Routes are used to

- track the approximate location of a vehicle
- decide whether a potential timeslot allocation is feasible
- show the order in which deliveries are to be made

A route is currently modelled as $t \longrightarrow (v, d)$, a time ordered map of a vehicle/delivery pairing. The structure allows us to filter out deliveries that are located in a given timeslot in log time, and to provide a natural ordering of deliveries. But it obscures the fact that the model has little to say about the time that a delivery is to be made, and is potentially awkward when deciding where to insert a new delivery.

# Feasibility Conditions

We have a vehicle $v$ that has been assigned to the delivery route implied by $\{(d_i, \tau_i)\}_{1 \leq i \leq n}$, indexed in delivery order. Timeslots $\tau$ do not overlap, and multiple deliveries can occupy the same timeslot. Package delivery times are denoted $t(d)$, delivery service times are denoted $s(d)$. Transit delivery times from $d_1$ to $d_2$ are denoted $t(d_1, d_2)$ and

Facts

- $\tau \equiv [t_b, t_e)$, we write $b(\tau) = t_b,\ e(\tau) = t_e$.
- $\tau_1 < \tau_2 \Rightarrow b(\tau_2) \geq e(\tau_1)$ and $\tau_1 = \tau_2 \Rightarrow b(\tau_2) = b(\tau_1), e(\tau_2) = e(\tau_1)$
- $\tau_1 \leq \cdots \leq \tau_n$

Feasibility implies

- $b(\tau_0) + t(d_0, d_1) \leq e(\tau_1)$
- $\max(b(\tau_0) + t(d_0, d_1), b(\tau_1)) + s(d_1) + t(d_1, d_2) \leq e(\tau_2)$

Introduce the *slack variable* $\sigma$:

- $\sigma_1 = e(\tau_1) - b(\tau_0) - t(d_0, d_1)$

- $\sigma_2 = e(\tau_2) - \max(e(\tau_1) - \sigma_1, b(\tau_1)) - s(d_1) - t(d_1, d_2)$

and rewrite (apologies for blow-by-blow, but we are working this out as we correct handwritten mistakes)

- $b(\tau_0) + t(d_0, d_1) + \sigma_1 = e(\tau_1)$
- $\max(e(\tau_1) - \sigma_1, b(\tau_1)) + s(d_1) + t(d_1, d_2) + \sigma_2 = e(\tau_2)$
- $\max(e(\tau_2) - \sigma_2, b(\tau_2)) + s(d_2) + t(d_2, d_3) + \sigma_3 = e(\tau_3)$

generally …

- $\max(e(\tau_i) - \sigma_i, b(\tau_i)) + s(d_i) + t(d_i, d_{i+1}) + \sigma_{i+1} = e(\tau_{i+1})$

and if we want a slack calculation

$$\sigma_{i+1} = e(\tau_{i+1}) - \max(e(\tau_i) - \sigma_i, b(\tau_i)) + s(d_i) + t(d_i, d_{i+1})$$

# How to use Slackness

We want to insert a new delivery $d_{i*}$ in between $d_i$ and $d_{i+1}$. We need to check that both conditions are satisfied:

- $\max(e(\tau_i) - \sigma_i, b(\tau_i)) + s(d_i) + t(d_i, d_{i*}) + \sigma_{i*} = e(\tau_{i*})$,
- $\max(e(\tau_{i*}) - \sigma_{i*}, b(\tau_{i*})) + s(d_{i*}) + t(d_{i*}, d_{i+1}) + \sigma_{i+1} = e(\tau_{i+1})$

Implications

- we calculate and save one $\sigma$ per timeslot: $\sigma = \min_{\tau_j = \tau} \{\sigma_j\}$
- when inserting, we only need to update $\sigma_j$ within a timeslot. Any journey that crosses a timeslot boundary "resets" the slack calculation.