Spring 2025

06 Transit Network III: Operations

CIVIL-324 Urban public transport systems

École polytechnique fédérale de Lausanne

Overview

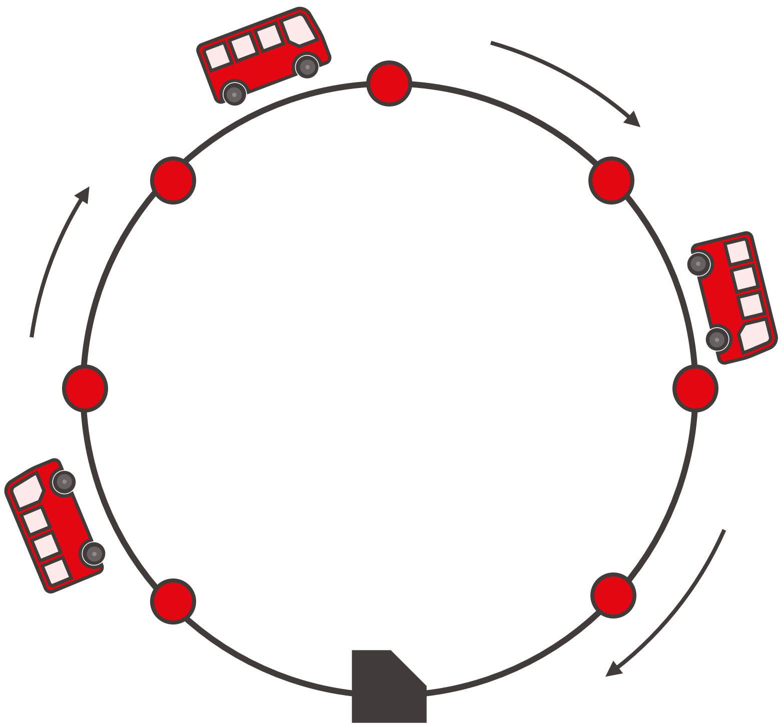
- Transit network design problems discussed in this course
  - Stop location and line planning
    - where to put transit stations and how to connect them into transit lines?
  - Scheduling and pricing
    - how to design the timetable and coordinate multiple transit lines?
    - how to price transit trips under different operational objectives?
  - **Operations**
    - how to design the vehicle schedules given the transit time table?
    - how to manage the potential delay during the operations?

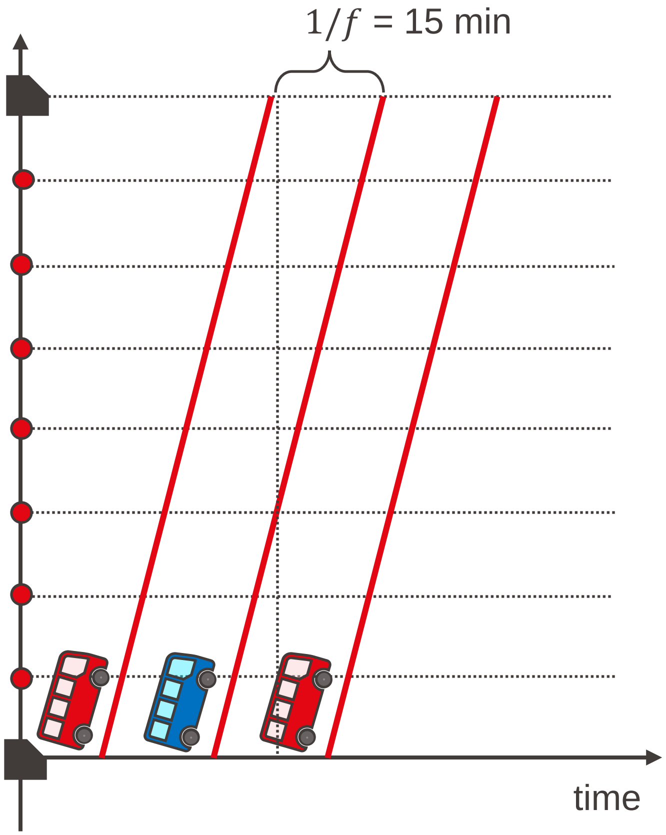
Overview

- Operations
  - How to design the vehicle schedules given the transit time table?
    - min fleet size
    - vehicle scheduling: heuristics and optimization method
  - How to manage the potential delay during the operations?
    - delay management

Vehicle scheduling

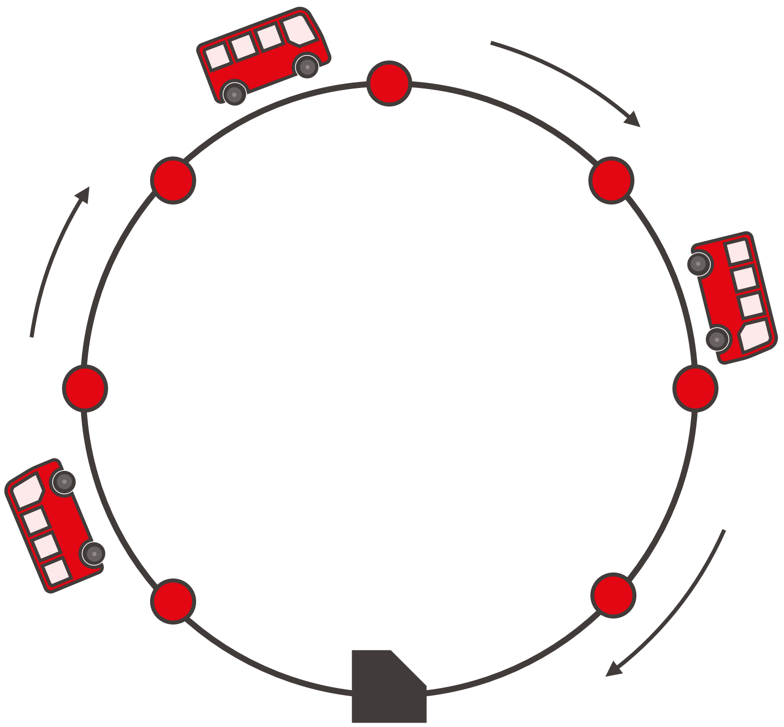
- Single transit line with single terminal

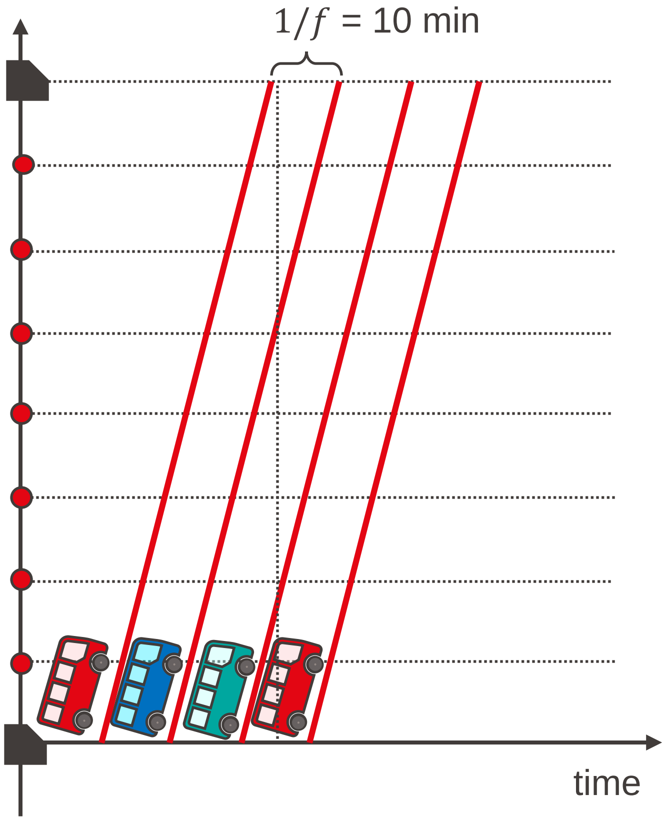




Vehicle scheduling

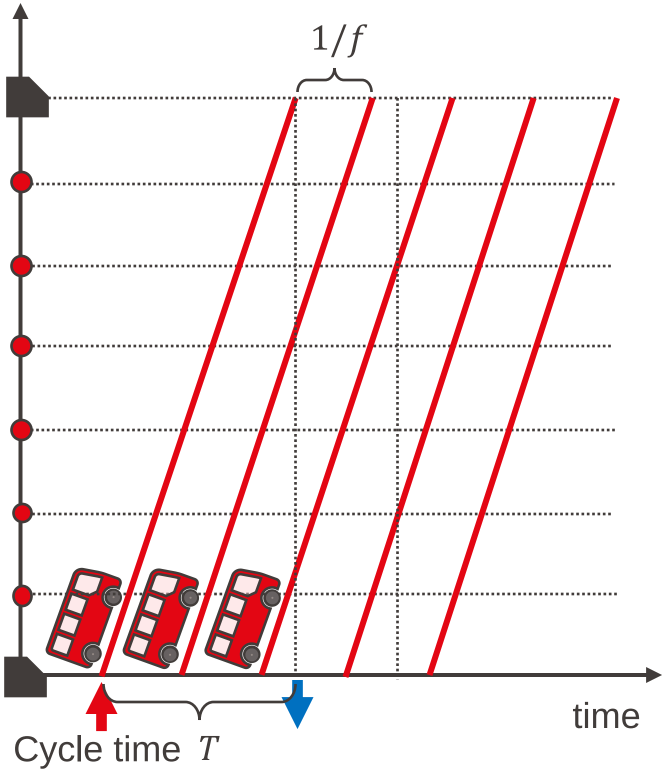
- Single transit line with single terminal

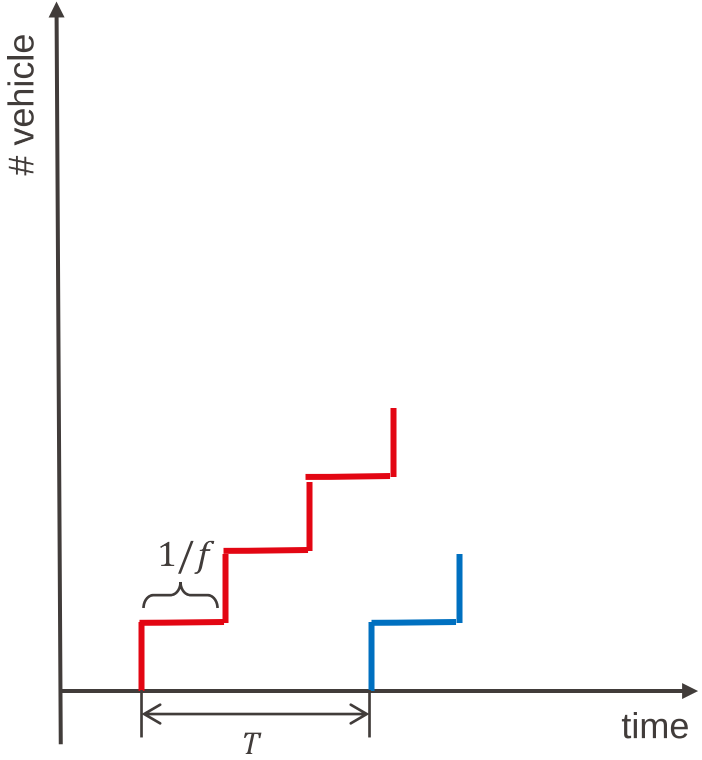




Vehicle scheduling

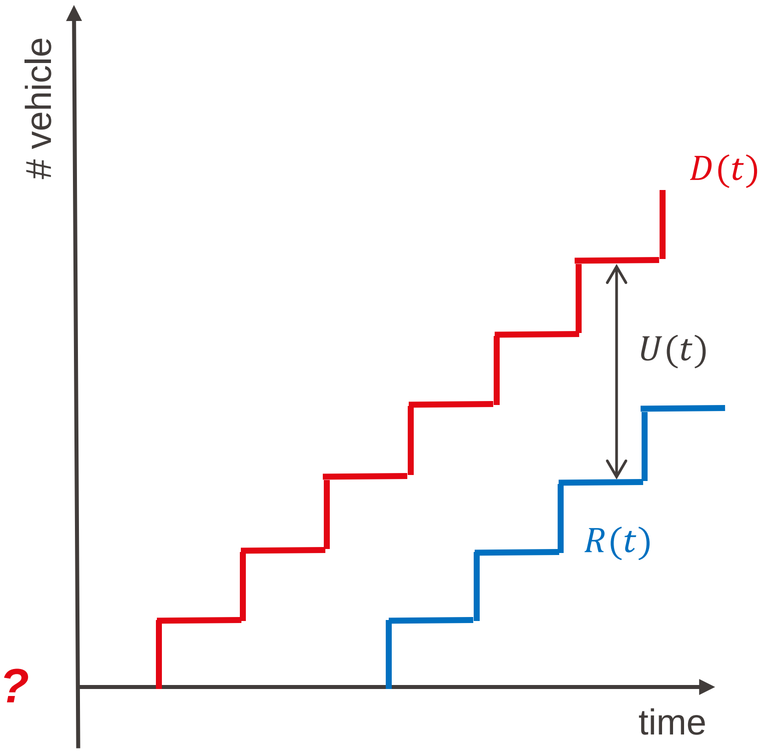
- Single transit line with single terminal
  - How many vehicles are needed to service transit line given its timetable?





Vehicle scheduling

- Single transit line with single terminal
  - # departures $D(t)$
  - # returns $R(t)$
  - # in-service $U(t) = D(t) - R(t)$



Min fleet size

$M = \max U(t)$ **= 3**

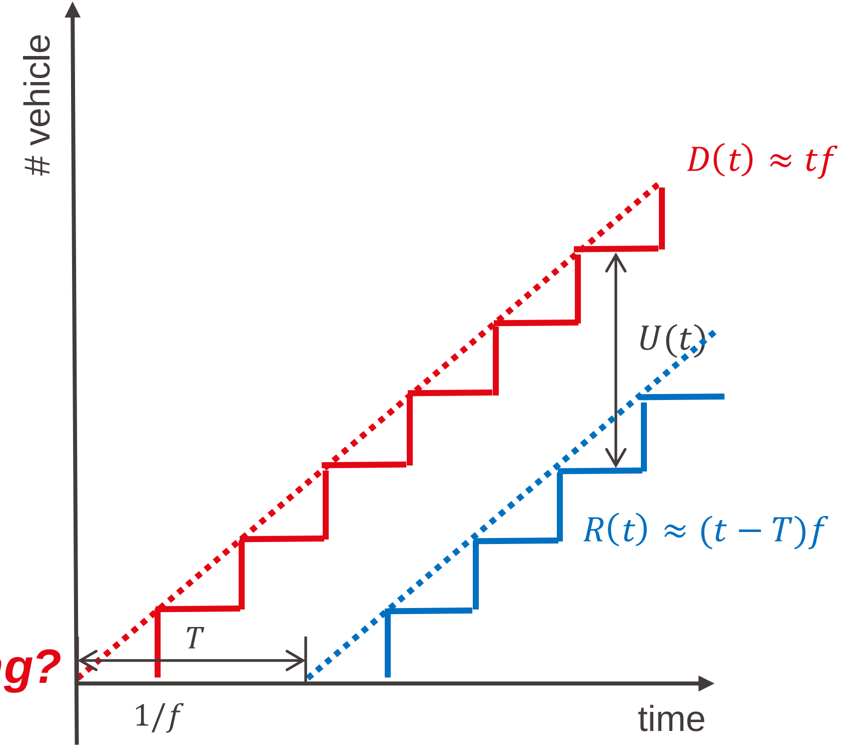
- ***Q: How to approximate $U(t)$ and M?***

Vehicle scheduling

- Single transit line with single terminal
  - # departures $D(t)$
  - # returns $R(t)$
  - # in-service $U(t) = D(t) - R(t)$
  - Frequency f
  - Cycle time T

Min fleet size

$$M = \max U(t) = \max\{D(t) - R(t)\}$$
$$= \max\{tf - (t-T)f\}$$
$$= Tf$$



- ***Q: What if frequency is time-varying?***

Vehicle scheduling

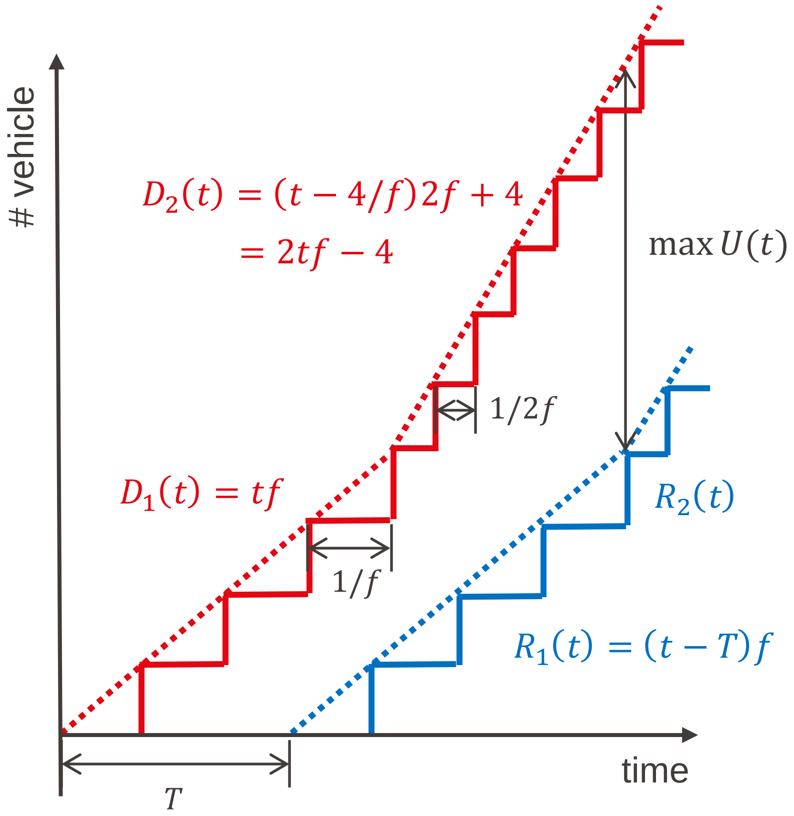
- Single transit line with single terminal
  - \# departures $D(t)$
  - \# returns $R(t)$
  - \# in-service $U(t) = D(t) - R(t)$
  - Frequency f
  - Cycle time T

Min fleet size

$$M = \max U(t)$$

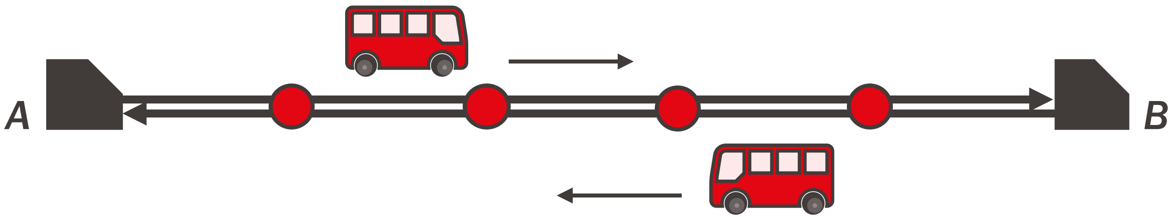
$$= \max_{i \geq j} \{D_i(t) - R_j(t)\}$$

$$= 2Tf$$

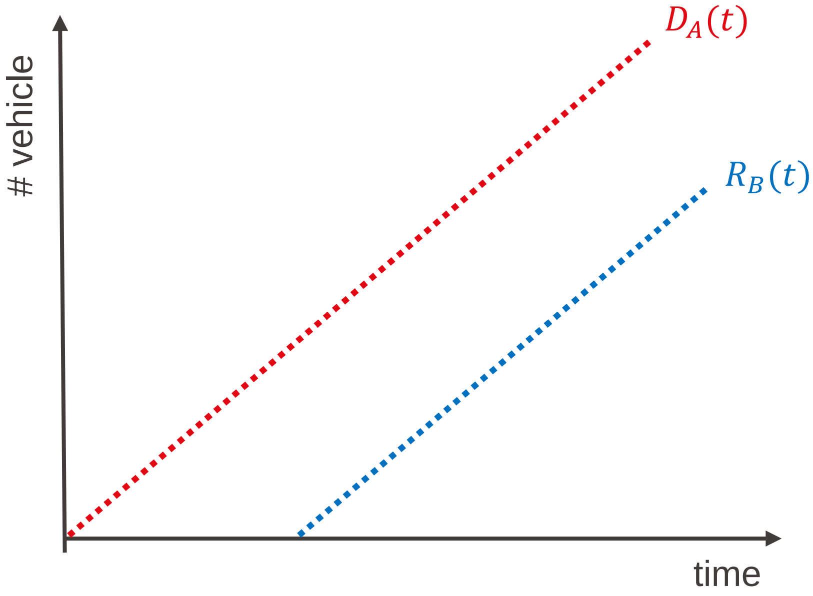


Vehicle scheduling

- Single transit line with two terminals

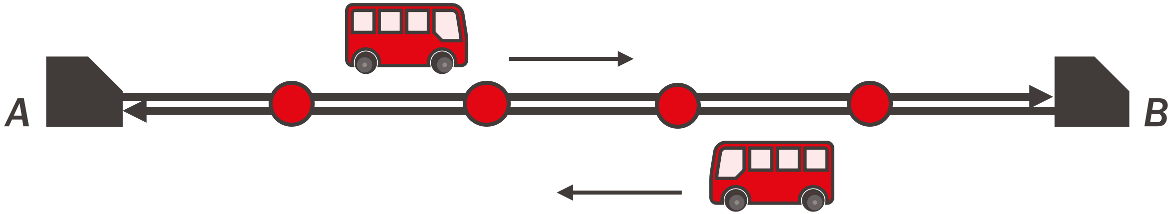


- # departures from A $D_A(t)$
- # returns to B $R_B(t)$

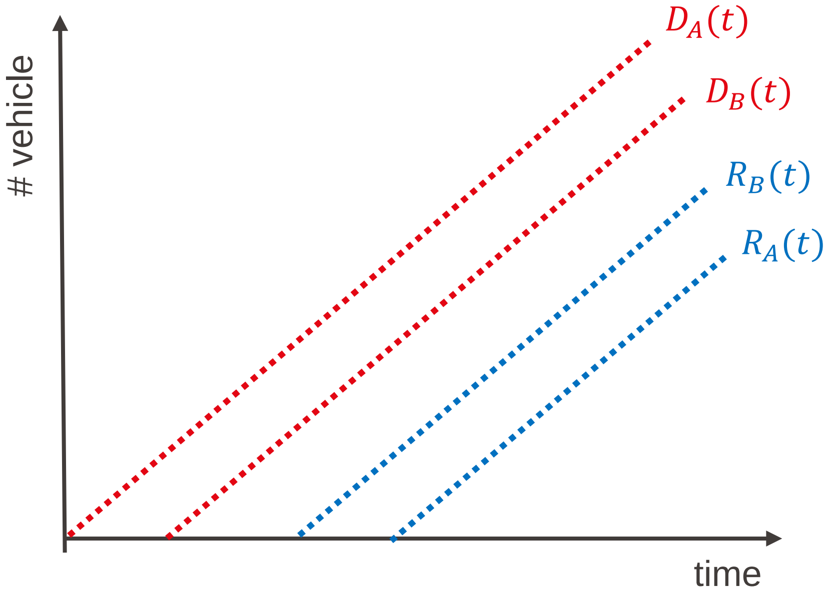


Vehicle scheduling

- Single transit line with two terminals



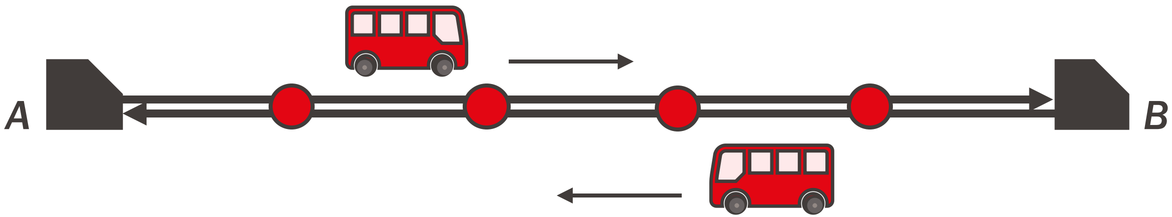
- # departures from A $D_{\mathrm{A}}(t)$
- # returns to B $R_B(t)$
- # departures from B $D_B(t)$
- # returns to A $R_A(t)$



- ***Q: How to compute # in-service?***

Vehicle scheduling

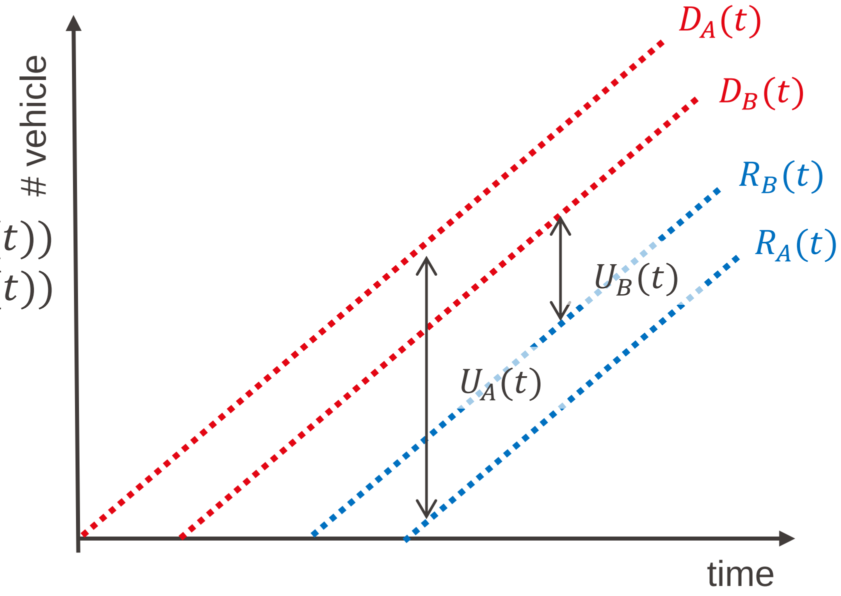
- Single transit line with two terminals



- # departures from A/B $D_{\mathrm{A}}(t)/D_B(t)$
- # returns to A/B $R_A(t)/R_B(t)$
- # in-service

$$\begin{aligned} U(t) &= (D_A(t) - R_B(t)) + (D_B(t) - R_A(t)) \\ &= (D_A(t) - R_A(t)) + (D_B(t) - R_B(t)) \\ &= U_A(t) + U_B(t) \end{aligned}$$

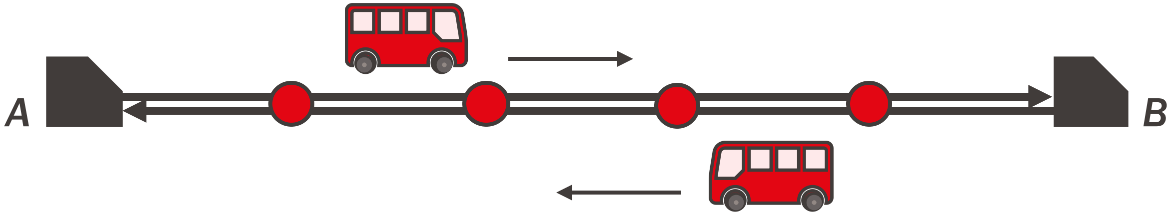
- Min fleet $M = \max U(t)$



- ***Q: What if vehicles returning to B directly go back to A?***

Vehicle scheduling

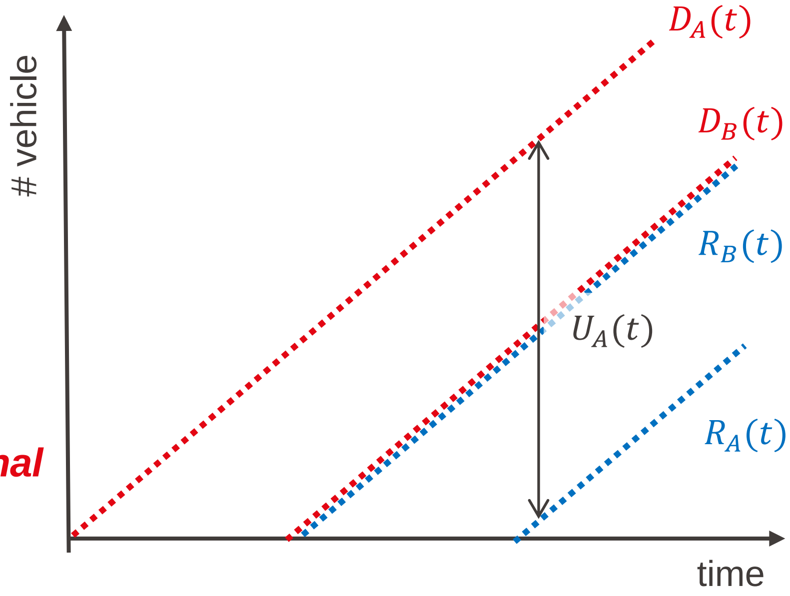
- Single transit line with two terminals



- # departures from A/B $D_A(t)/D_B(t))$
- # returns to A/B $R_A(t)/R_B(t)$
- # in-service

$$U(t) = U_A(t) + U_B(t)$$
$$= (D_A(t) - R_A(t)) + \mathbf{0}$$

- Min fleet $M = \max U(t)$
  - ***reduce to the case of single terminal***



Vehicle scheduling

- Multiple transit line with single terminal (Hub-spoke network)
  - # departures of line l $D_l(t)$
  - # returns line l $R_l(t)$
  - # in-service

$$U(t) = \sum_l D_l(t) - \sum_l R_l(t) = \sum_l U_l(t)$$

  - Min fleet size

$$M = \max U(t) = \max \sum_l U_l(t)$$

$$\leq \sum_l \max U_l(t)$$

sum of single-line fleets

Vehicle scheduling

- Multiple transit line with single terminal (Hub-spoke network)
  - # departures of line l $D_l(t)$
  - # returns line l $R_l\ (t)$
  - # in-service

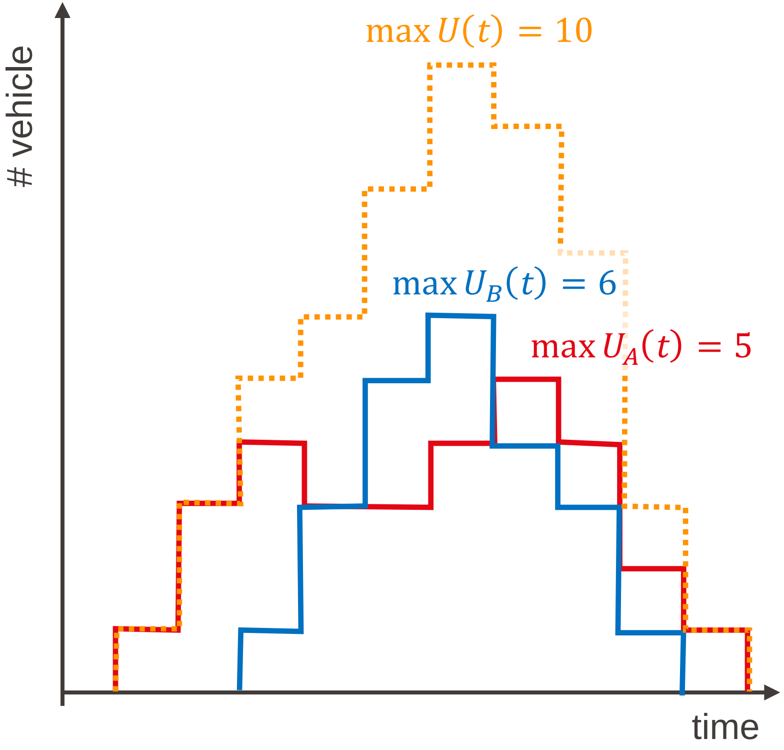
$$U(t) = \sum_l D_l(t) - \sum_l R_l(t) = \sum_l U_l(t)$$

  - Min fleet size

$$M = \max U(t) = \max \sum_l U_l\ (t)$$

$$\leq \sum_l \max U_l(t)$$

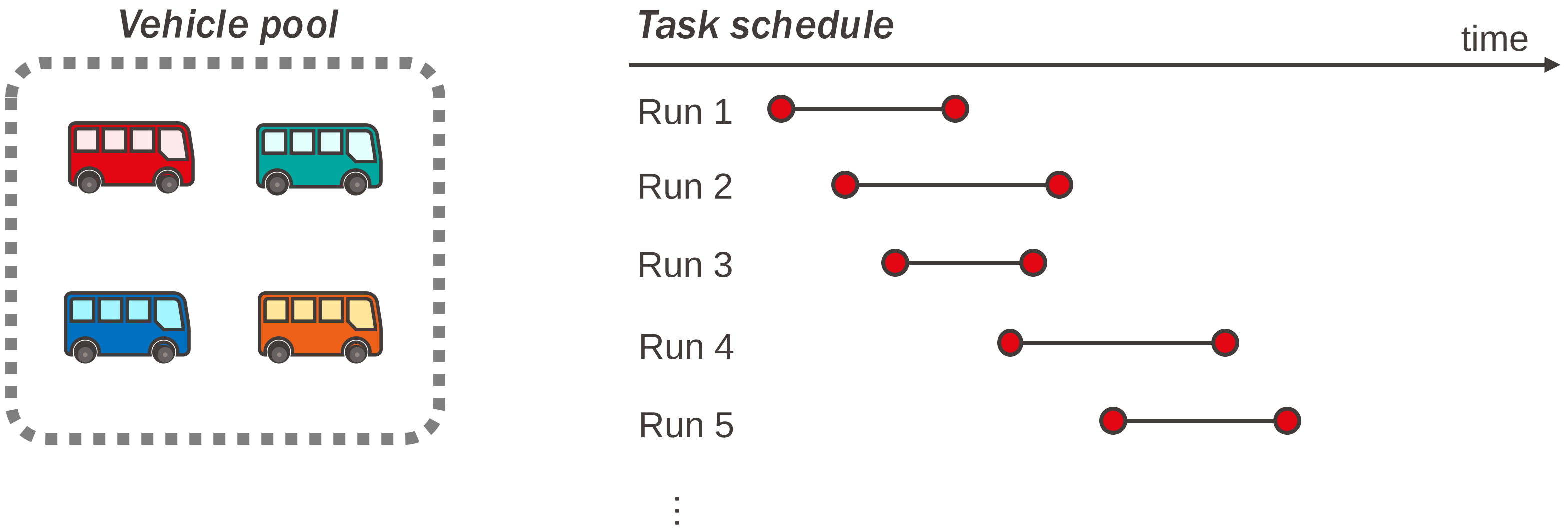
sum of single-line fleets



Questions?

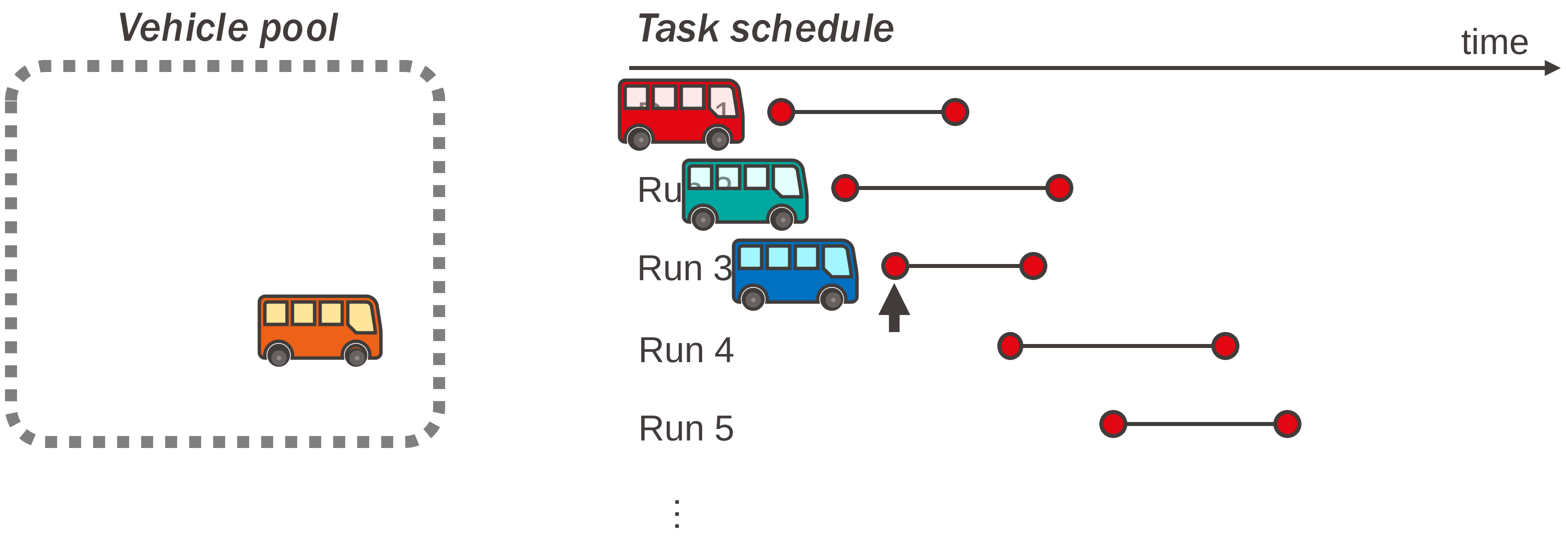
Vehicle scheduling

- Heuristic I: Last-in-first-out (LIFO)
  - Assign each run to the last idle vehicle



Vehicle scheduling

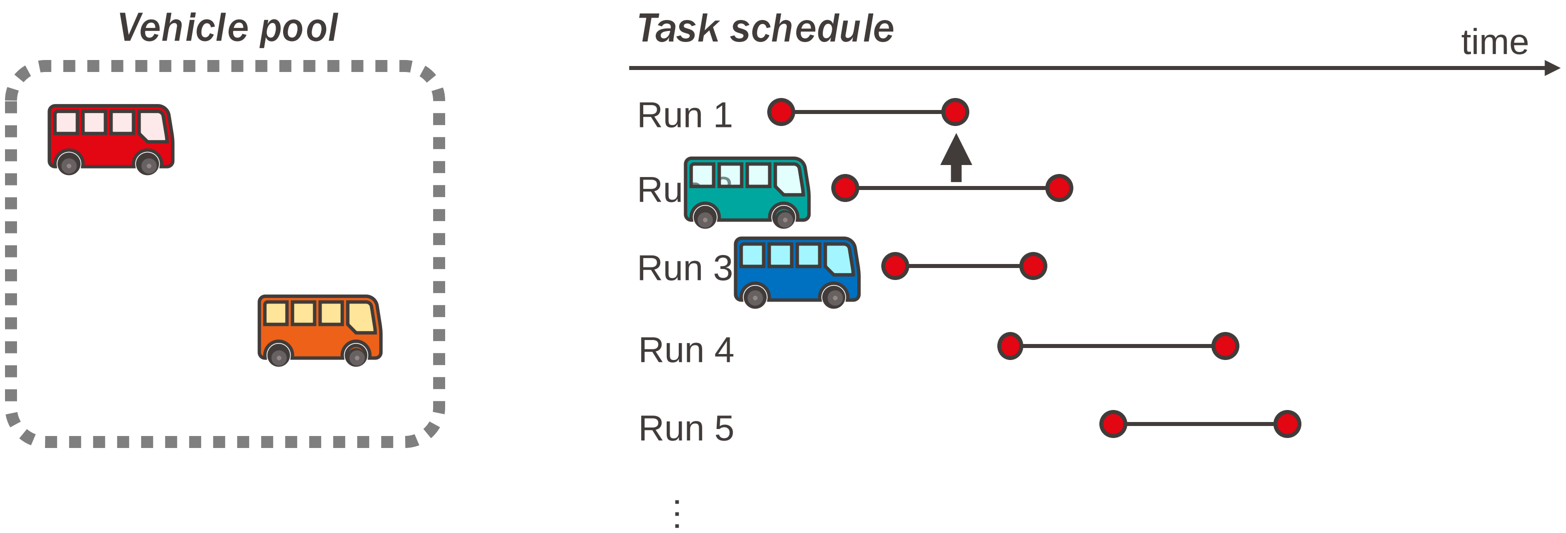
- Heuristic I: Last-in-first-out (LIFO)
  - Assign each run to the last idle vehicle



- ***Q: What is the next vehicle movement?***

Vehicle scheduling

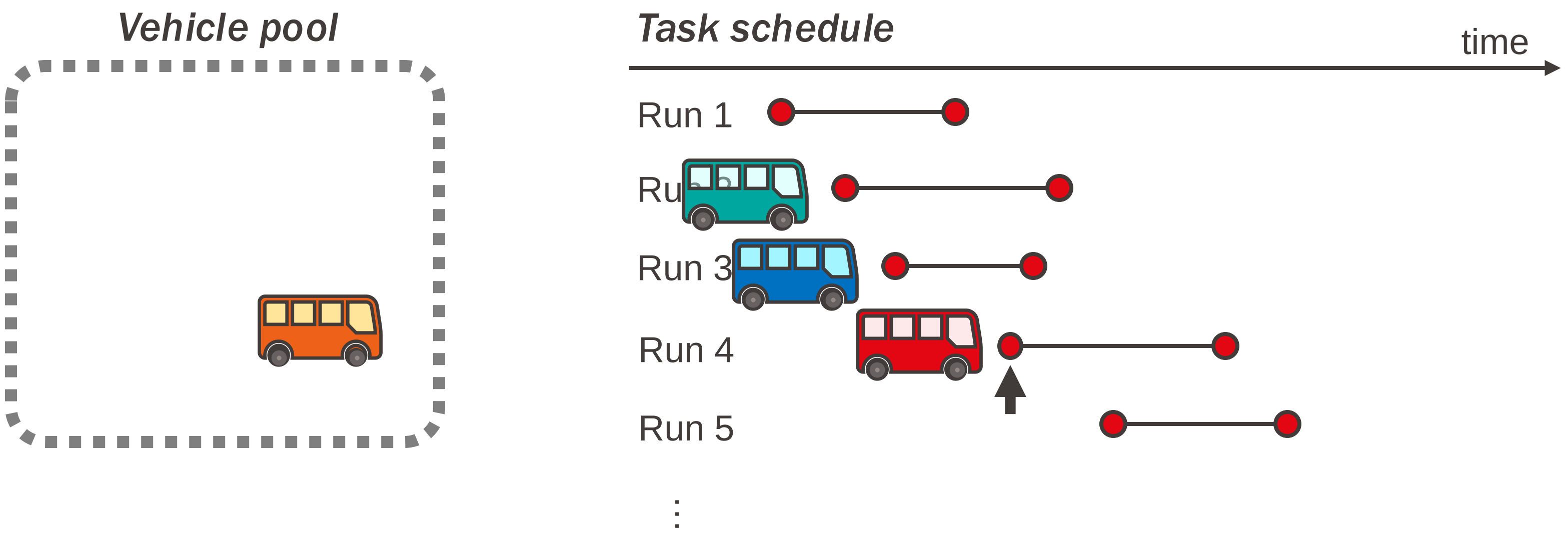
- Heuristic I: Last-in-first-out (LIFO)
  - Assign each run to the last idle vehicle



- ***Q: How to assign the next vehicle based on LIFO?***

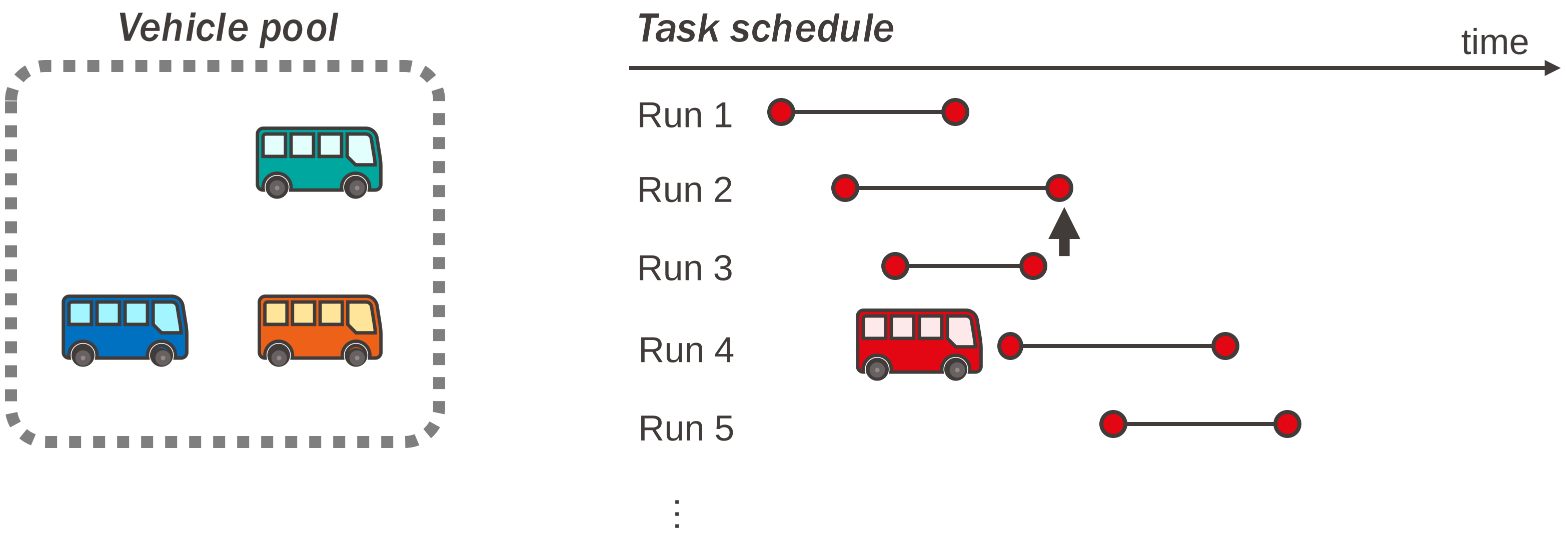
Vehicle scheduling

- Heuristic I: Last-in-first-out (LIFO)
  - Assign each run to the last idle vehicle



Vehicle scheduling

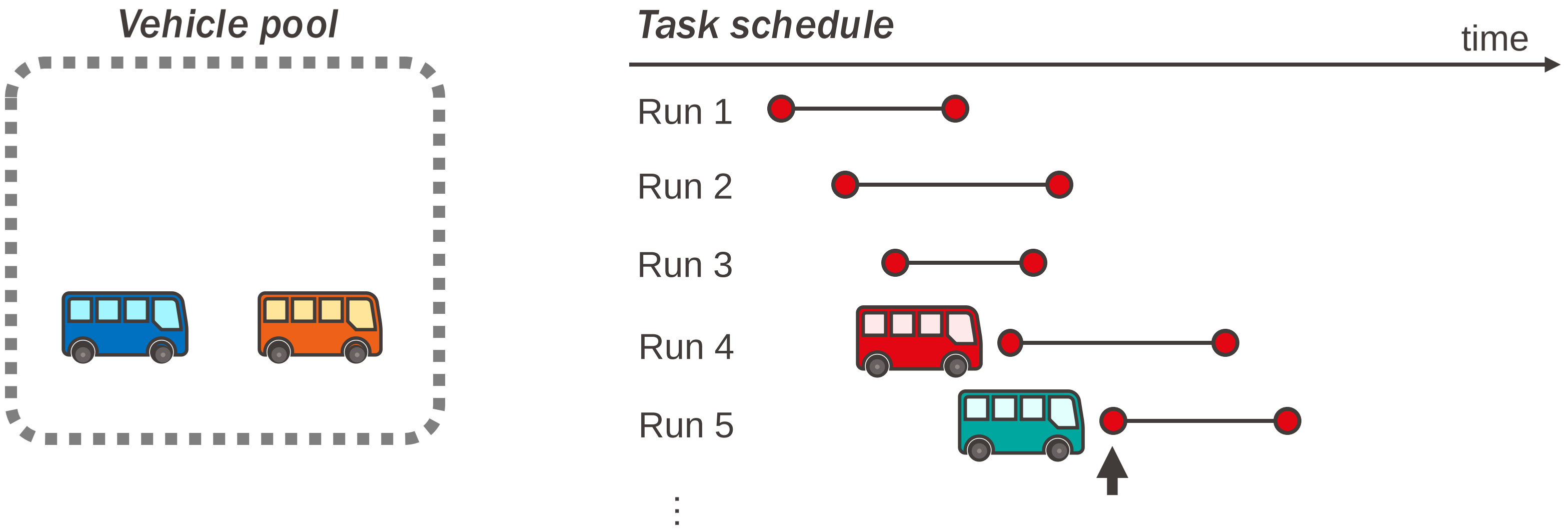
- Heuristic I: Last-in-first-out (LIFO)
  - Assign each run to the last idle vehicle



- ***Q: How to assign the next vehicle based on LIFO?***

Vehicle scheduling

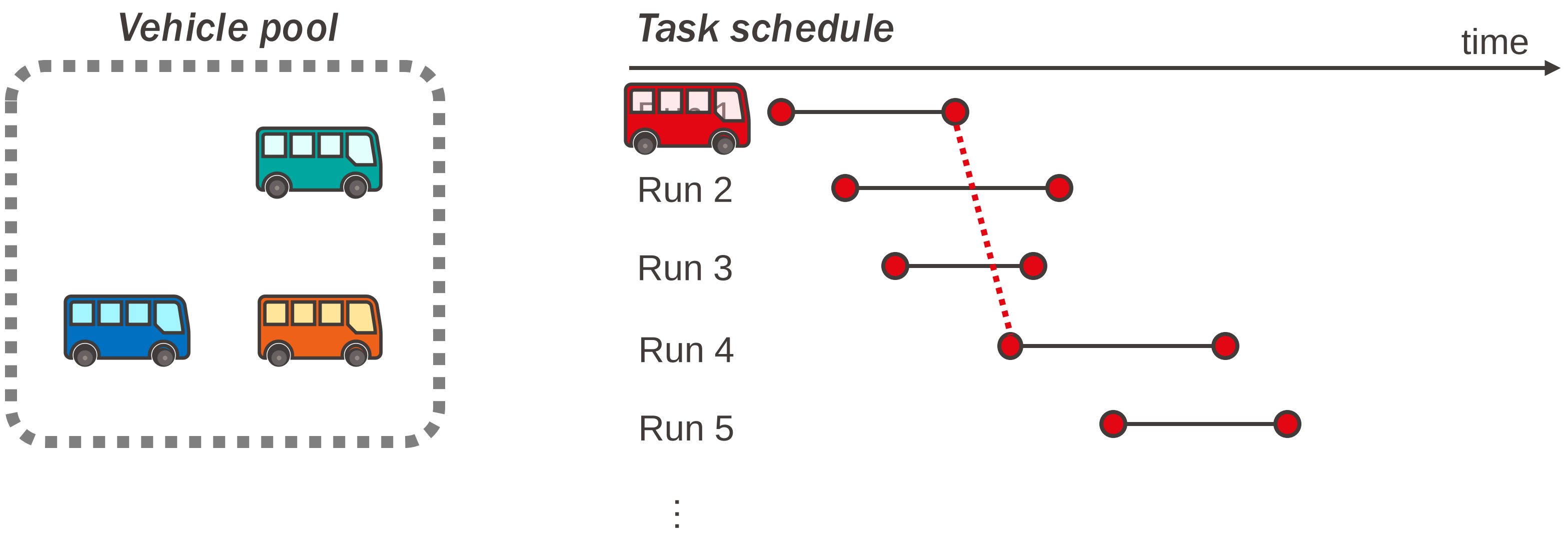
- Heuristic I: Last-in-first-out (LIFO)
  - Assign each run to the last idle vehicle



- ***Q: How many vehicles are used to fulfill the five tasks?***

Vehicle scheduling

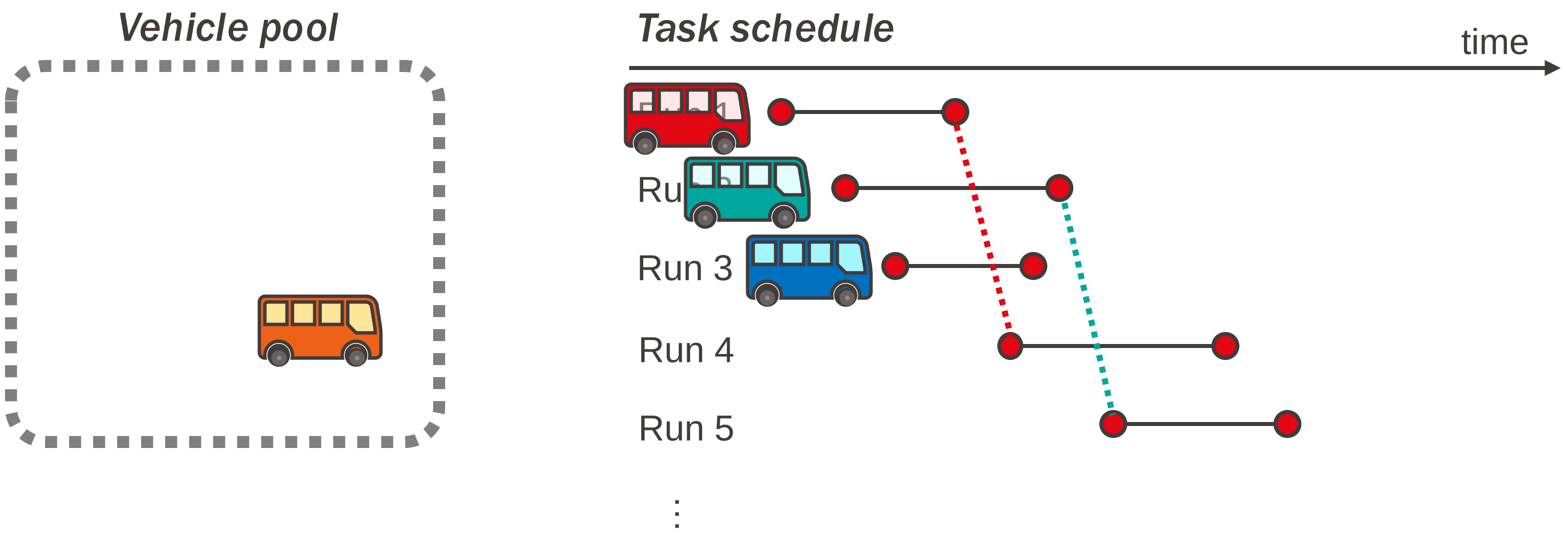
- Heuristic II: Greedy
  - Sequentially assign each vehicle’s schedule
  - Min connection between tasks



- ***Q: How to assign other vehicles based on the greedy principle?***

Vehicle scheduling

- Heuristic II: Greedy
  - Sequentially assign each vehicle's schedule
  - Min connection between tasks



- *Q: How many vehicles are used to fulfill the five tasks?*

Vehicle scheduling

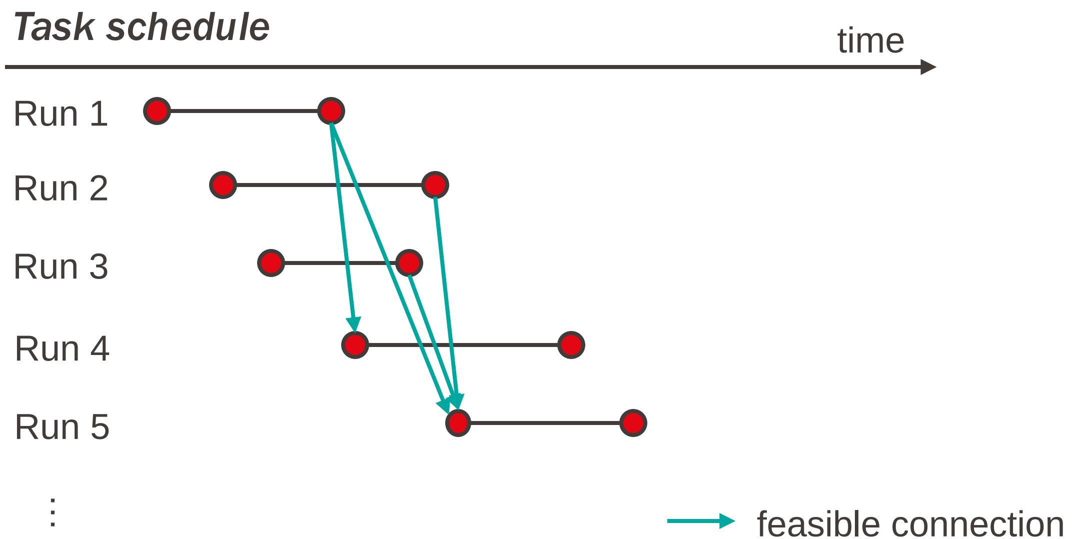
- Heuristic I: Last-in-first-out (LIFO)
  - Assign each run to the last idle vehicle

- Heuristic II: Greedy
  - Sequentially assign each vehicle’s schedule
  - Min connection between tasks

- Both end up using the min fleet we derived before

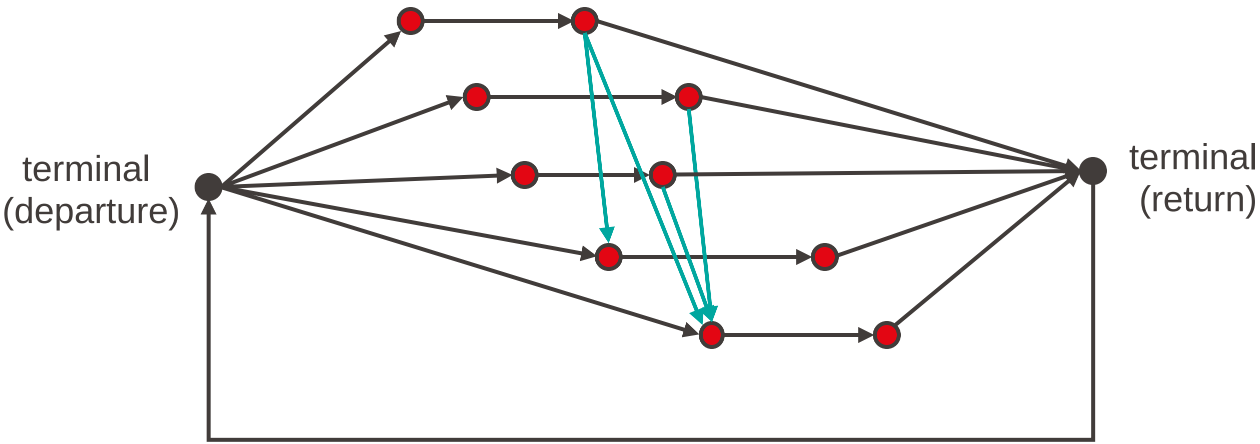
Vehicle scheduling

- Optimization method
  - Construct a circulation network
    - Step 1: link all feasible connections



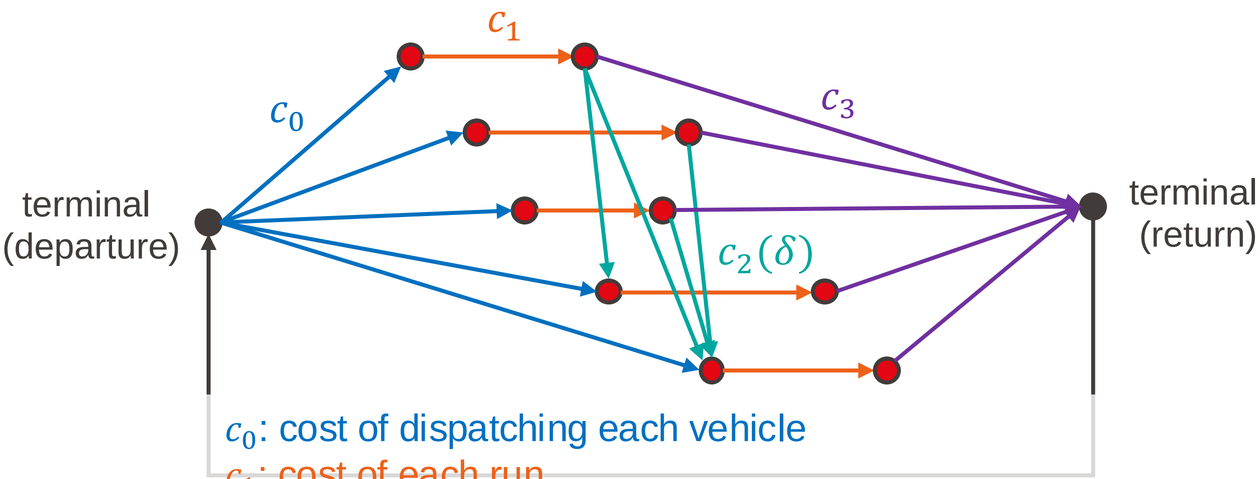
Vehicle scheduling

- Optimization method
  - Construct a circulation network
    - Step 1: link all feasible connections
    - Step 2: link tasks to terminals and connect two terminals



Vehicle scheduling

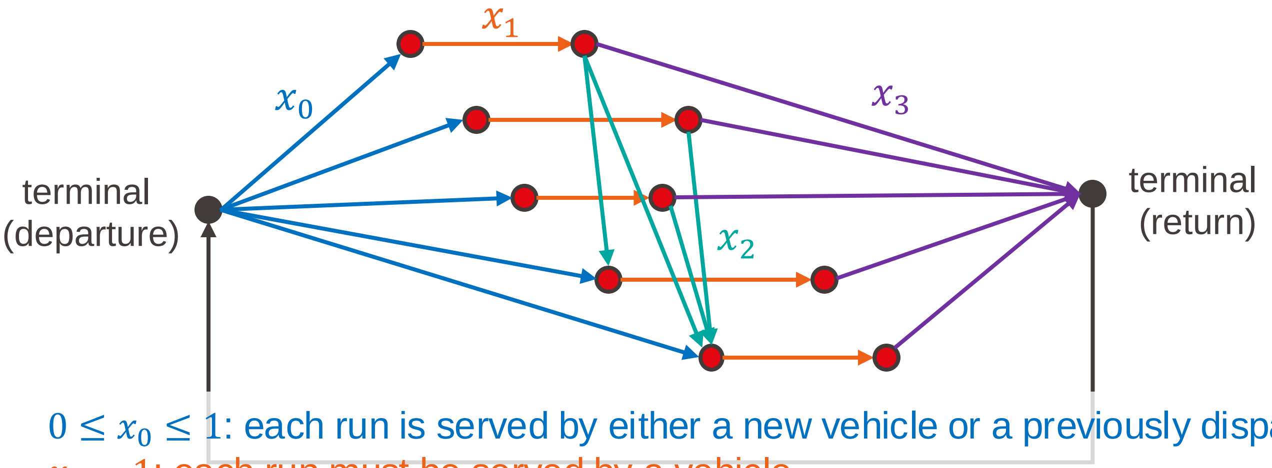
- Optimization method
  - Construct a circulation network
    - Step 1: link all feasible connections
    - Step 2: link tasks to terminals and connect two terminals
    - Step 3: assign link costs



c_0: cost of dispatching each vehicle
c_1: cost of each run
$c_2(\delta)$: cost of each connection with layover time δ
c_3: zero cost for returning vehicles

Vehicle scheduling

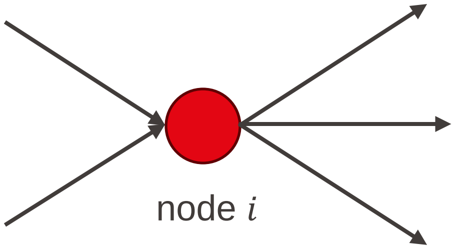
- Optimization method
  - Construct a circulation network
    - Step 1: link all feasible connections
    - Step 2: link tasks to terminals and connect two terminals
    - Step 3: assign link costs
    - Step 4: define link and node constraints



$0 \leq x_0 \leq 1$: each run is served by either a new vehicle or a previously dispatched vehicle
$x_1 = 1$: each run must be served by a vehicle
$0 \leq x_2 \leq 1$: each feasible connection is taken by at most one vehicle
$0 \leq x_3 \leq 1$: at most one vehicle returns after each run

Vehicle scheduling

- Optimization method
  - Construct a circulation network
    - Step 1: link all feasible connections
    - Step 2: link tasks to terminals and connect two terminals
    - Step 3: assign link costs
    - Step 4: define link and node constraints



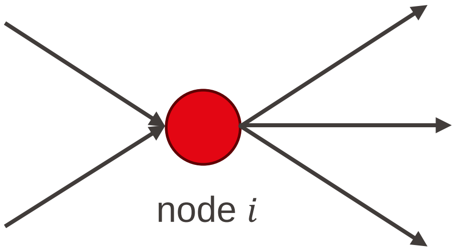
$$\sum_{a \in A_i^-} x_a = \sum_{a \in A_i^+} x_a$$

- A_i^-: set of links ending at node i
- A_i^+: set of links starting from node i

- ***Q: What does this constraint mean?***

Vehicle scheduling

- Optimization method
  - Construct a circulation network
    - Step 1: link all feasible connections
    - Step 2: link tasks to terminals and connect two terminals
    - Step 3: assign link costs
    - Step 4: define link and node constraints



total inflow $\sum_{a \in A_i^-} x_a = \sum_{a \in A_i^+} x_a$ total outflow

- A_i^-: set of links ending at node i
- A_i^+: set of links starting from node i

- ***Q: How to express constraints in a more compact way?***

Vehicle scheduling

- Optimization method
  - Construct a circulation network
    - Step 1: link all feasible connections
    - Step 2: link tasks to terminals and connect two terminals
    - Step 3: assign link costs
    - Step 4: define link and node constraints

$$Mx = 0$$

$$L \leq x \leq U$$

- M: node-link incidence matrix

$$M_{ia} = \begin{cases} 1, & \text{if link } a \text{ starts from node } i \\ -1, & \text{if link } a \text{ ends at node } i \\ 0, & \text{otherwise} \end{cases}$$

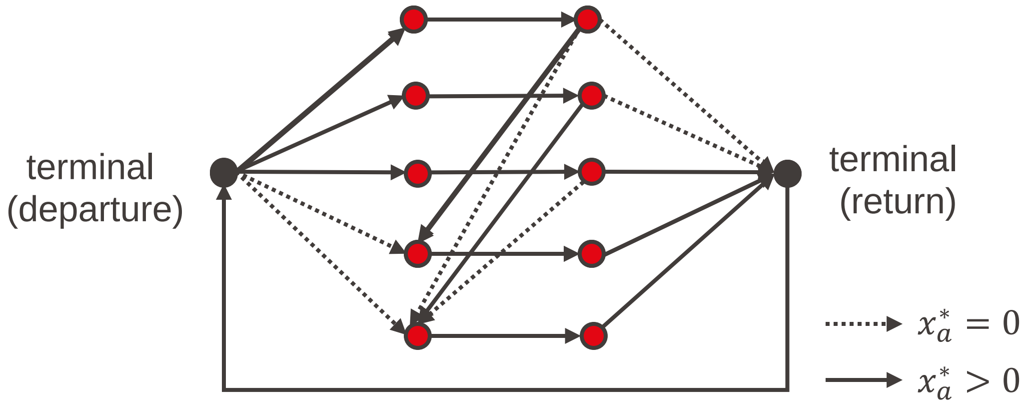
- L, U: lower and upper bounds of link flows

- ***Q: How to solve vehicle schedules using the circulation network?***

Vehicle scheduling

- Optimization method
  - Construct a circulation network
  - Solve the circulation problem
    - min total cost, including vehicle dispatches and connection layovers
    - subject to schedule feasibility constraint

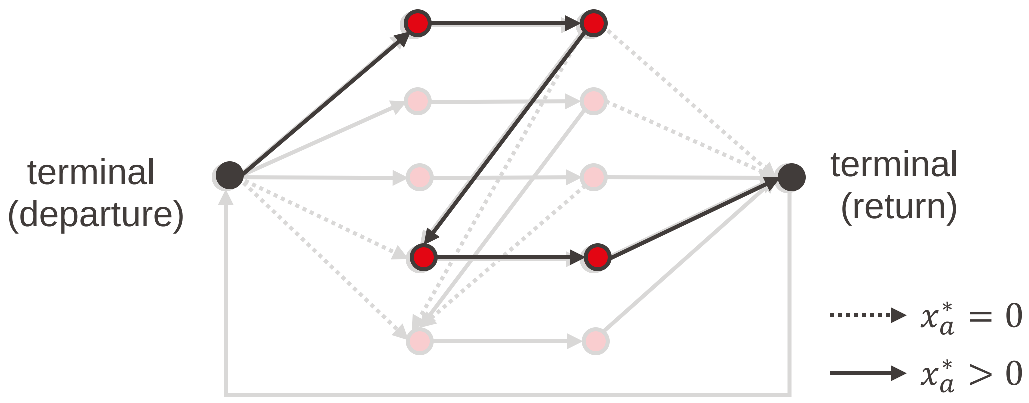
$$\min_x c^T x$$
$$s.t.\ Mx = 0$$
$$L \le x \le U$$



- ***Q: How to interpret the optimal solution x^*?***

Vehicle scheduling

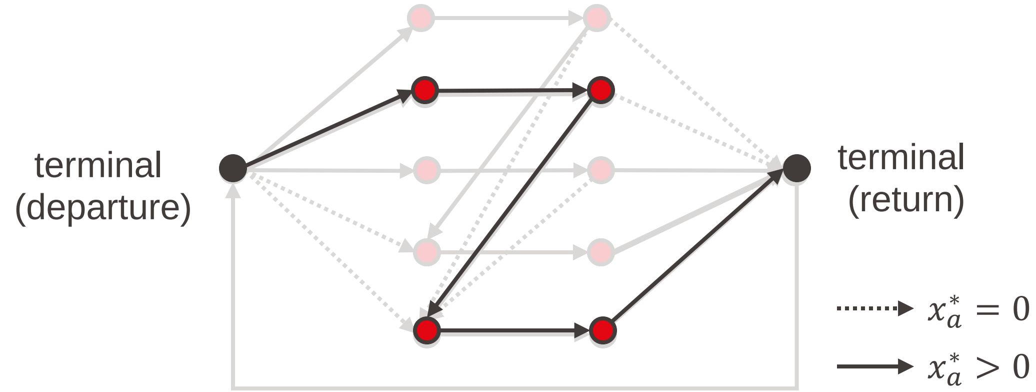
- Optimization method
  - Construct a circulation network
  - Solve the circulation problem
    - min total cost, including vehicle dispatches and connection layovers
    - subject to schedule feasibility constraint



- Vehicle 1 serve 1st and 4th runs, then return

Vehicle scheduling

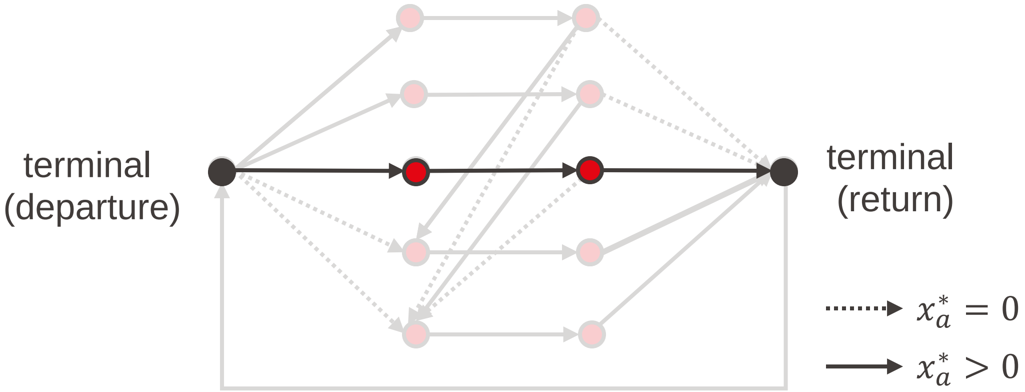
- Optimization method
  - Construct a circulation network
  - Solve the circulation problem
    - min total cost, including vehicle dispatches and connection layovers
    - subject to schedule feasibility constraint



- Vehicle 1 serve 1st and 4th runs, then return
- Vehicle 2 serve 2nd and 5th runs, then return

Vehicle scheduling

- Optimization method
  - Construct a circulation network
  - Solve the circulation problem
    - min total cost, including vehicle dispatches and connection layovers
    - subject to schedule feasibility constraint



- Vehicle 1 serve 1st and 4th runs, then return
- Vehicle 2 serve 2nd and 5th runs, then return
- Vehicle 3 serve 3rd run, then return

Questions?

Delay management

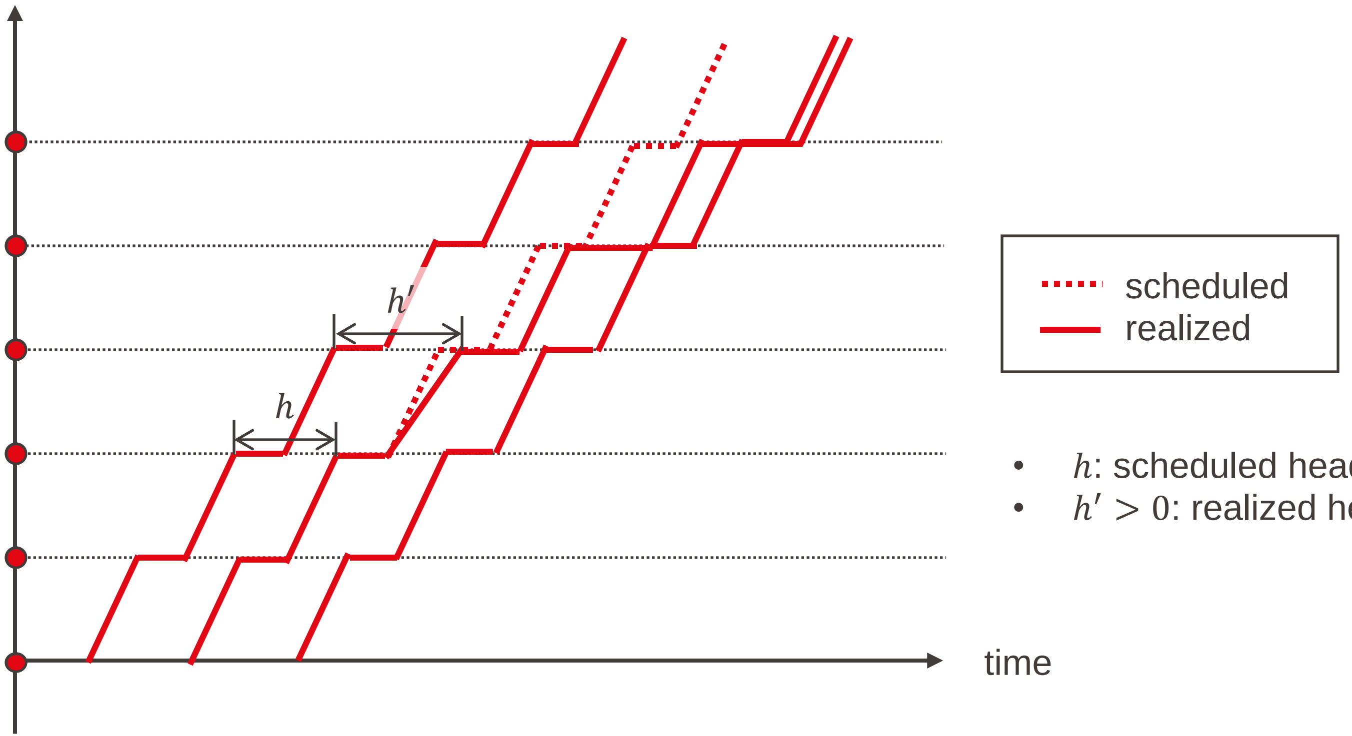
- "Bus bunching" phenomenon
  - Multiple buses arrive at the stop at the same time
  - How does it happen?
    - when a bus runs late, its headway increases.
    - more travelers arrive at the next stop, causing longer stops.
    - the next bus has shorter headway and shorter stops

"Once a bus gets behind schedule, it's nearly impossible to get back on track."

--- Gayah and Guler

Delay management

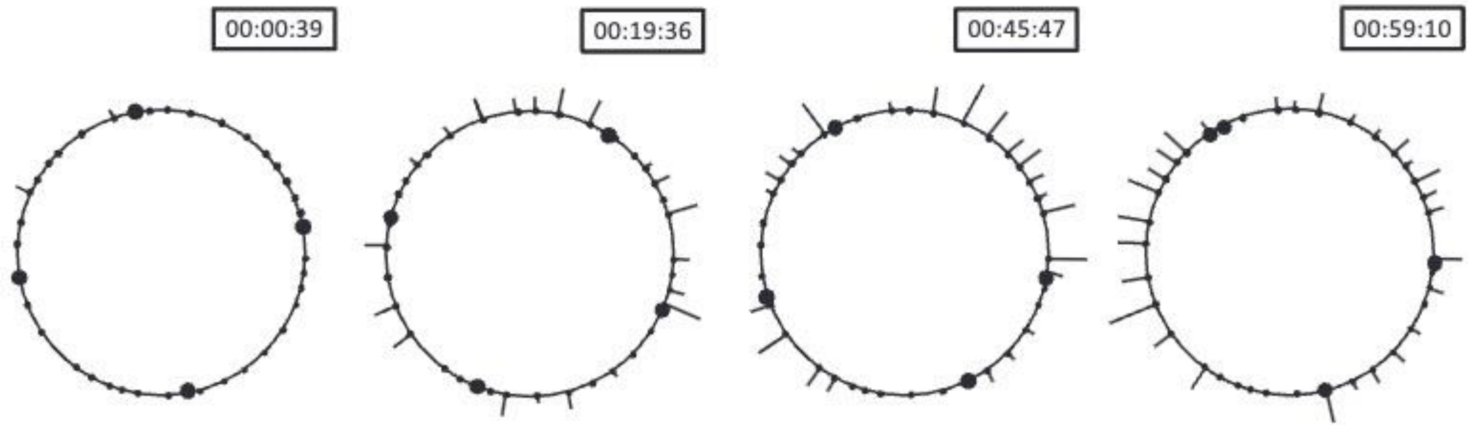
- "Bus bunching" phenomenon
  - Multiple buses arrive at the stop at the same time



- h: scheduled headway
- $h' > 0$: realized headway

Delay management

- "Bus bunching" phenomenon
  - Multiple buses arrive at the stop at the same time



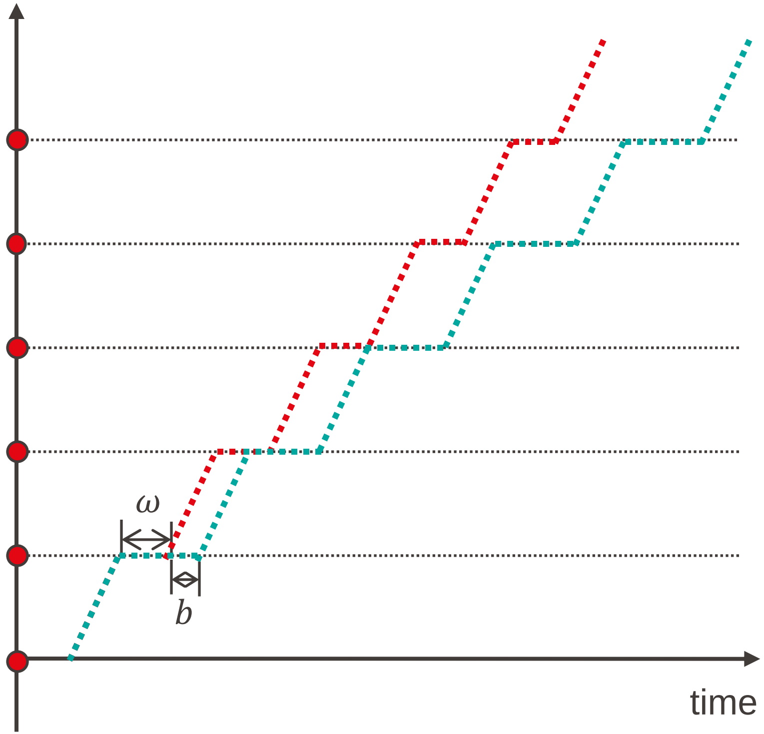
A simulation of bus bunching
https://www.youtube.com/shorts/V9vjvfbM7ko

Delay management

- Solutions to bus-bunching
  - Schedule-based method
    - simple and static
  - Dynamic delay method
    - more robust but require real-time info
  - Bus-splitting method
    - enabled by modular vehicle technologies

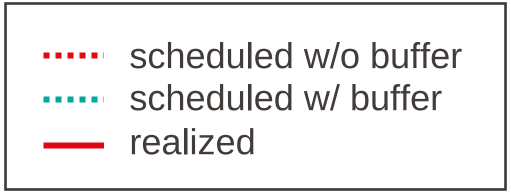
Delay management

- Schedule-based method
  - Add buffer time at some stops



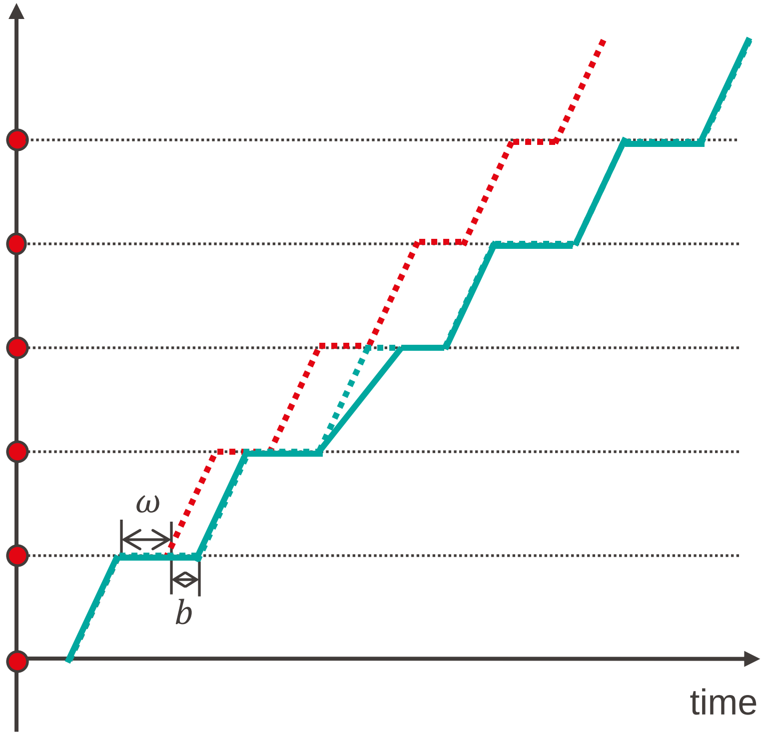
Strategy 1: add buffer b to every stop

- essentially extend stop time from ω to $\omega + b$



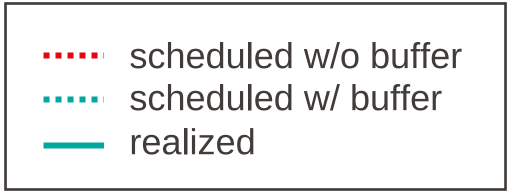
Delay management

- Schedule-based method
  - Add buffer time at some stops



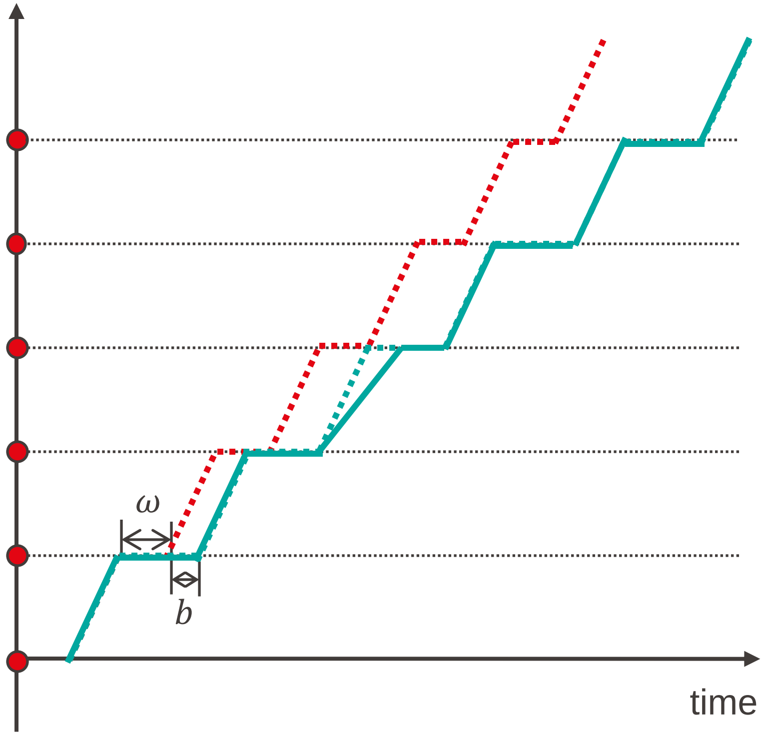
Strategy 1: add buffer b to every stop

- essentially extend stop time from ω to $\omega + b$
- schedule is not much affected as long as delay is shorter than $\omega + b$
- ***Q: What is the disadvantage?***



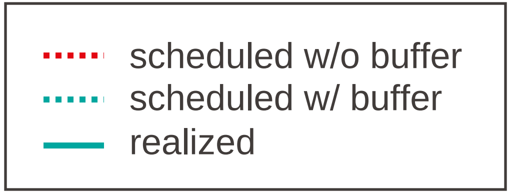
Delay management

- Schedule-based method
  - Add buffer time at some stops



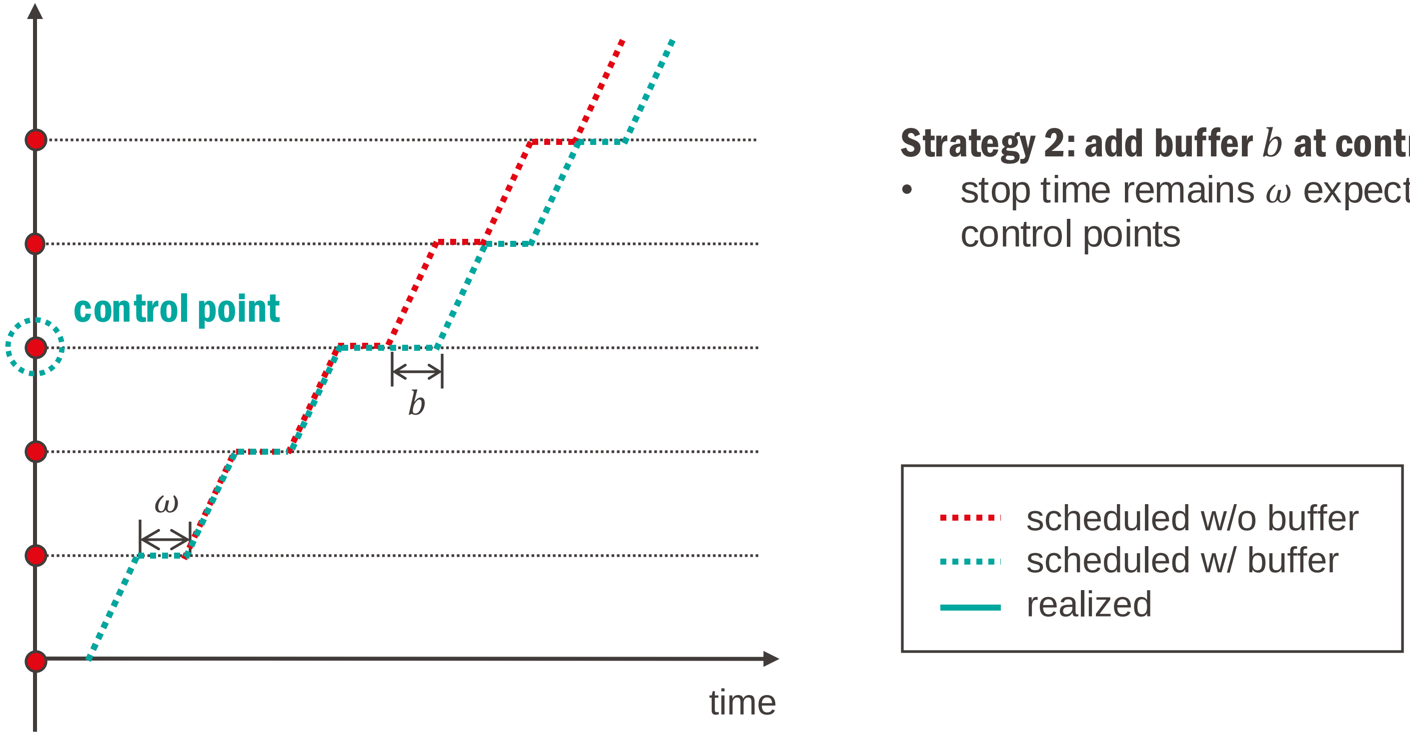
Strategy 1: add buffer b to every stop

- essentially extend stop time from ω to $\omega + b$
- schedule is not much affected as long as delay is shorter than $\omega + b$
- too conservative with much longer schedule



Delay management

- Schedule-based method
  - Add buffer time at some stops

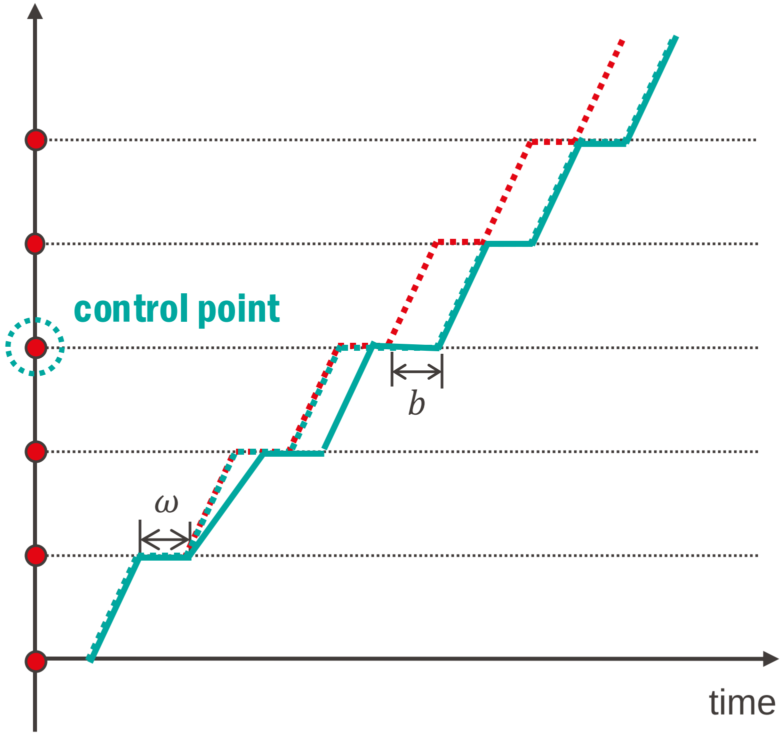


Strategy 2: add buffer b at control points

- stop time remains ω expect for the control points

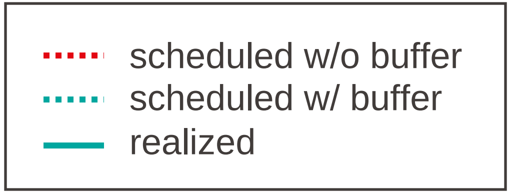
Delay management

- Schedule-based method
  - Add buffer time at some stops



Strategy 2: add buffer b at control points

- stop time remains ω expect for the control points
- balance between schedule robustness and efficiency
- both control point and buffer time can be optimized



Questions?

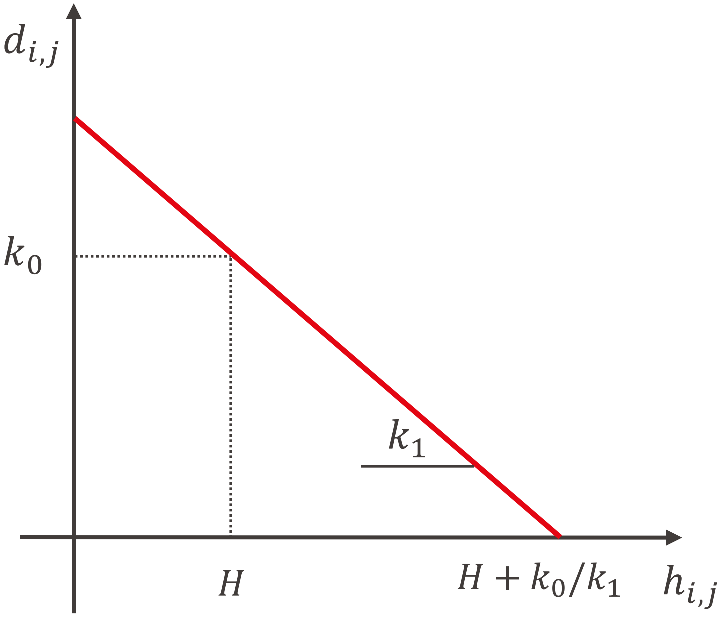
Delay management

- Dynamic delay method
  - Instead of a constant buffer time, add a dynamic delay $d_{i,j}$ for bus i at stop j

$$d_{i,j} = \left[k_0 - k_1(h_{i,j} - H)\right]_+$$

where

- k_0, k_1: constant parameters
- $h_{i,j}$: realized headway
- H: schedule headway
- $[x]_+ = \max\{0, x\}$



- ***Q: How to interpret the delay as function of realized headway?***

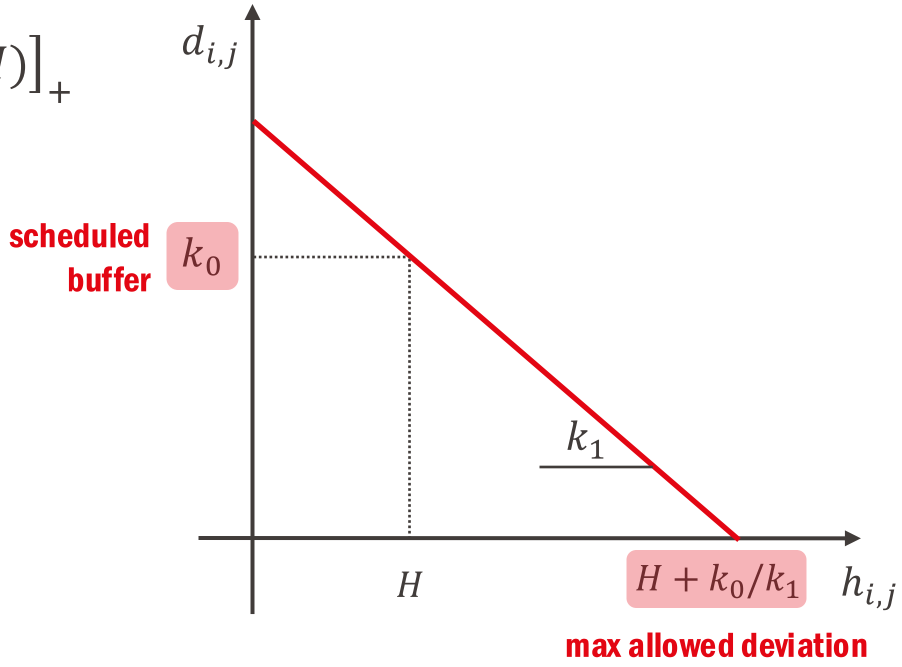
Delay management

- Dynamic delay method
  - Instead of a constant buffer time, add a dynamic delay $d_{i,j}$ for bus i at stop j

$$d_{i,j} = \left[k_0 - k_1(h_{i,j} - H)\right]_+$$

where

- k_0, k_1: constant parameters
- $h_{i,j}$: realized headway
- H: schedule headway
- $[x]_+ = \max\{0, x\}$



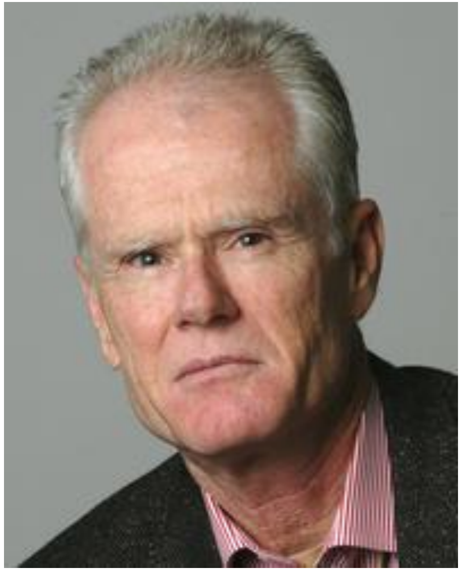
Delay management

- Dynamic delay method
  - Instead of a constant buffer time, add a dynamic delay $d_{i,j}$ for bus i at stop j
    - strategy proposed by Prof. Carlos Daganzo

$$d_{i,j} = \left[b - (\alpha + \beta)(h_{i,j} - H)\right]_+$$

where

- b: stop-wise buffer time
- α: ratio of average inter-stop travel time to scheduled headway
- β:extra inter-stop time per unit headway difference
- $h_{i,j}$: realized headway
- H: schedule headway
- $[x]_+ = \max\{0, x\}$



Delay management

- Dynamic delay method
  - Instead of a constant buffer time, add a dynamic delay $d_{i,j}$ for bus i at stop j
    - an extension considering the real-time coordination

$$d_{i,j} = \left[k_0 + k_1\left(h_{i,j} - h\right) + k_2(h_{i+1,j} - h)\right]_+$$

- add more delay to bus i at stop j if bus $i + 1$ comes late

- ***Q: Is $h_{i+1,j}$ available when computing $d_{i,j}$?***

Delay management

- Dynamic delay method
  - Instead of a constant buffer time, add a dynamic delay $d_{i,j}$ for bus i at stop j
    - an extension considering the real-time coordination

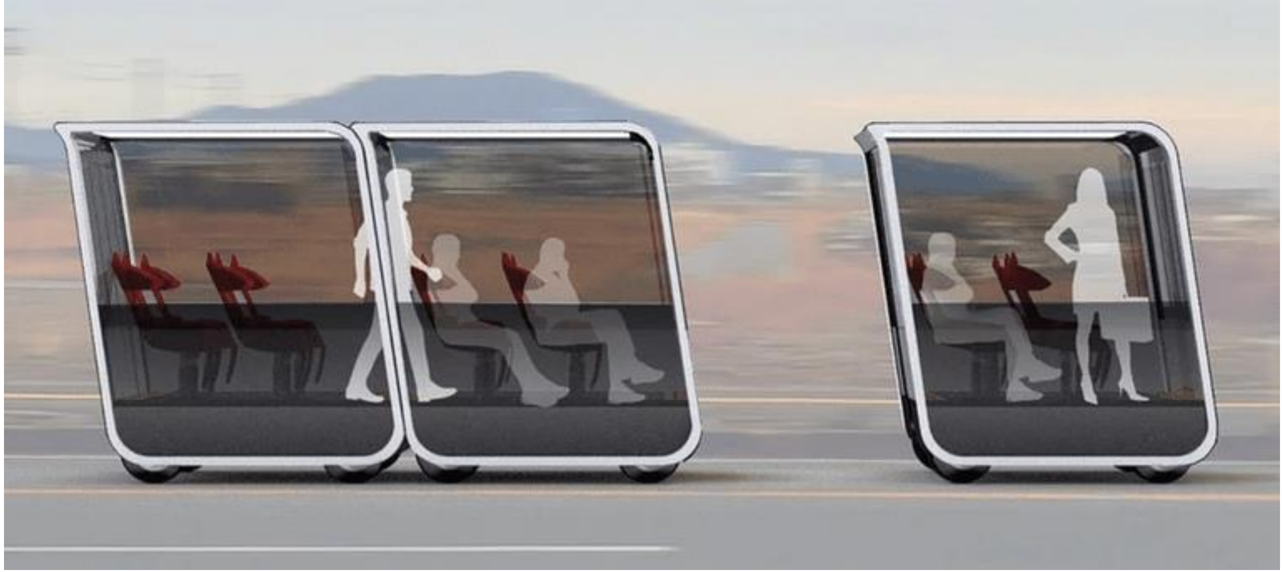
$$d_{i,j} = \left[k_0 + k_1\left(h_{i,j} - h\right) + k_2(\tilde{h}_{i+1,j} - h)\right]_+$$

- add more delay to bus i at stop j if bus $i+1$ comes late
- approximate $h_{i+1,j}$ by upstream headway of bus $i+1$, e.g., $h_{i+1,j-1}$

Questions?

Delay management

- Autonomous modular vehicle (AMV)

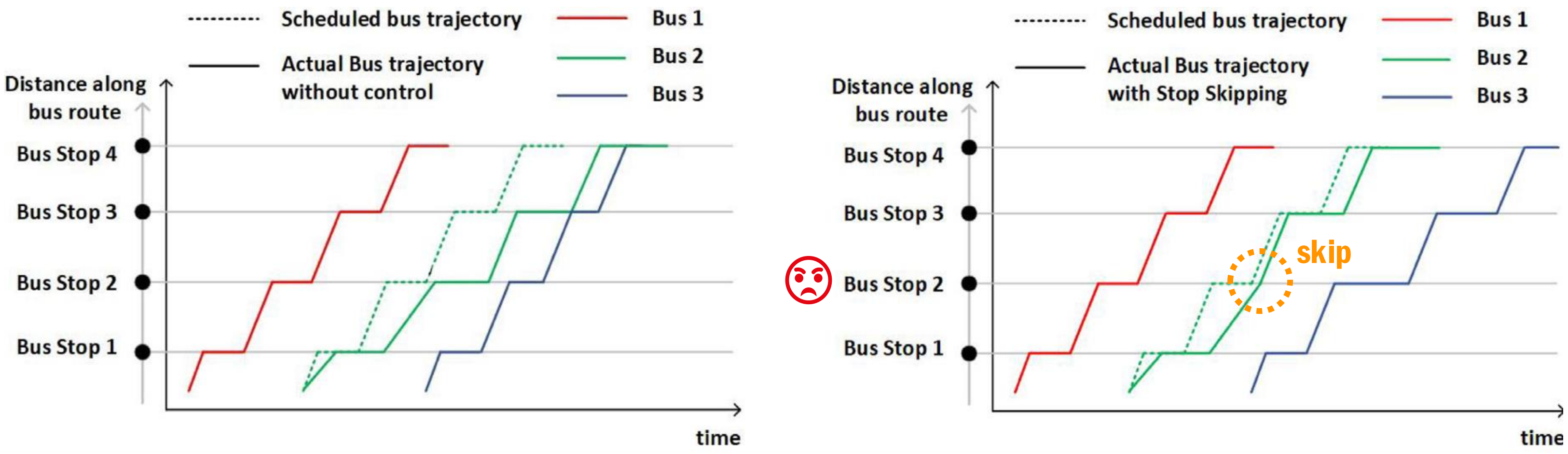


Autonomous modular vehicle (NEXT Future Transportation)

https://youtu.be/kJlQaCIUHTI?si=WIkiUsa1SBREugy6

Delay management

- Bus-splitting method
  - Baseline method: stop skipping
    - skip some stops to catch up the schedule
    - passengers at the skipped stop suffer from a doubled headway



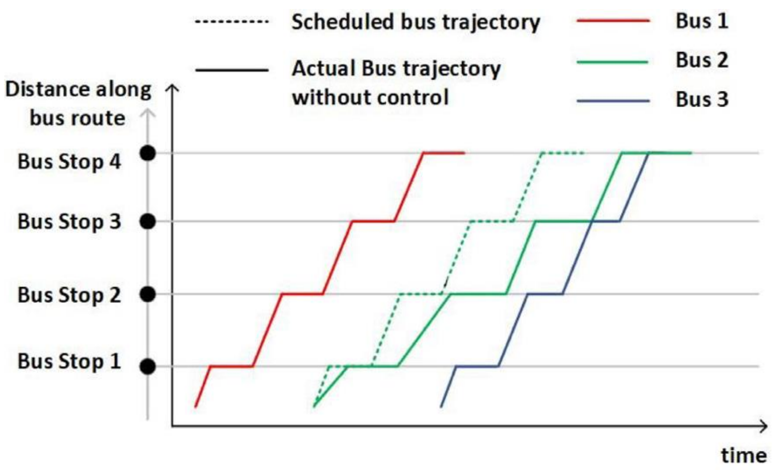
(a) No control

(b) Stop-Skipping

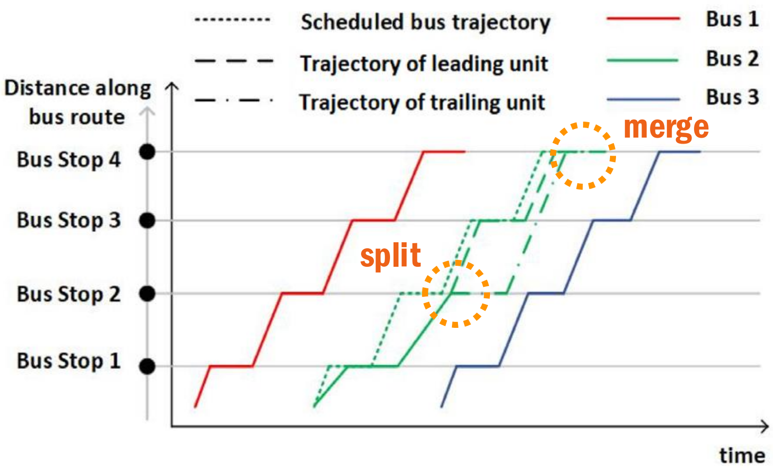
Khan et al. (2023) TRC

Delay management

- Bus-splitting method
  - Vehicle split into modules when a delay occurs
  - Each module only serves some stops but skips the others
  - Modules merge once catching up the schedule



(a) No control



(c) Bus-Splitting

Khan et al. (2023) TRC

Questions?