Dependable Dynamic Routing for Urban Transport Systems Through Integer Linear Programming

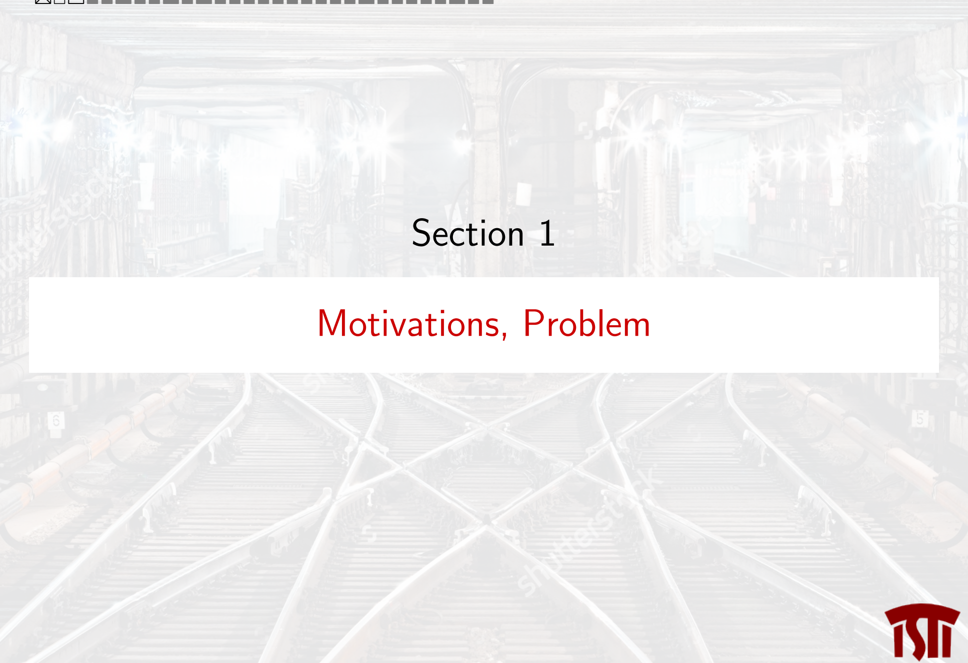
Davide Basile (1,2) Felicita Di Giandomenico (2) Stefania Gnesi (2)

(1) Dipartimento di Ingegneria Informatica, Università di Firenze
(2) Istituto di Scienza e Tecnologia dell'Informazione "A. Faedo",
Consiglio Nazionale delle Ricerche, ISTI-CNR, Pisa, Italy

16 November 2017

Overline of the presentation

1. Motivations, Problem
2. Summary of Contribution
3. Background
4. Dynamic Vehicle Routing Model
5. Implementation, Experiments
6. Conclusion



Section 1

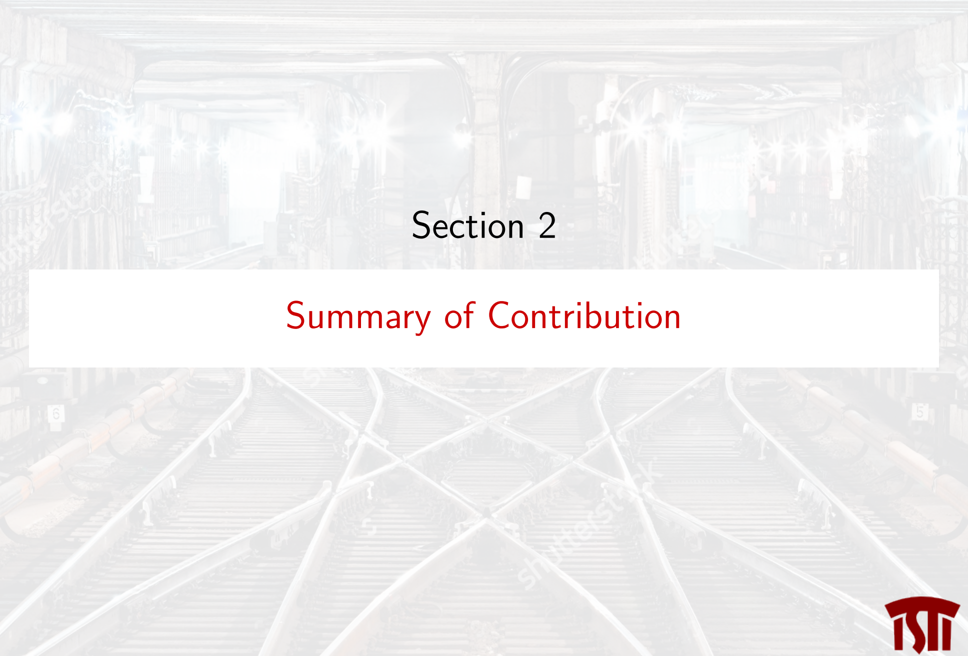
Motivations, Problem

Motivations

- Motivations: build novel solutions for transport systems, reduce *costs*, improve *sustainability*, *reliability* and *safety*;
- Driver-less transport solutions already deployed in metropolitan cities;
- **Optimal vehicles routes** → reduce costs, energy consumption, improve delivered service, user satisfiability;
- Traditionally two-step approaches: planning and execute;
    - assumption: track availability,
    - *problem*: not possible in urban area, e.g. track obstructions;
- *solution*: **Dynamic vehicles routing**, possible through novel technologies: GPS, GNSS, GIS, ITS, etc...

Dynamic Vehicle Routing Problem

- Widely studied and adopted for planning routes of vehicles and supervising their movements,
- Recently, solutions for train and subway lines have been extended to other urban transport systems, e.g. tramway lines,
- *Tramway lines*: characteristics
  - signals, priorities, traffic lights already adopted,
  - driver to enforce speed, braking, safety distance,
  - **problem**: tracks obstructions due to accidents, other vehicles etc..., needs to adopt temporary routes until normal operations are restored,
  - **innovation**: radar and gps systems to detected obstructions, dynamic routing to improve user satisfaction, energy consumption, etc...



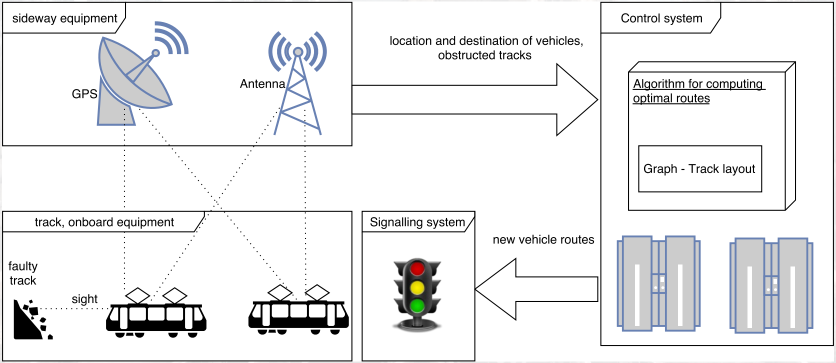
Section 2

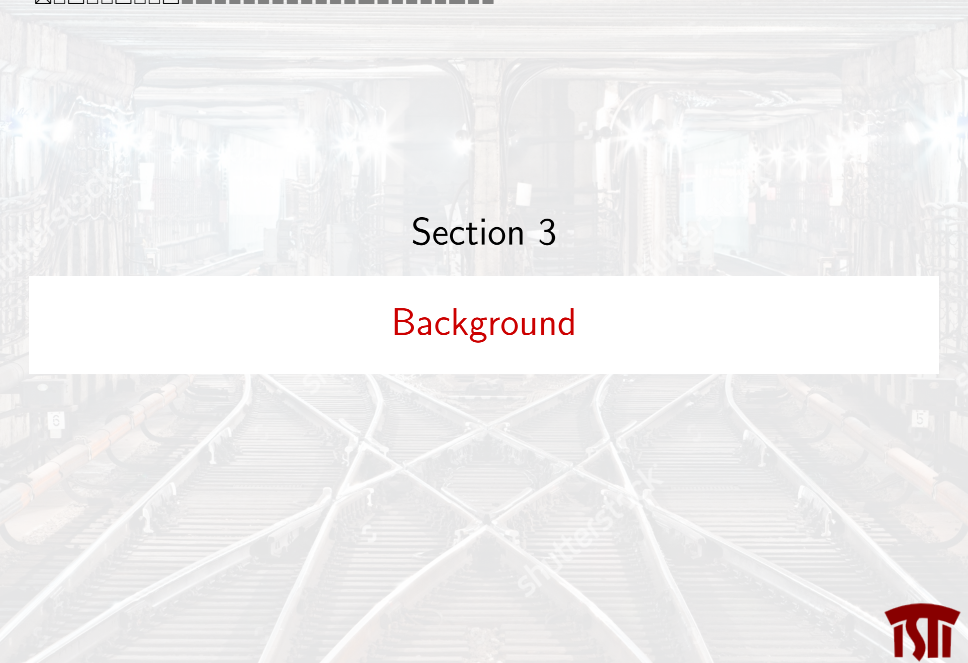
Summary of Contribution

Contribution

- **Dynamic routing algorithm** for urban area:
    - possible obstructed tracks can be detected,
    - new routes are dynamically recomputed
- Input: graph abstracting an urban map, location and destination of vehicles, unavailable tracks
    - nodes: points, edges: itineraries
- Output: **optimal routes** for each vehicle (deteriorated tracks) s.t.:
    - collisions on itineraries or points are *ruled out* on routes,
    - obstructed tracks are *not traversed* by routes,
    - *progress* of the network, *minimum* arriving time for all vehicles.
- modelled as an *optimization combinatorial problem*:
    - routes are flows in a graph (flow problem),
    - objective function: minimise arrival time,
    - linear constraints ensuring safety,
    - open-source implementation in AMPL.

Dynamic Vehicle Routing Architecture



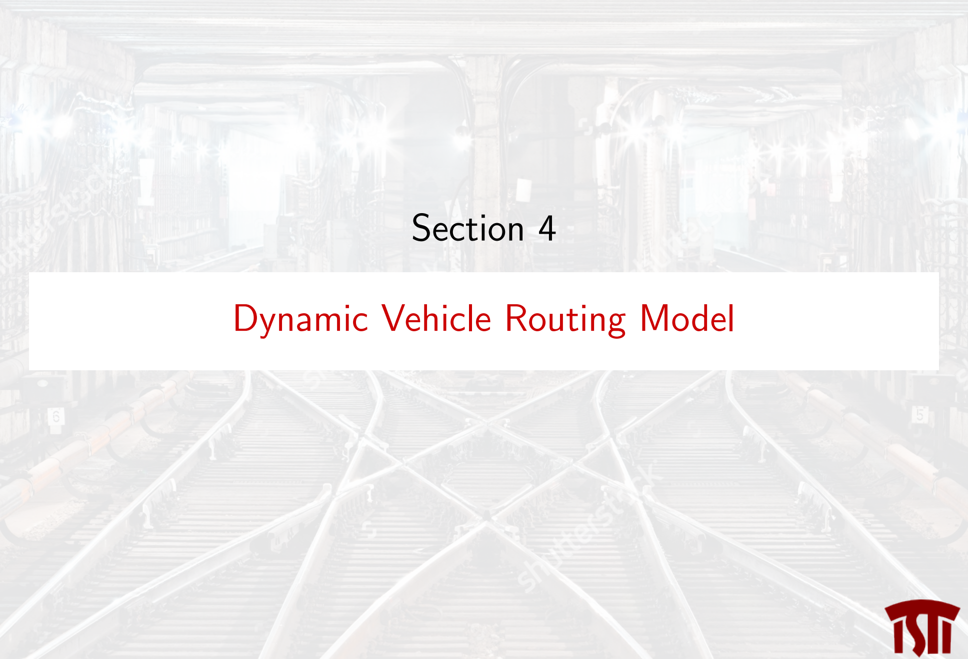


Section 3

Background

Network Flow Problem

- Flow network: directed graph $G = (Q, T)$ with source and sink, each edge has capacity and receives flow,
- $\forall q \in Q. FS(q)$ forwad star, $BS(q)$ backward star,
- $\forall t \in T. x_t$ flow variable, a_t capacity, c_t cost,
- network flow problem: optimize flow s.t.:
  - $\forall t \in T. x_t \leq a_t$ capacity,
  - $\forall q \in Q. \sum_{t \in BS(q)} x_t - \sum_{t \in FS(q)} x_t = \begin{cases} -d & \text{if } q = q_s \\ 0 & \text{if } q \neq q_s, q_f \\ d & \text{if } q = q_f \end{cases}$
  - $\forall t \in T. x_t \in \mathbb{N}$ integrity
- Objective function examples:
  - $\texttt{max}\ \ d$ (maximum flow problem)
  - $\texttt{min}\ \ \sum_{t \in T} x_t c_t$ (minimum cost flow problem)
- solved through ILP, e.g. simplex algorithm



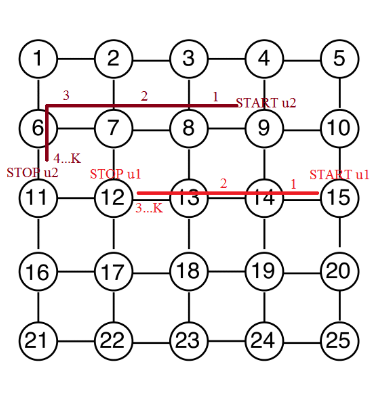
Section 4

Dynamic Vehicle Routing Model

Dynamic Vehicle Routing Model

- modeled as a *network flow problem*,
- routes will be flow in the graph abstracting the urban map:
  - maximum capacity $a_t = 1$ (only one vehicle in each itinerary),
  - flow $d = 1$ (each flow corresponds to a route),
  - costs left as future work (energy, performance, etc...)
- round-trips are not considered (e.g. destination: next-stop)
- U set of *vehicles*,
- it is possible to split the flow variables into time steps $1, \ldots, K$
  - vehicles can be in the same edge in different time steps,
- $K = |U| * |T|$: upper-bound to max. num. of steps,
- flow variables (routes) $x_{u,k,t}$:
  - $u \rightarrow$ vehicle, $k \rightarrow$ discrete time step, $t \rightarrow$ itinerary
- $x_{u,k,t} = 1$: vehicle u at time step k is in itinerary t,
- the model will optimize the values of variables $x_{u,k,t}$.

Dynamic Vehicle Routing Model (2)

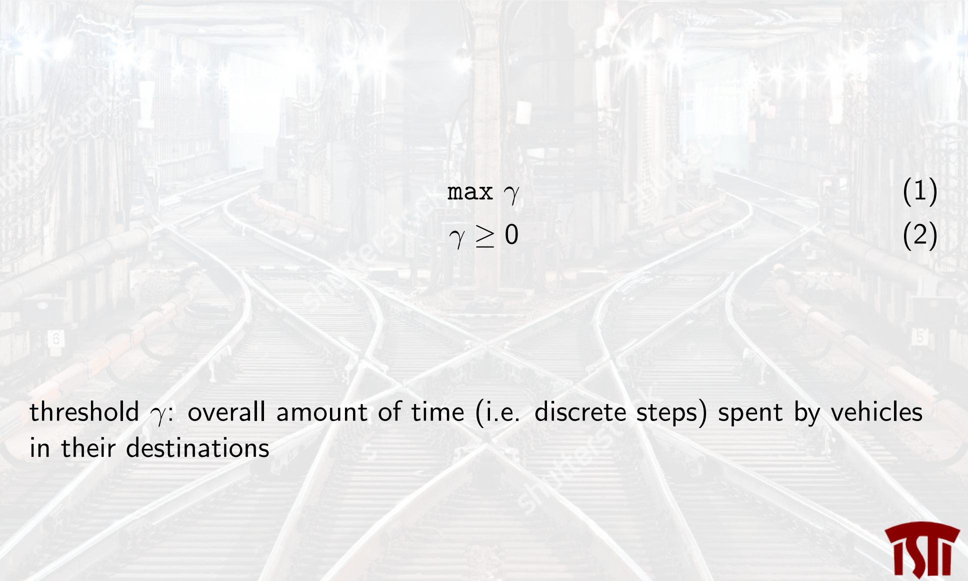


$$F = \{(1,2),(1,6),(2,1),(2,7),(6,1),(7,2),(10,2)\}$$

$$x_{u_1,1,(15,14)} = 1, x_{u_1,2,(14,13)} = 1, x_{u_1,i,(13,12)} = 1 \; i = 3,\ldots,K$$

$$x_{u_2,1,(9,8)} = 1, x_{u_2,2,(8,7)} = 1, x_{u_2,3,(7,6)} = 1, x_{u_2,j,(6,11)} = 1, \; j = 4,\ldots,K$$

Objective Function



$$\mathtt{max}\ \gamma \tag{1}$$

$$\gamma \geq 0 \tag{2}$$

threshold γ: overall amount of time (i.e. discrete steps) spent by vehicles in their destinations

Flow Constraints

binary flow variables

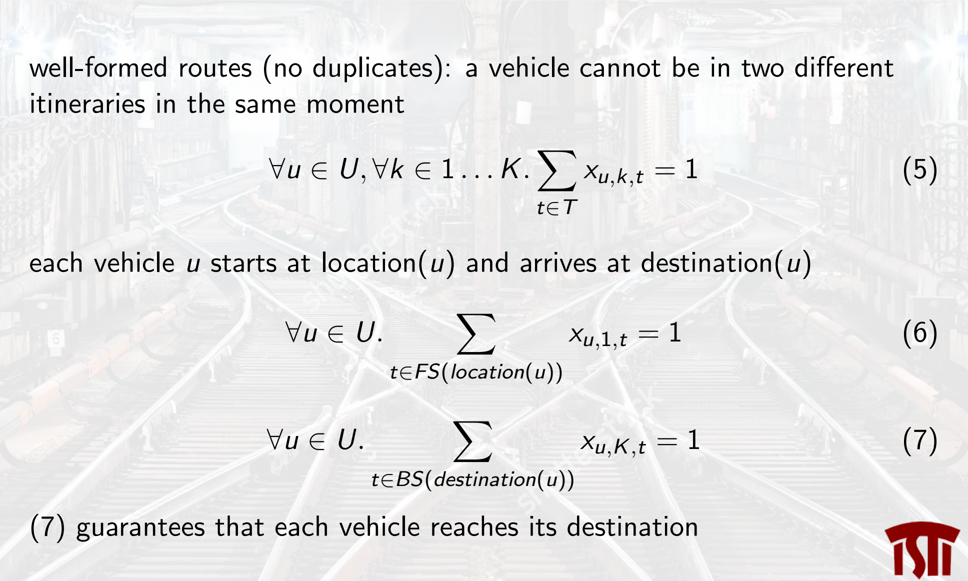
$$\forall k \in 1 \ldots K, \forall u \in U, \forall t \in T.\ x_{u,k,t} \in \{0,1\} \tag{3}$$

minimum arrival time:

- $\sum_{u \in U, k \in K, t \in BS(destination(u))} x_{u,k,t}$: amount of time that vehicles spent in their destinations
- at most $|K| * |U|$: the problem is bounded
- by maximizing this value the length of each route is minimized: the later vehicles arrive, the small the sum will be

$$\sum_{u \in U, k \in K, t \in BS(destination(u))} x_{u,k,t} \geq \gamma \tag{4}$$

Flow Constraints (1)



well-formed routes (no duplicates): a vehicle cannot be in two different itineraries in the same moment

$$\forall u \in U, \forall k \in 1 \ldots K. \sum_{t \in T} x_{u,k,t} = 1 \tag{5}$$

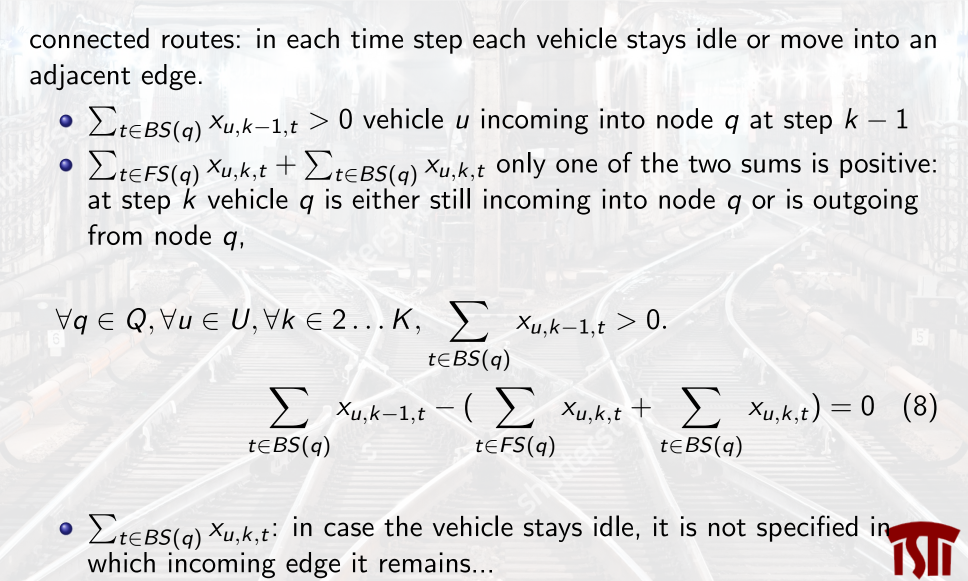
each vehicle u starts at location(u) and arrives at destination(u)

$$\forall u \in U. \sum_{t \in FS(location(u))} x_{u,1,t} = 1 \tag{6}$$

$$\forall u \in U. \sum_{t \in BS(destination(u))} x_{u,K,t} = 1 \tag{7}$$

(7) guarantees that each vehicle reaches its destination

Flow Constraints (2)



connected routes: in each time step each vehicle stays idle or move into an adjacent edge.

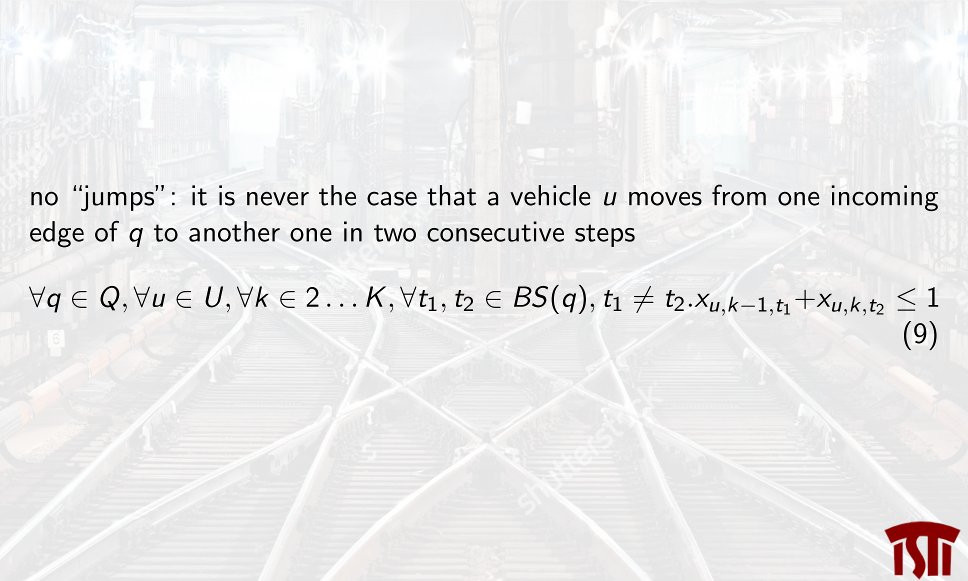
- $\sum_{t\in BS(q)} x_{u,k-1,t} > 0$ vehicle u incoming into node q at step $k-1$
- $\sum_{t\in FS(q)} x_{u,k,t} + \sum_{t\in BS(q)} x_{u,k,t}$ only one of the two sums is positive: at step k vehicle q is either still incoming into node q or is outgoing from node q,

$$\forall q \in Q, \forall u \in U, \forall k \in 2 \dots K, \sum_{t\in BS(q)} x_{u,k-1,t} > 0.$$

$$\sum_{t\in BS(q)} x_{u,k-1,t} - (\sum_{t\in FS(q)} x_{u,k,t} + \sum_{t\in BS(q)} x_{u,k,t}) = 0 \tag{8}$$

- $\sum_{t\in BS(q)} x_{u,k,t}$: in case the vehicle stays idle, it is not specified in which incoming edge it remains...

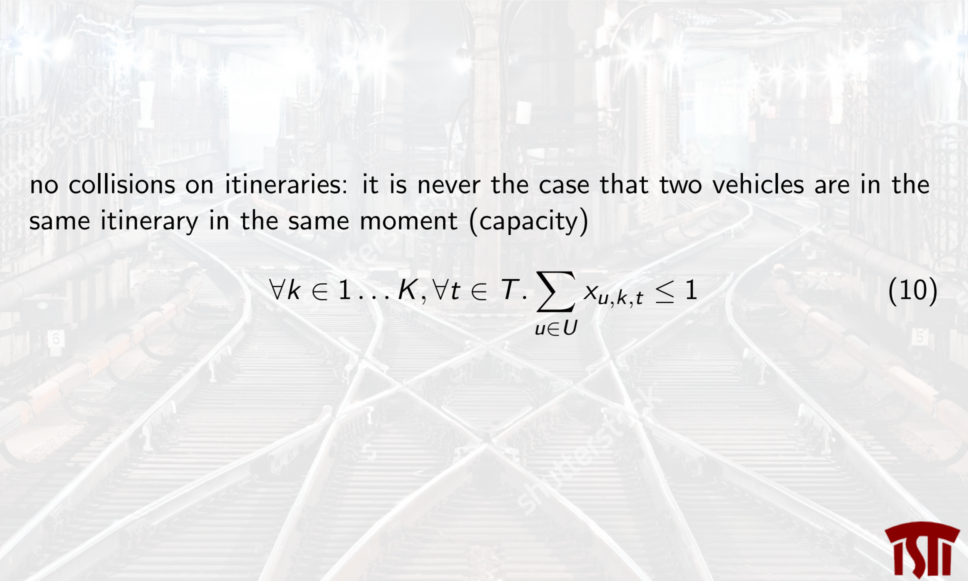
Flow Constraints (3)



no “jumps”: it is never the case that a vehicle u moves from one incoming edge of q to another one in two consecutive steps

$$\forall q \in Q, \forall u \in U, \forall k \in 2 \ldots K, \forall t_1, t_2 \in BS(q), t_1 \neq t_2 . x_{u,k-1,t_1} + x_{u,k,t_2} \leq 1 \tag{9}$$

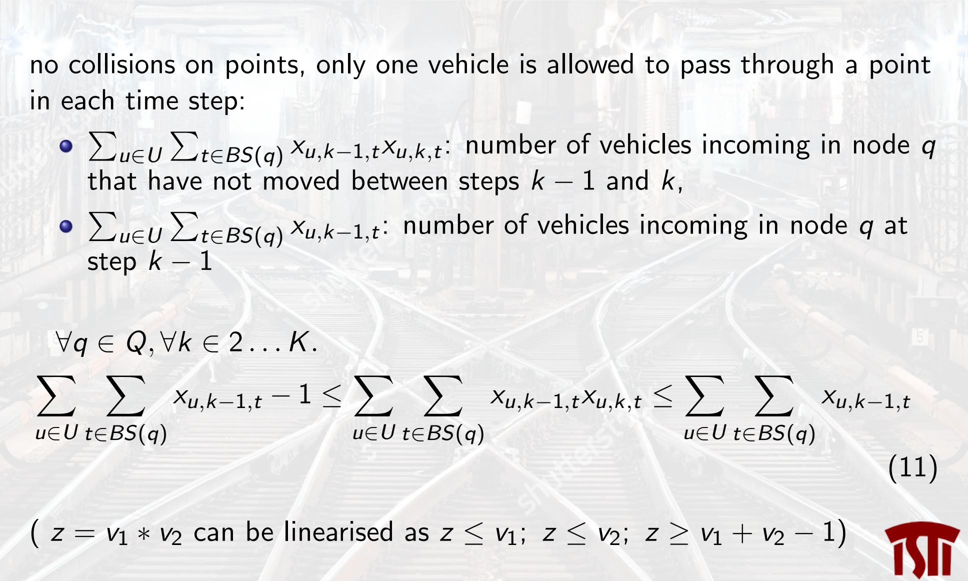
Safety Constraints



no collisions on itineraries: it is never the case that two vehicles are in the same itinerary in the same moment (capacity)

$$\forall k \in 1 \ldots K, \forall t \in T. \sum_{u \in U} x_{u,k,t} \leq 1 \tag{10}$$

Safety Constraints



no collisions on points, only one vehicle is allowed to pass through a point in each time step:

- $\sum_{u \in U} \sum_{t \in BS(q)} x_{u,k-1,t} x_{u,k,t}$: number of vehicles incoming in node q that have not moved between steps $k-1$ and k,
- $\sum_{u \in U} \sum_{t \in BS(q)} x_{u,k-1,t}$: number of vehicles incoming in node q at step $k-1$

$$\forall q \in Q, \forall k \in 2 \ldots K.$$

$$\sum_{u \in U} \sum_{t \in BS(q)} x_{u,k-1,t} - 1 \leq \sum_{u \in U} \sum_{t \in BS(q)} x_{u,k-1,t} x_{u,k,t} \leq \sum_{u \in U} \sum_{t \in BS(q)} x_{u,k-1,t} \tag{11}$$

($z = v_1 * v_2$ can be linearised as $z \leq v_1$; $z \leq v_2$; $z \geq v_1 + v_2 - 1$)

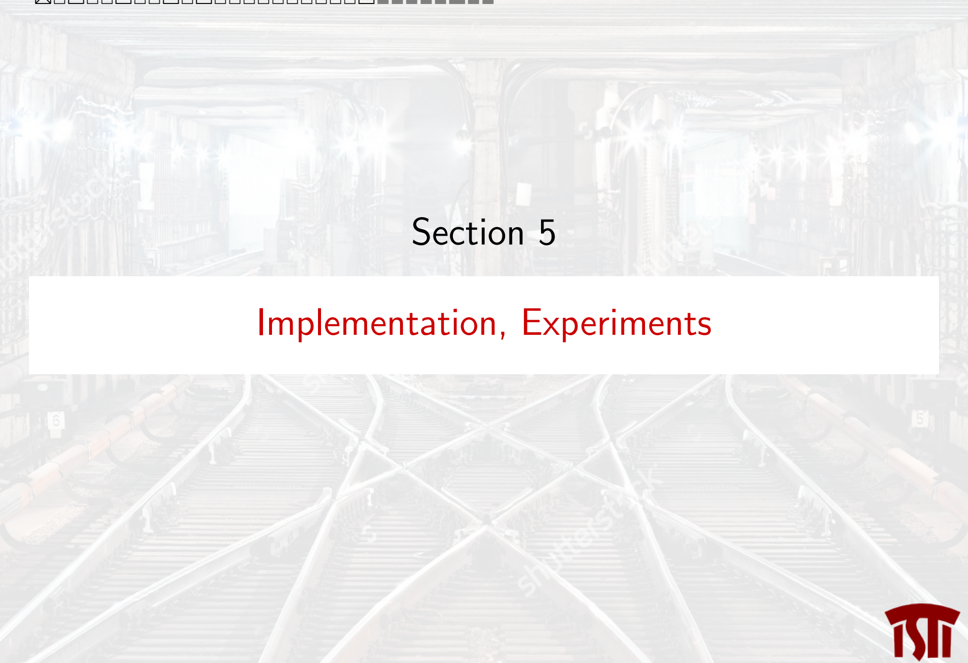
Safety Constraints - Graph Structure

unavailable itineraries are never traversed

$$\forall t \in F. \sum_{u \in U} \sum_{k \in 1\ldots K} x_{u,k,t} = 0 \tag{12}$$

graph structure: no inner cycles

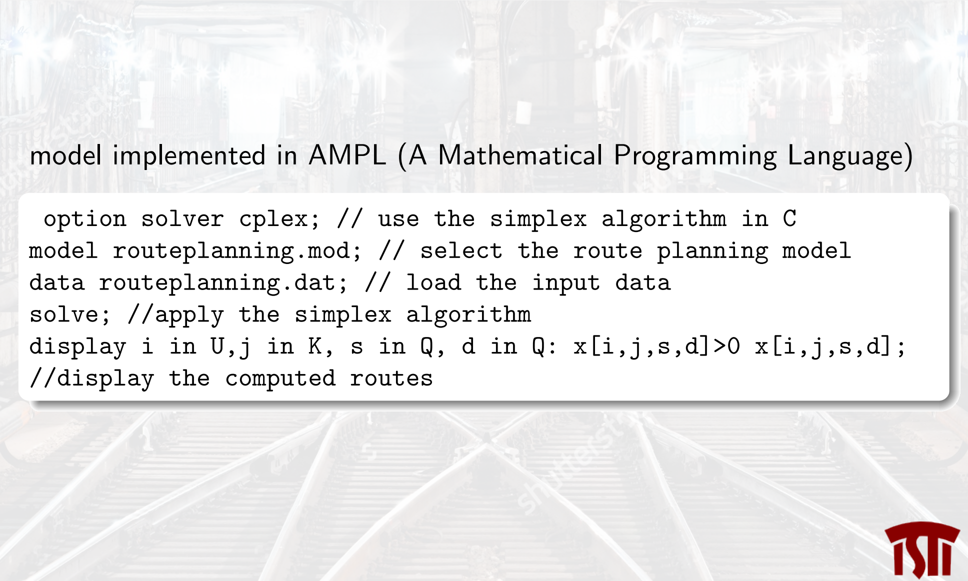
$$\forall q \in Q, \forall u \in U, \forall k \in 1 \ldots K, \forall t \in T. \sum_{t \in FS(q)} x_{u,k,t} + \sum_{t \in BS(q)} x_{u,k,t} \leq 1 \tag{13}$$



Section 5

Implementation, Experiments

Implementation



model implemented in AMPL (A Mathematical Programming Language)

```
 option solver cplex; // use the simplex algorithm in C
model routeplanning.mod; // select the route planning model
data routeplanning.dat; // load the input data
solve; //apply the simplex algorithm
display i in U,j in K, s in Q, d in Q: x[i,j,s,d]>0 x[i,j,s,d];
//display the computed routes
```

Implementation (2)

The input file `routeplanning.data` contains:

- number of vehicles u and nodes n,
- two binary matrix t and F
  - `t[q1,q2]=1` iff there is an edge $(q1, q2)$
  - `F[q1,q2]=1` iff itinerary $(q1, q2)$ is unavailable
- two arrays `location[u]` and `destination[u]`

Implementation (3)

The input file `routeplanning.mod` contains the model and additional constraints:
each flow only uses edges:

$$\forall u \in U, k \in K, s \in Q, d \in Q : x[u,k,s,d] <= t[s,d]$$

connected path:

$$\forall q \in Q, \forall u \in U, \forall k \in 2 \ldots K.$$

$$\sum_{t \in BS(q)} x_{u,k-1,t} - \Big(\sum_{t' \in FS(q), t \in BS(q)} x_{u,k,t'} \mathbf{x_{u,k-1,t}} + \sum_{t \in BS(q)} x_{u,k,t} \mathbf{x_{u,k-1,t}} \Big) = 0$$

linearization constraints

Experiments

random generation of obstructed tracks, locations and destinations

	Experiment 1
Obstructed tracks	(2,7) (6,7) (9,8) (11,6) (13,14) (17,12) (19,18) (22,23) (23,24)

Vehicle 1	location=1, destination=25
	route=(1,2)(2,3)(3,8)(8,9)(9,14)(14,15)(15,20)(20,25)

Vehicle 2	location=25, destination=1
	route=(25,20)(20,15)(15,10)(10,5)(5,4)(4,3)(3,2)(2,1)

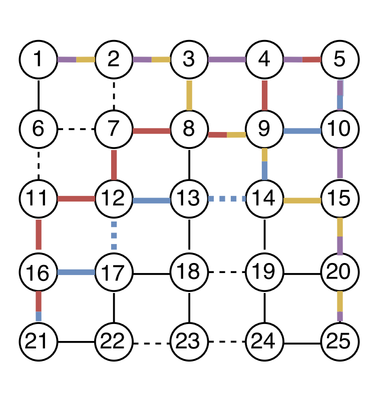
Vehicle 3	location=5, destination= 21
	route=(5,10)(10,9)(9,14)(14,13)(13,12)(12,17)(17,16)(16,21)

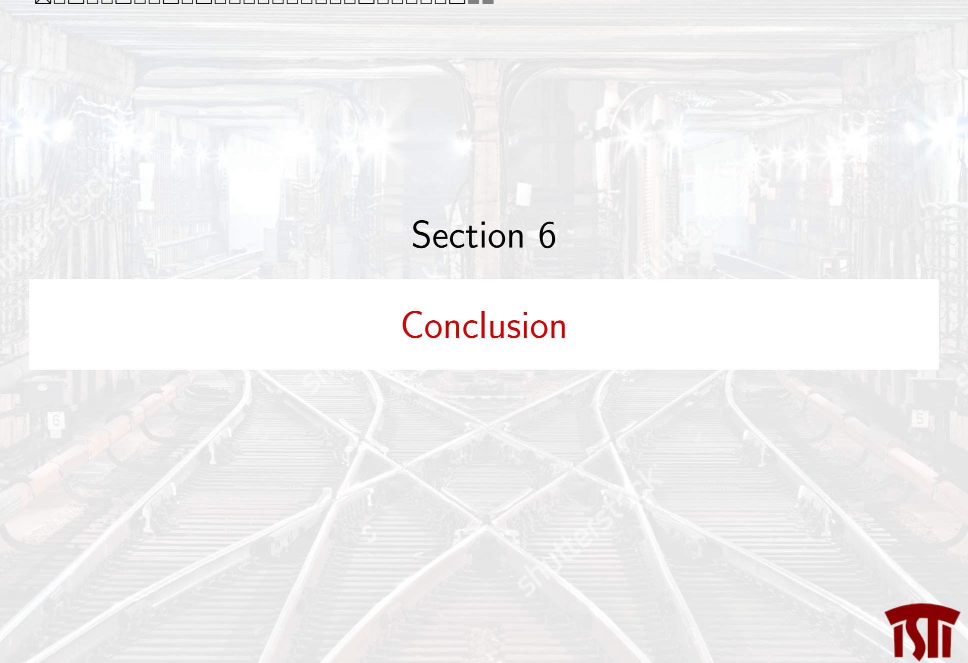
Vehicle 4	location=21, destination=5
	route=(21,16)(16,11)(11,12)(12,7)(7,8)(8,9)(9,4)(4,5)

	Experiment 2
Obstructed tracks	(1,6) (2,1) (2,7) (6,1) (6,7) (7,2) (9,8) (10,2) (10,15) (23,22)
Vehicle 1	location=14, destination=11, route=(14,13)(13,12)(12,11)
Vehicle 2	location=8, destination=10, route=(8,9)(9,10)
Vehicle 3	location=13, destination=18, route=(13,18)
Vehicle 4	location=21, destination=12, route=(21,16)(16,11)(11,12)
	Experiment 3
Obstructed tracks	(1,2) (1,6) (2,1) (2,7) (6,1) (7,2) (10,2)
Vehicle 1	location=15, destination=12, route=(15,14)(14,13)(13,12)
Vehicle 2	location=9, destination=11, route=(9,8)(8,7)(7,6)(6,11)
Vehicle 3	location=14, destination=19, route=(14,19)
Vehicle 4	location=22, destination=13, route=(22,17)(17,12)(12,13)

Experiment (2)

Experiment 1:





Section 6

Conclusion

Conclusion

- Dependable dynamic vehicle routing through ILP:
  - input: urban map (abstracted as a graph), vehicles locations and destinations (e.g. next stop), obstructed tracks,
  - output: set of routes with guarantees on,
    - absence of deadlocks, absence of collisions, minimum arriving time:
  - modeled as a flow problem on graph (nodes are points, edges are itineraries)
  - implemented in AMPL
- Future work:
  - apply to a (portion of a) real world urban scenario
  - include performance and energy aspects (e.g. speed, braking, fuel)
  - simulations on a whole day with failures

thanks for your attention