Macroscopic Fundamental Diagrams of arterial road networks governed by adaptive traffic signal systems

Tim Garoni

School of Mathematical Sciences
Monash University

Australian Government
Australian Research Council

Transportation Research Part B 49 (2013) 1–23

Contents lists available at SciVerse ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

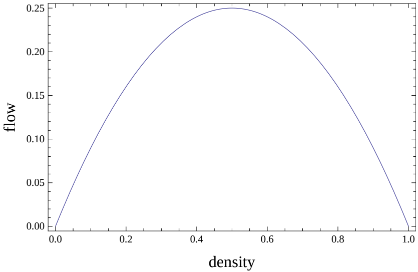
A comparative study of Macroscopic Fundamental Diagrams of arterial road networks governed by adaptive traffic signal systems

Lele Zhang [a,b], Timothy M Garoni [b,*], Jan de Gier [c]

[a] *ARC Centre of Excellence for Mathematics and Statistics of Complex Systems, Department of Mathematics and Statistics, University of Melbourne, Victoria 3010, Australia*
[b] *School of Mathematical Sciences, Monash University, Clayton, Victoria 3800, Australia*
[c] *Department of Mathematics and Statistics, The University of Melbourne, Victoria 3010, Australia*

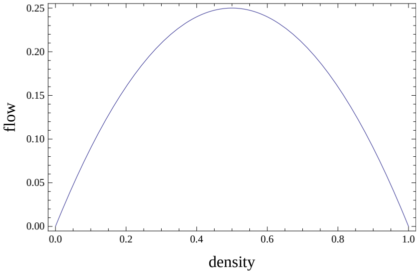
Fundamental Diagram

- Consider a one-dimensional flow (vehicles along a freeway)
- The functional relationship between flow and density is the **fundamental diagram** (Greenshields, 1935)



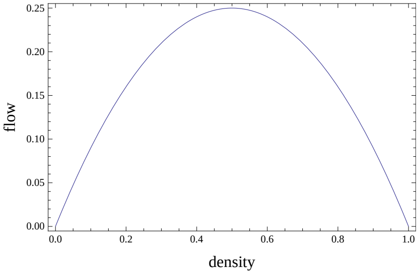
Fundamental Diagram

- Consider a one-dimensional flow (vehicles along a freeway)
- The functional relationship between flow and density is the **fundamental diagram** (Greenshields, 1935)



Fundamental Diagram

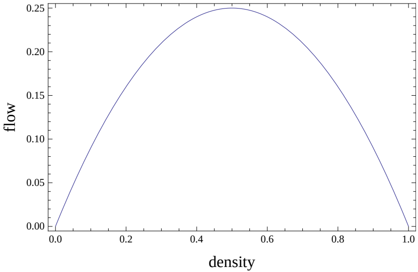
- Consider a one-dimensional flow (vehicles along a freeway)
- The functional relationship between flow and density is the **fundamental diagram** (Greenshields, 1935)



- Intuitively makes sense to have a unimodal FD in one dimension

Fundamental Diagram

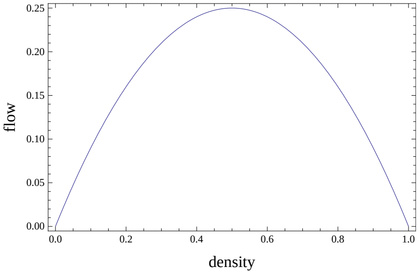
- Consider a one-dimensional flow (vehicles along a freeway)
- The functional relationship between flow and density is the **fundamental diagram** (Greenshields, 1935)



- Intuitively makes sense to have a unimodal FD in one dimension
- What should happen in a network?

Fundamental Diagram

- Consider a one-dimensional flow (vehicles along a freeway)
- The functional relationship between flow and density is the **fundamental diagram** (Greenshields, 1935)



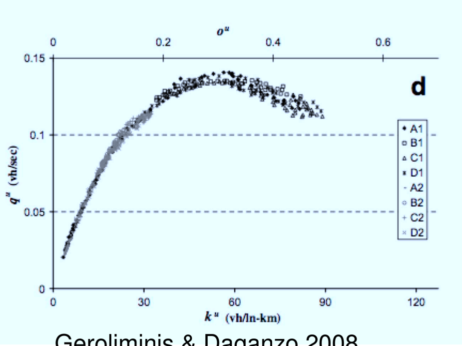
- Intuitively makes sense to have a unimodal FD in one dimension
- What should happen in a network?
- How should one even **define** network flow?
(No prescribed direction)

Macroscopic Fundamental Diagrams

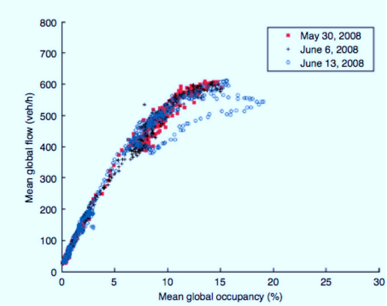
- Simplest idea: relate arithmetic means of link density and flow
- If network has link set Λ:

$$\rho = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}\rho_\lambda, \qquad J = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}J_\lambda$$

- ρ_λ is density of link λ and J_λ is its flow



Geroliminis & Daganzo 2008
Empirical data from Yokohama



Buisson & Ladier 2009
Empirical data from Toulouse

Two Extreme Cases

- Existence of MFDs is trivial:
  - If all links have the same FD
  - and if the distribution of congestion is always perfectly uniform
  - then network MFD coincides with common link FD

Two Extreme Cases

- Existence of MFDs is trivial:
  - If all links have the same FD
  - and if the distribution of congestion is always perfectly uniform
  - then network MFD coincides with common link FD
  - This is not very interesting...

Two Extreme Cases

- Existence of MFDs is trivial:
  - If all links have the same FD
  - and if the distribution of congestion is always perfectly uniform
  - then network MFD coincides with common link FD
  - This is not very interesting...
- Existence of MFDs is impossible:
  - If one has a network and is free to vary the demand on each link in any way imaginable, then no MFD can exist
  - e.g. half the links have $\rho_\lambda = 1$ and other half have $\rho_\lambda = 0$, then $\rho = 1/2$ and $J = 0$
  - e.g. all links have $\rho_\lambda = 1/2$, then $\rho = 1/2$ but $J > 0$ (could even have $J = J_{\text{max}}$)
  - Existence of MFDs clearly not independent of demand

Two Extreme Cases

- Existence of MFDs is trivial:
  - If all links have the same FD
  - and if the distribution of congestion is always perfectly uniform
  - then network MFD coincides with common link FD
  - This is not very interesting...
- Existence of MFDs is impossible:
  - If one has a network and is free to vary the demand on each link in any way imaginable, then no MFD can exist
  - e.g. half the links have $\rho_\lambda = 1$ and other half have $\rho_\lambda = 0$, then $\rho = 1/2$ and $J = 0$
  - e.g. all links have $\rho_\lambda = 1/2$, then $\rho = 1/2$ but $J > 0$ (could even have $J = J_{\text{max}}$)
  - Existence of MFDs clearly not independent of demand
- MFDs are interesting because there is something in between
- In practice, on many networks the demand will rise and fall in a fairly constrained way during a typical day

What are MFDs?

Consider a fixed network with link set Λ

First of all, one needs to agree on what ρ and J mean.

- $\rho_\lambda(t)$ and $J_\lambda(t)$ are stochastic processes
- Aggregate variables

$$\rho(t) = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}\rho_\lambda(t) \qquad J(t) = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}J_\lambda(t)$$

- MFD is the relationship between $\mathbb{E}J(t)$ and $\mathbb{E}\rho(t)$
- Can be interested in instantaneous or stationary MFDs

What are MFDs?

Consider a fixed network with link set Λ

First of all, one needs to agree on what ρ and J mean.

- $\rho_\lambda(t)$ and $J_\lambda(t)$ are stochastic processes
- Aggregate variables

$$\rho(t) = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}\rho_\lambda(t) \qquad J(t) = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}J_\lambda(t)$$

- MFD is the relationship between $\mathbb{E}J(t)$ and $\mathbb{E}\rho(t)$
- Can be interested in instantaneous or stationary MFDs
- "Heterogeneity" is also important

$$h(t) = \sqrt{\frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda}[\rho_\lambda(t) - \rho(t)]^2}$$

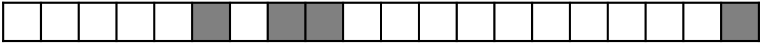
  Helbing 2009; Mazloumian, Geroliminis & Helbing 2010; Geroliminis & Sun 2011; de Gier, G & Zhang 2013
- J, ρ, h all stochastic processes
- In time dependent context, heterogeneity can explain hysteresis

Asymmetric Simple Exclusion Process (ASEP)

"*Everything should be made as simple as possible, but not simpler*"

(Albert Einstein)

- Want an Ising model of traffic flow
- One-dimensional stochastic cellular automata very popular in statistical mechanics starting in 1990s
  - Such models do a reasonable job of explaining qualitative behaviour of freeway traffic
  - "Phantom" jams emerge as consequence of collective behaviour
- Cellular automata are discrete dynamical systems
- Space, time, and state variables are discrete

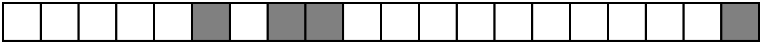


Asymmetric Simple Exclusion Process (ASEP)

"*Everything should be made as simple as possible, but not simpler*"

(Albert Einstein)

- Want an Ising model of traffic flow
- One-dimensional stochastic cellular automata very popular in statistical mechanics starting in 1990s
  - Such models do a reasonable job of explaining qualitative behaviour of freeway traffic
  - "Phantom" jams emerge as consequence of collective behaviour
- Cellular automata are discrete dynamical systems
- Space, time, and state variables are discrete



- ASEP with open boundaries:
  - If $x_1(t) = 0$, then with probability α, $x_1(t+1) = 1$
  - For each cell $i = 1, \ldots, L$ with $x_i(t) = 1$
    - If $x_{i+1}(t) = 0$ then with probability p, $x_i(t+1) = 0$ and $x_{i+1}(t+1) = 1$
    - Else $x_i(t+1) = 1$
  - If $x_L(t) = 1$, then with probability β, $x_L(t+1) = 0$

Asymmetric Simple Exclusion Process (ASEP)

"*Everything should be made as simple as possible, but not simpler*"

(Albert Einstein)

- Want an Ising model of traffic flow
- One-dimensional stochastic cellular automata very popular in statistical mechanics starting in 1990s
  - Such models do a reasonable job of explaining qualitative behaviour of freeway traffic
  - "Phantom" jams emerge as consequence of collective behaviour
- Cellular automata are discrete dynamical systems
- Space, time, and state variables are discrete

- ASEP with open boundaries:
  - If $x_1(t) = 0$, then with probability α, $x_1(t+1) = 1$
  - For each cell $i = 1, \ldots, L$ with $x_i(t) = 1$
    - If $x_{i+1}(t) = 0$ then with probability p, $x_i(t+1) = 0$ and $x_{i+1}(t+1) = 1$
    - Else $x_i(t+1) = 1$
  - If $x_L(t) = 1$, then with probability β, $x_L(t+1) = 0$

Asymmetric Simple Exclusion Process (ASEP)

"*Everything should be made as simple as possible, but not simpler*"

(Albert Einstein)

- Want an Ising model of traffic flow
- One-dimensional stochastic cellular automata very popular in statistical mechanics starting in 1990s
  - Such models do a reasonable job of explaining qualitative behaviour of freeway traffic
  - "Phantom" jams emerge as consequence of collective behaviour
- Cellular automata are discrete dynamical systems
- Space, time, and state variables are discrete

- ASEP with open boundaries:
  - If $x_1(t) = 0$, then with probability α, $x_1(t+1) = 1$
  - For each cell $i = 1, \ldots, L$ with $x_i(t) = 1$
    - If $x_{i+1}(t) = 0$ then with probability p, $x_i(t+1) = 0$ and $x_{i+1}(t+1) = 1$
    - Else $x_i(t+1) = 1$
  - If $x_L(t) = 1$, then with probability β, $x_L(t+1) = 0$

Asymmetric Simple Exclusion Process (ASEP)

"*Everything should be made as simple as possible, but not simpler*"

(Albert Einstein)

- Want an Ising model of traffic flow
- One-dimensional stochastic cellular automata very popular in statistical mechanics starting in 1990s
  - Such models do a reasonable job of explaining qualitative behaviour of freeway traffic
  - "Phantom" jams emerge as consequence of collective behaviour
- Cellular automata are discrete dynamical systems
- Space, time, and state variables are discrete

- ASEP with open boundaries:
  - If $x_1(t) = 0$, then with probability α, $x_1(t+1) = 1$
  - For each cell $i = 1, \ldots, L$ with $x_i(t) = 1$
    - If $x_{i+1}(t) = 0$ then with probability p, $x_i(t+1) = 0$ and $x_{i+1}(t+1) = 1$
    - Else $x_i(t+1) = 1$
  - If $x_L(t) = 1$, then with probability β, $x_L(t+1) = 0$

Asymmetric Simple Exclusion Process (ASEP)

"*Everything should be made as simple as possible, but not simpler*"

(Albert Einstein)

- Want an Ising model of traffic flow
- One-dimensional stochastic cellular automata very popular in statistical mechanics starting in 1990s
  - Such models do a reasonable job of explaining qualitative behaviour of freeway traffic
  - "Phantom" jams emerge as consequence of collective behaviour
- Cellular automata are discrete dynamical systems
- Space, time, and state variables are discrete

- ASEP with open boundaries:
  - If $x_1(t) = 0$, then with probability α, $x_1(t+1) = 1$
  - For each cell $i = 1, \dots, L$ with $x_i(t) = 1$
    - If $x_{i+1}(t) = 0$ then with probability p, $x_i(t+1) = 0$ and $x_{i+1}(t+1) = 1$
    - Else $x_i(t+1) = 1$
  - If $x_L(t) = 1$, then with probability β, $x_L(t+1) = 0$

Asymmetric Simple Exclusion Process (ASEP)

"*Everything should be made as simple as possible, but not simpler*"

(Albert Einstein)

- Want an Ising model of traffic flow
- One-dimensional stochastic cellular automata very popular in statistical mechanics starting in 1990s
  - Such models do a reasonable job of explaining qualitative behaviour of freeway traffic
  - "Phantom" jams emerge as consequence of collective behaviour
- Cellular automata are discrete dynamical systems
- Space, time, and state variables are discrete

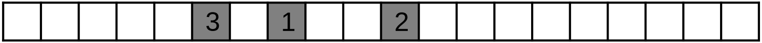
- ASEP with open boundaries:
  - If $x_1(t) = 0$, then with probability α, $x_1(t+1) = 1$
  - For each cell $i = 1, \ldots, L$ with $x_i(t) = 1$
    - If $x_{i+1}(t) = 0$ then with probability p, $x_i(t+1) = 0$ and $x_{i+1}(t+1) = 1$
    - Else $x_i(t+1) = 1$
  - If $x_L(t) = 1$, then with probability β, $x_L(t+1) = 0$

Nagel-Schreckenberg process

- NaSch generalizes ASEP

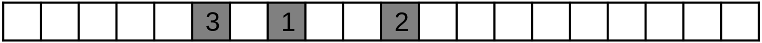
Nagel-Schreckenberg process

- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



Nagel-Schreckenberg process

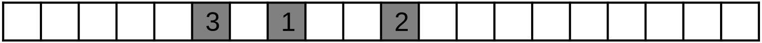
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\text{max}}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle

Nagel-Schreckenberg process

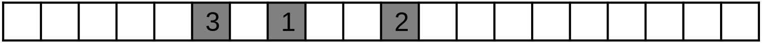
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle

Nagel-Schreckenberg process

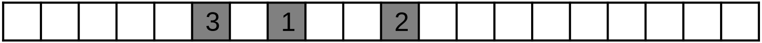
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \dots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:

Nagel-Schreckenberg process

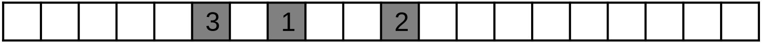
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$

Nagel-Schreckenberg process

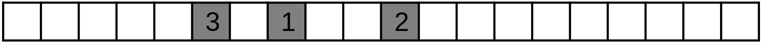
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$
  - $v_n \mapsto \min(v_n, d_n)$

Nagel-Schreckenberg process

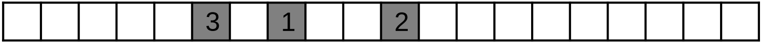
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$
  - $v_n \mapsto \min(v_n, d_n)$
  - $v_n \mapsto \max(v_n - 1, 0)$ with probability p

Nagel-Schreckenberg process

- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$
  - $v_n \mapsto \min(v_n, d_n)$
  - $v_n \mapsto \max(v_n - 1, 0)$ with probability p
  - $x_n \mapsto x_n + v_n$

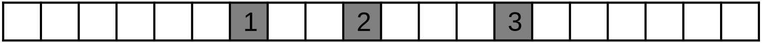
Nagel-Schreckenberg process

- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$

- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$
  - $v_n \mapsto \min(v_n, d_n)$
  - $v_n \mapsto \max(v_n - 1, 0)$ with probability p
  - $x_n \mapsto x_n + v_n$

Nagel-Schreckenberg process

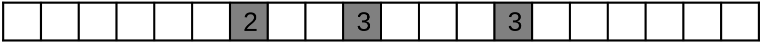
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$
  - $v_n \mapsto \min(v_n, d_n)$
  - $v_n \mapsto \max(v_n - 1, 0)$ with probability p
  - $x_n \mapsto x_n + v_n$

Nagel-Schreckenberg process

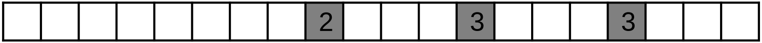
- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$



- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$
  - $v_n \mapsto \min(v_n, d_n)$
  - $v_n \mapsto \max(v_n - 1, 0)$ with probability p
  - $x_n \mapsto x_n + v_n$

Nagel-Schreckenberg process

- NaSch generalizes ASEP
  - Vehicles can have different speeds $0, 1, \ldots, v_{\max}$

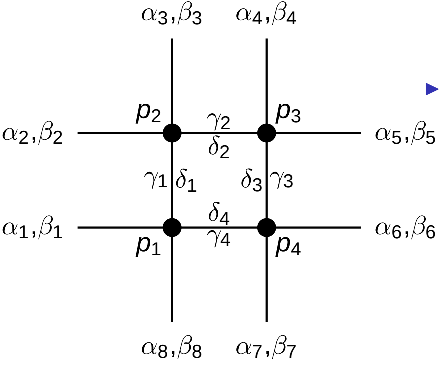


- Let x_n and v_n denote the position & speed of the *nth* vehicle
- Let d_n denote the gap in front of the *nth* vehicle
- The NaSch rules are as follows:
  - $v_n \mapsto \min(v_n + 1, v_{\max})$
  - $v_n \mapsto \min(v_n, d_n)$
  - $v_n \mapsto \max(v_n - 1, 0)$ with probability p
  - $x_n \mapsto x_n + v_n$

NetNaSch model

Goal: Minimal stat-mech model that can mimic realistic traffic signals

- Take multiple NaSch models and glue them together

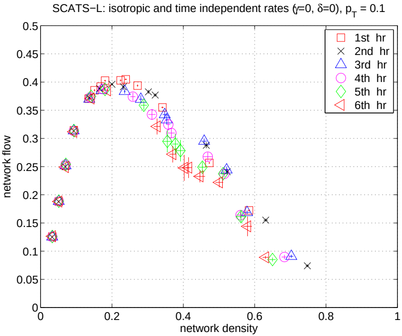


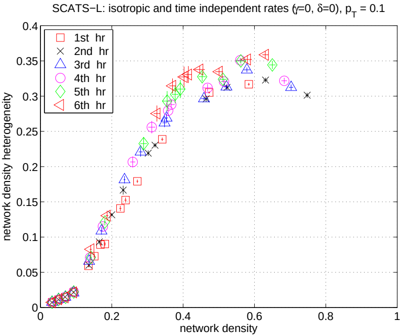
- Need to include:
  - Multiple lanes with lane changing
  - Turning decisions (random)
  - Input and output (endogenous/exogenous)
  - Appropriate rules for how vehicles traverse intersections

Varying all the $\alpha_\lambda, \beta_\lambda, \gamma_\lambda, \delta_\lambda, p_n$... cannot give an MFD
Varying a lower-dimensional space of parameters can

Static demand – Approach to Stationarity

Generate MFD by setting $\alpha_\lambda = \alpha$, $\beta_\lambda = \beta$, $\gamma_\lambda = \delta_\lambda = 0$ for all $\lambda \in \Lambda$

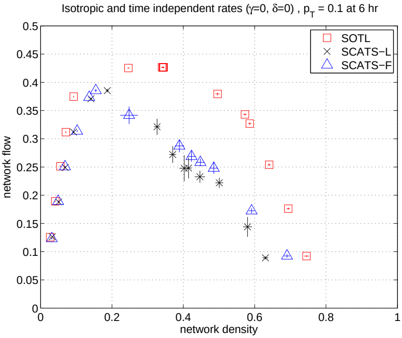




- Intersections governed by model of SCATS with adaptive linking
- Instantaneous MFD converges to stationary curve
- Although there is uniform boundary demand, the density distribution in the network is not homogeneous

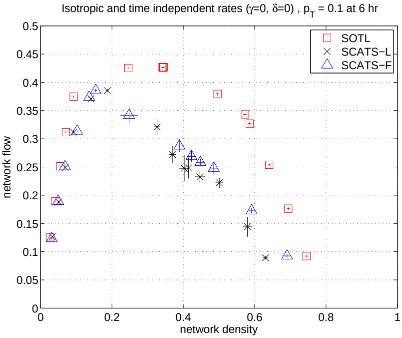
Static demand – Stationary MFDs

- Use MFDs to quantify performance of signal systems

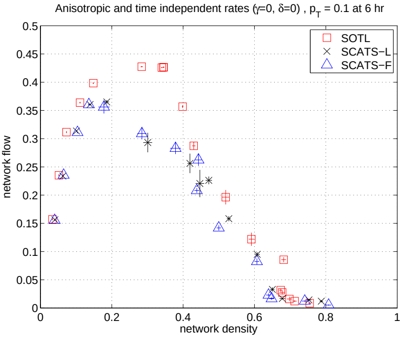


Static demand – Stationary MFDs

- Use MFDs to quantify performance of signal systems



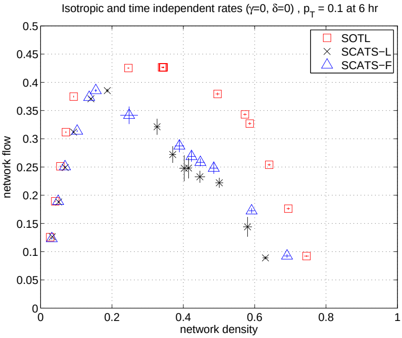
Isotropic boundary demand

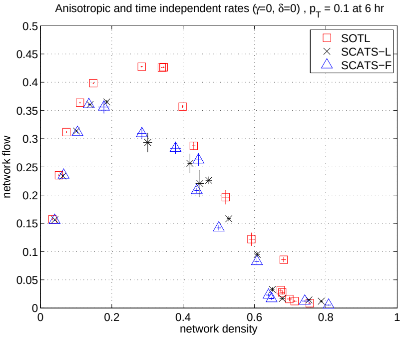


Higher demand on west side

Static demand – Stationary MFDs

- Use MFDs to quantify performance of signal systems



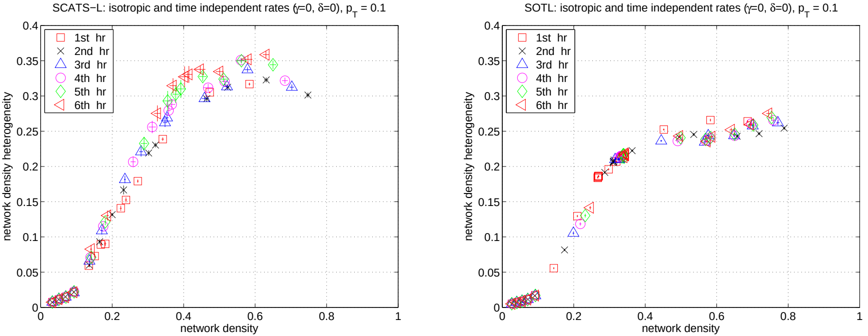


Isotropic boundary demand Higher demand on west side

- Anisotropic demand can still produce well-defined MFD

Self-organizing traffic lights

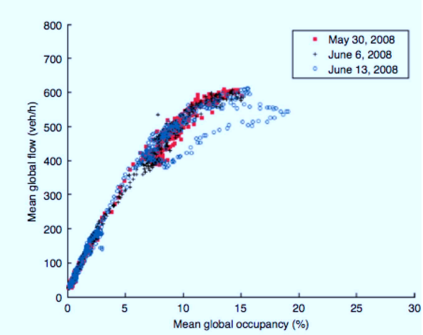
- SOTL is a toy model of a highly adaptive acyclic signal system
- Always gives green to phase with the highest demand



- SOTL has lower heterogeneity than SCATS
- Accounts for its better MFD

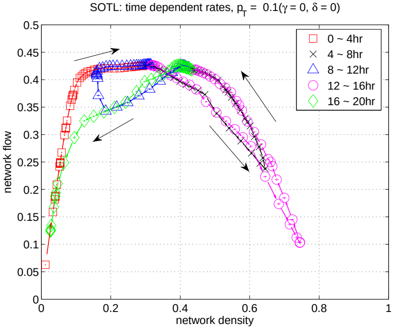
Time-dependent demand

- Vary α, β over 24 hours to mimic am/pm peaks
- Hysteresis observed - clockwise and anticlockwise



Buisson & Ladier 2009

Empirical data from Toulouse

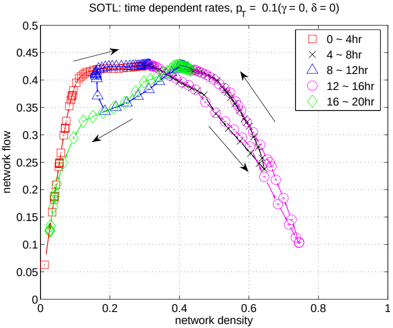


Zhang, G & de Gier 2013

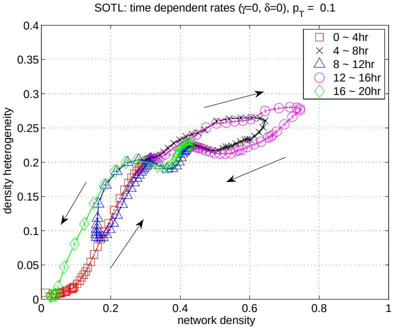
Simulated data

Time-dependent demand

- Hysteresis in MFD consequence of heterogeneity



Zhang, G & de Gier 2013
Simulated data



Zhang, G & de Gier 2013
Simulated data

Two-bin model

- Consider two adjacent networks (bins) exchanging vehicles
- Each bin has same well-defined MFD $J(\rho)$

$$\frac{\mathrm{d}\rho_1}{\mathrm{d}t} = \frac{a_1 - b_1 J(\rho_1) + p_2 J(\rho_2) - p_1 J(\rho_1)}{L_1}$$

$$\frac{\mathrm{d}\rho_2}{\mathrm{d}t} = \frac{a_2 - b_2 J(\rho_2) + p_1 J(\rho_1) - p_2 J(\rho_2)}{L_2}$$

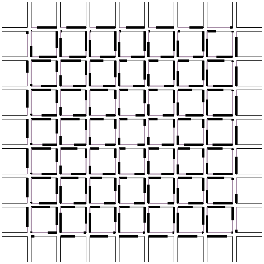
Two-bin model

- Consider two adjacent networks (bins) exchanging vehicles
- Each bin has same well-defined MFD $J(\rho)$

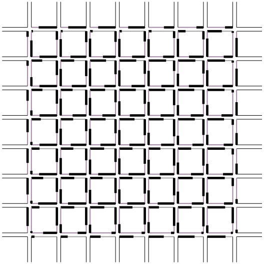
$$\frac{d\rho_1}{dt} = \frac{a_1 - b_1 J(\rho_1) + p_2 J(\rho_2) - p_1 J(\rho_1)}{L_1}$$

$$\frac{d\rho_2}{dt} = \frac{a_2 - b_2 J(\rho_2) + p_1 J(\rho_1) - p_2 J(\rho_2)}{L_2}$$

- Let bin 1 be boundary layer, bin 2 the interior



Loading



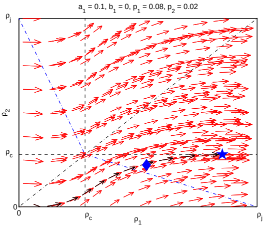
Recovery

Two-bin model

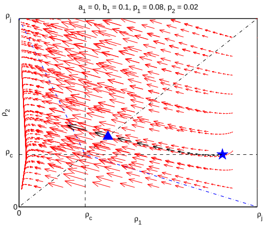
- Consider two adjacent networks (bins) exchanging vehicles
- Each bin has same well-defined MFD $J(\rho)$

$$\frac{\mathrm{d}\rho_1}{\mathrm{d}t} = \frac{a_1 - b_1 J(\rho_1) + p_2 J(\rho_2) - p_1 J(\rho_1)}{L_1}$$
$$\frac{\mathrm{d}\rho_2}{\mathrm{d}t} = \frac{a_2 - b_2 J(\rho_2) + p_1 J(\rho_1) - p_2 J(\rho_2)}{L_2}$$

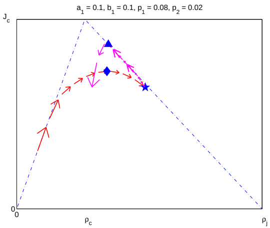
- Let bin 1 be boundary layer, bin 2 the interior



Loading



Recovery



Instantaneous MFD

Open Problems

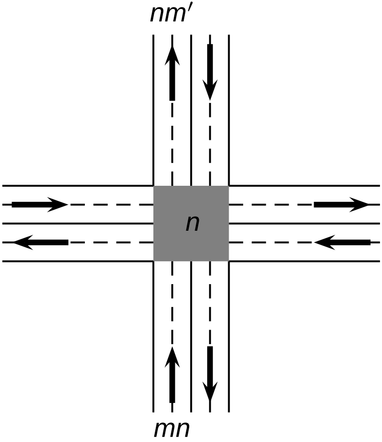
- Can we observe anticlockwise hysteresis empirically?
- Can we understand cross-correlations between flow, density and density heterogeneity?
- How does driver adaptivity affect the shape of MFDs?

Open Problems

- Can we observe anticlockwise hysteresis empirically?
- Can we understand cross-correlations between flow, density and density heterogeneity?
- How does driver adaptivity affect the shape of MFDs?
- How should one partition networks in order to produce well-defined MFDs?
- Several groups are attempting to use MFDs as a basis for perimeter control?

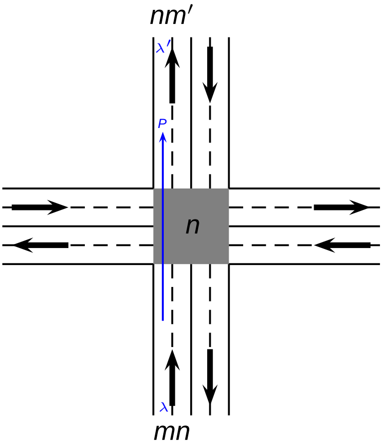
Paths

Consider a particular node n in a traffic network



Paths

Consider a particular node n in a traffic network

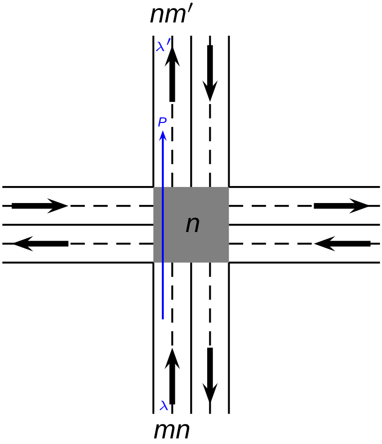


Definition

A path P is an ordered pair of lanes (λ, λ') with $\lambda \in mn$ and $\lambda' \in nm'$

Paths

Consider a particular node n in a traffic network



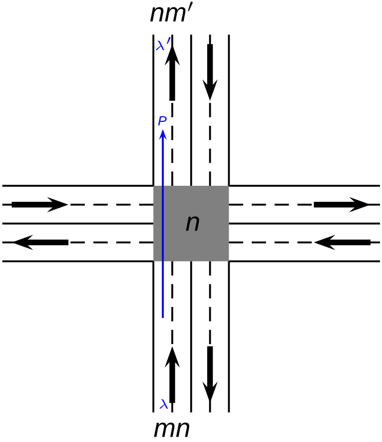
Definition

A path P is an ordered pair of lanes (λ, λ') with $\lambda \in mn$ and $\lambda' \in nm'$

- Vehicles can only move from one link to another along paths

Paths

Consider a particular node n in a traffic network



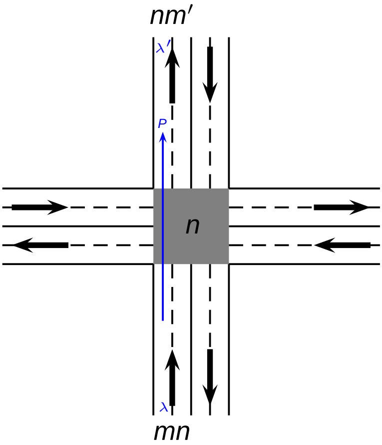
Definition

A path P is an ordered pair of lanes (λ, λ') with $\lambda \in mn$ and $\lambda' \in nm'$

- Vehicles can only move from one link to another along paths
- Ignore the actual dynamics through the intersection

Paths

Consider a particular node *n* in a traffic network



Definition

A path *P* is an ordered pair of lanes (λ, λ') with $\lambda \in mn$ and $\lambda' \in nm'$

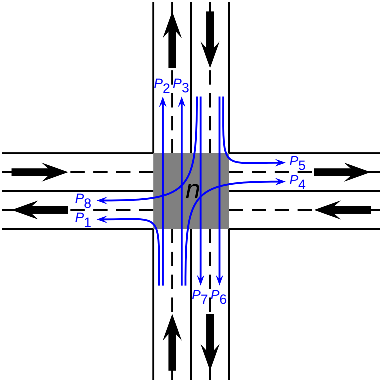
- Vehicles can only move from one link to another along paths
- Ignore the actual dynamics through the intersection
- No cells in the intersection – we use paths to glue the CA on adjacent links together

Phases

We can't simply let all paths be traversed at once – vehicles would crash inside the intersection

Phases

We can't simply let all paths be traversed at once – vehicles would crash inside the intersection

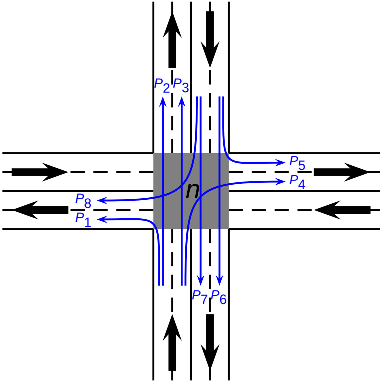


Definition

A phase $\mathcal{P}$ of node n is a subset of the paths belonging to n

Phases

We can't simply let all paths be traversed at once – vehicles would crash inside the intersection



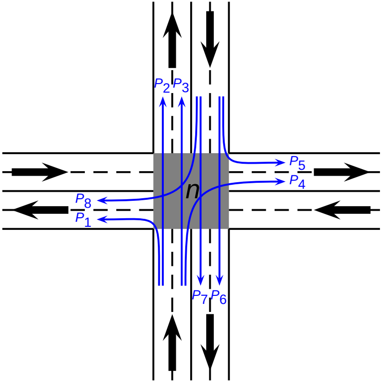
Definition

A phase $\mathcal{P}$ of node *n* is a subset of the paths belonging to *n*

- At each instant node *n* has a current phase $\mathcal{P}_{\text{current}}$

Phases

We can't simply let all paths be traversed at once – vehicles would crash inside the intersection



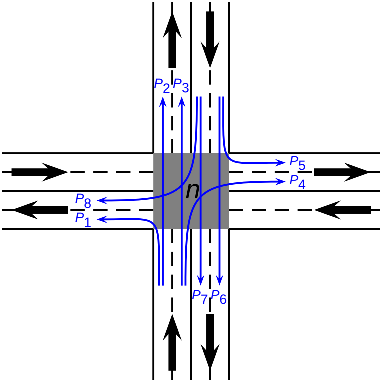
Definition

A phase $\mathcal{P}$ of node *n* is a subset of the paths belonging to *n*

- At each instant node *n* has a current phase $\mathcal{P}_{\text{current}}$
- Only paths in $\mathcal{P}_{\text{current}}$ may be traversed

Phases

We can't simply let all paths be traversed at once – vehicles would crash inside the intersection



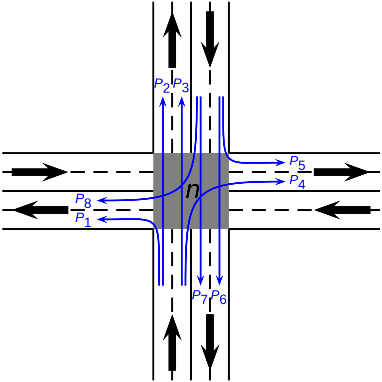
Definition

A phase $\mathcal{P}$ of node n is a subset of the paths belonging to n

- At each instant node n has a current phase $\mathcal{P}_{\text{current}}$
- Only paths in $\mathcal{P}_{\text{current}}$ may be traversed
- Implement traffic signals using phases

Phases

We can't simply let all paths be traversed at once – vehicles would crash inside the intersection



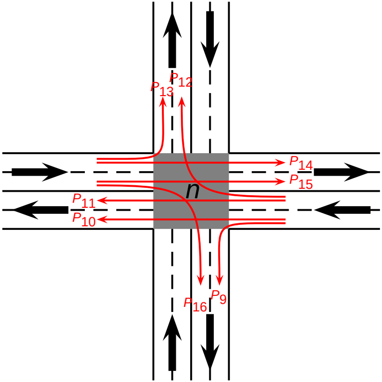
Definition

A phase $\mathcal{P}$ of node n is a subset of the paths belonging to n

- At each instant node n has a current phase $\mathcal{P}_{\text{current}}$
- Only paths in $\mathcal{P}_{\text{current}}$ may be traversed
- Implement traffic signals using phases
- Time t: $\mathcal{P}_{\text{current}} = \mathcal{P}_1 = \{P_1, \ldots, P_8\}$

Phases

We can't simply let all paths be traversed at once – vehicles would crash inside the intersection



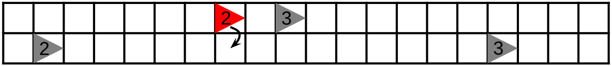
Definition

A phase $\mathcal{P}$ of node n is a subset of the paths belonging to n

- At each instant node n has a current phase $\mathcal{P}_{\text{current}}$
- Only paths in $\mathcal{P}_{\text{current}}$ may be traversed
- Implement traffic signals using phases
- Time $t + \Delta t$: $\mathcal{P}_{\text{current}} = \mathcal{P}_2 = \{P_9, \ldots, P_{16}\}$

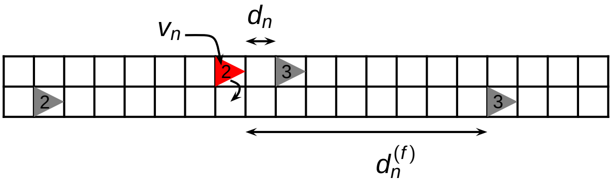
Lane changing (dynamic)

In order to model freeways or urban networks we need multiple lanes and lane changing



Lane changing (dynamic)

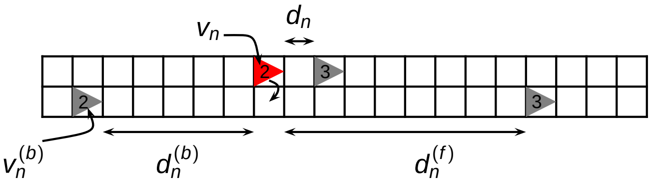
In order to model freeways or urban networks we need multiple lanes and lane changing



- If $\min(v_n + 1, d_n^{(f)}, v_{\max}) > \min(v_n + 1, d_n, v_{\max})$ the lane change is desirable

Lane changing (dynamic)

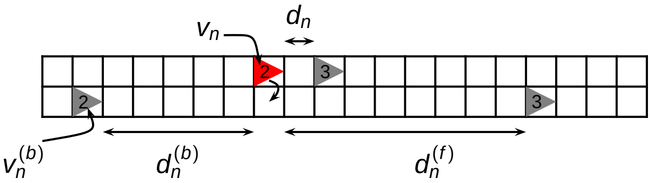
In order to model freeways or urban networks we need multiple lanes and lane changing



- If $\min(v_n + 1, d_n^{(f)}, v_{\max}) > \min(v_n + 1, d_n, v_{\max})$ the lane change is desirable
- If $d_n^{(b)} \geq v_n^{(b)}$ the lane change is safe

Lane changing (dynamic)

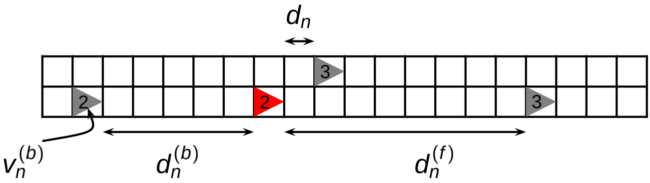
In order to model freeways or urban networks we need multiple lanes and lane changing



- If $\min(v_n + 1, d_n^{(f)}, v_{\max}) > \min(v_n + 1, d_n, v_{\max})$ the lane change is desirable
- If $d_n^{(b)} \geq v_n^{(b)}$ the lane change is safe
- If desirable and safe accept with probability p_{change}

Lane changing (dynamic)

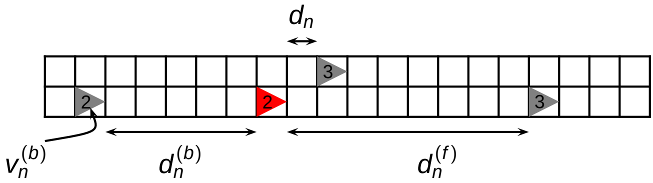
In order to model freeways or urban networks we need multiple lanes and lane changing



- If $\min(v_n + 1, d_n^{(f)}, v_{\max}) > \min(v_n + 1, d_n, v_{\max})$ the lane change is desirable
- If $d_n^{(b)} \geq v_n^{(b)}$ the lane change is safe
- If desirable and safe accept with probability p_{change}

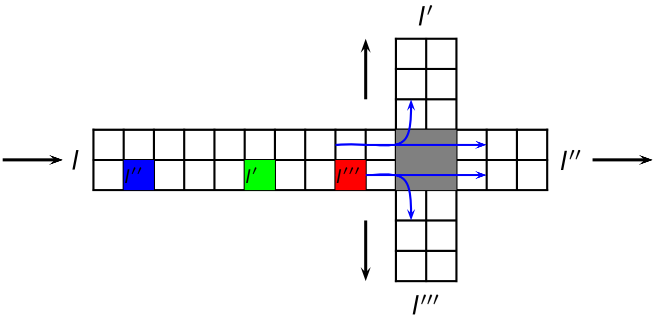
Lane changing (dynamic)

In order to model freeways or urban networks we need multiple lanes and lane changing

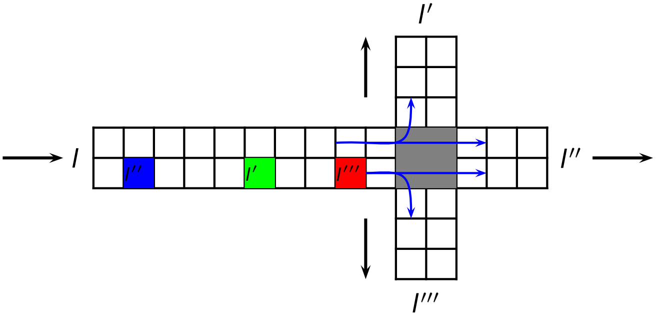


- If $\min(v_n + 1, d_n^{(f)}, v_{\max}) > \min(v_n + 1, d_n, v_{\max})$ the lane change is desirable
- If $d_n^{(b)} \geq v_n^{(b)}$ the lane change is safe
- If desirable and safe accept with probability p_{change}
- Allow only left→right (right→left) at odd (even) time steps

Lane changing (topological)

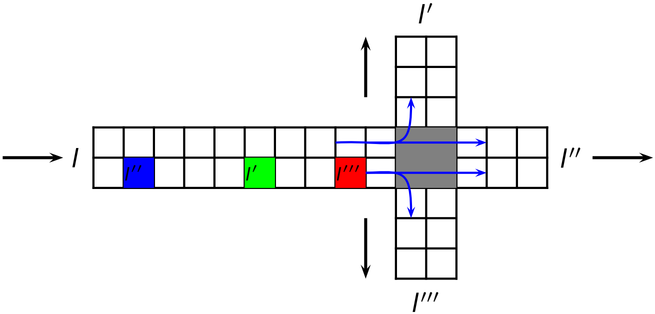


Lane changing (topological)



- Each vehicle wants to be in a lane for which there exists a path consistent with its desired turn

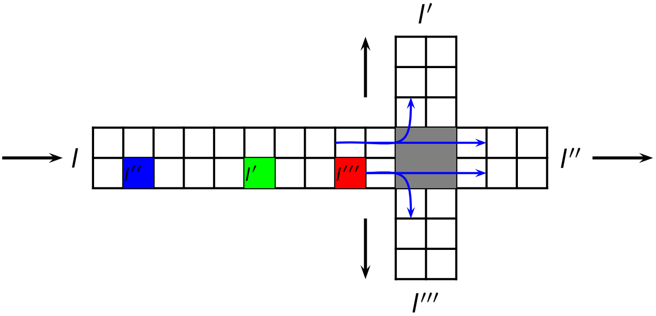
Lane changing (topological)



- Red car:
  not needed
  not allowed

- Each vehicle wants to be in a lane for which there exists a path consistent with its desired turn

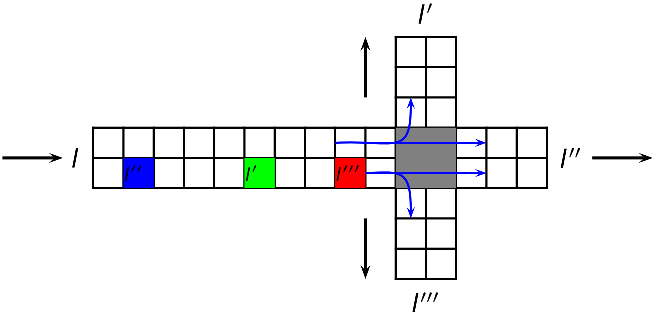
Lane changing (topological)



- Red car: not needed not allowed
- Blue car: not needed is allowed

- Each vehicle wants to be in a lane for which there exists a path consistent with its desired turn

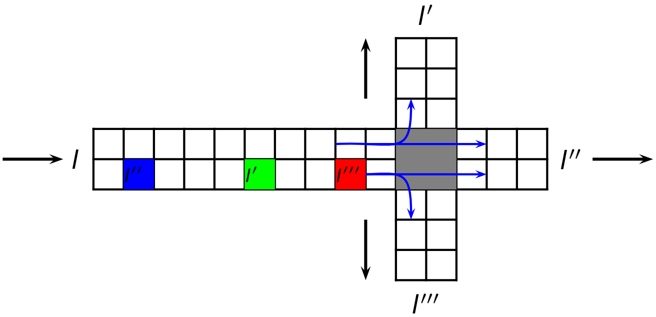
Lane changing (topological)



- Red car:
  not needed
  not allowed
- Blue car:
  not needed
  is allowed
- Green car:
  needed

- Each vehicle wants to be in a lane for which there exists a path consistent with its desired turn

Lane changing (topological)

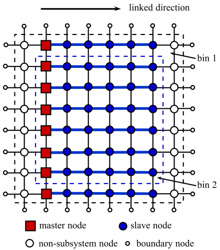


- Red car: not needed not allowed
- Blue car: not needed is allowed
- Green car: needed

- Each vehicle wants to be in a lane for which there exists a path consistent with its desired turn
- Only allow dynamical lane changing if it doesn't contradict topological lane changing – only blue car can

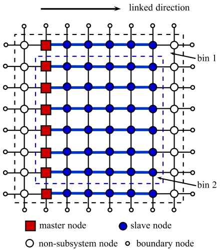
Boundaries

- We must consider open systems
- So some links only have one endpoint in the network



Boundaries

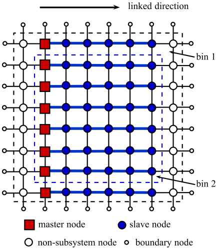
- We must consider open systems
- So some links only have one endpoint in the network



- Do not model traffic flow on boundary links

Boundaries

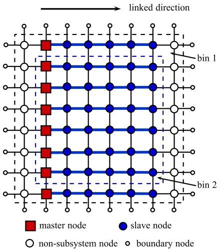
- We must consider open systems
- So some links only have one endpoint in the network



- Do not model traffic flow on boundary links
  - Each boundary lane λ has a fixed average density $\overline{\rho_\lambda}$

Boundaries

- We must consider open systems
- So some links only have one endpoint in the network



- Do not model traffic flow on boundary links
  - Each boundary lane λ has a fixed average density $\overline{\rho_\lambda}$
  - This is a boundary condition

Turning decisions

- Each vehicle should know which link it wants to turn into when it reaches the end of its current link

Turning decisions

- Each vehicle should know which link it wants to turn into when it reaches the end of its current link
- In this sense the model should be agent-based

Turning decisions

- Each vehicle should know which link it wants to turn into when it reaches the end of its current link
- In this sense the model should be agent-based
- A sophisticated approach would use origin-destination data and route planning algorithms

Turning decisions

- Each vehicle should know which link it wants to turn into when it reaches the end of its current link
- In this sense the model should be agent-based
- A sophisticated approach would use origin-destination data and route planning algorithms
- We take a simple approach

Turning decisions

- Each vehicle should know which link it wants to turn into when it reaches the end of its current link
- In this sense the model should be agent-based
- A sophisticated approach would use origin-destination data and route planning algorithms
- We take a simple approach
- For each node n, inlink $l = mn$, & outlink $l' = nm'$, we input $\mathbb{P}(l \to l') = \mathbb{P}(\text{vehicle on link } l \text{ wants to turn into link } l')$

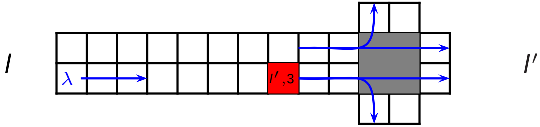
Turning decisions

- Each vehicle should know which link it wants to turn into when it reaches the end of its current link
- In this sense the model should be agent-based
- A sophisticated approach would use origin-destination data and route planning algorithms
- We take a simple approach
- For each node n, inlink $l = mn$, & outlink $l' = nm'$, we input $\mathbb{P}(l \to l') = \mathbb{P}(\text{vehicle on link } l \text{ wants to turn into link } l')$
- Turning decision made when vehicle first enters a link

Turning decisions

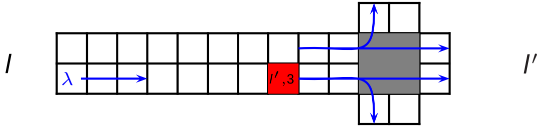
- Each vehicle should know which link it wants to turn into when it reaches the end of its current link
- In this sense the model should be agent-based
- A sophisticated approach would use origin-destination data and route planning algorithms
- We take a simple approach
- For each node n, inlink $l = mn$, & outlink $l' = nm'$, we input $\mathbb{P}(l \to l') = \mathbb{P}(\text{vehicle on link } l \text{ wants to turn into link } l')$
- Turning decision made when vehicle first enters a link
- Turning decisions affect lane changing dynamics

Mark paths



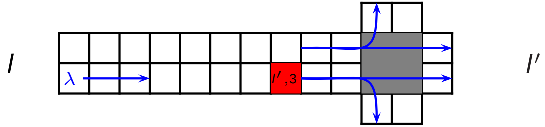
- Consider each lane λ of each link l

Mark paths



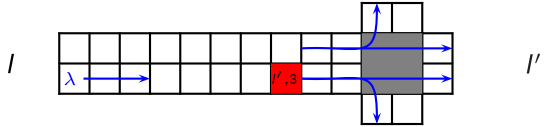
- Consider each lane λ of each link l
- Let **v** be the last vehicle on λ

Mark paths



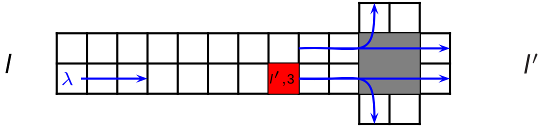
- Consider each lane λ of each link l
- Let $\mathbf{v}$ be the last vehicle on λ
- Suppose $x(\mathbf{v}) + v(\mathbf{v}) > \mathit{length}(\lambda)$

Mark paths



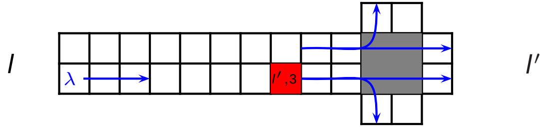
- Consider each lane λ of each link l
- Let $\mathbf{v}$ be the last vehicle on λ
- Suppose $x(\mathbf{v}) + v(\mathbf{v}) > length(\lambda)$
- If there exists $P \in \mathcal{P}_{\text{current}}$ with:

Mark paths



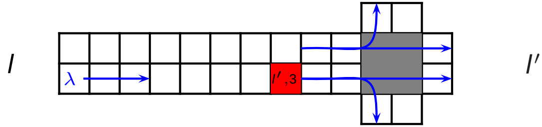
- Consider each lane λ of each link l
- Let **v** be the last vehicle on λ
- Suppose $x(\mathbf{v}) + v(\mathbf{v}) > length(\lambda)$
- If there exists $P \in \mathcal{P}_{\text{current}}$ with:
  - $inlane(P) = \lambda$

Mark paths



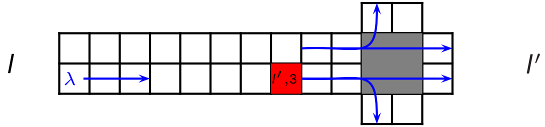
- Consider each lane λ of each link l
- Let $\mathbf{v}$ be the last vehicle on λ
- Suppose $x(\mathbf{v}) + v(\mathbf{v}) > length(\lambda)$
- If there exists $P \in \mathcal{P}_{\text{current}}$ with:
  - $inlane(P) = \lambda$
  - $outlane(P)$ has unoccupied first cell

Mark paths



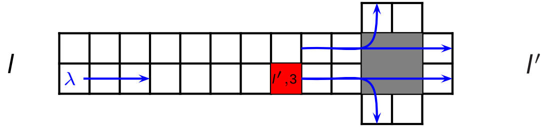
- Consider each lane λ of each link l
- Let $\mathbf{v}$ be the last vehicle on λ
- Suppose $x(\mathbf{v}) + v(\mathbf{v}) > \mathit{length}(\lambda)$
- If there exists $P \in \mathcal{P}_{\text{current}}$ with:
  - $\mathit{inlane}(P) = \lambda$
  - $\mathit{outlane}(P)$ has unoccupied first cell
  - $\mathit{outlink}(P) = \mathit{turn}(\mathbf{v})$

Mark paths



- Consider each lane λ of each link l
- Let $\mathbf{v}$ be the last vehicle on λ
- Suppose $x(\mathbf{v}) + v(\mathbf{v}) > length(\lambda)$
- If there exists $P \in \mathcal{P}_{\mathrm{current}}$ with:
  - $inlane(P) = \lambda$
  - $outlane(P)$ has unoccupied first cell
  - $outlink(P) = turn(\mathbf{v})$
- Then associate $\mathbf{v} \leftrightarrow P$ (in this case we say P is marked)

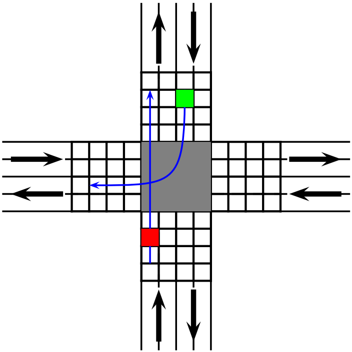
Mark paths



- Consider each lane λ of each link l
- Let $\mathbf{v}$ be the last vehicle on λ
- Suppose $x(\mathbf{v}) + v(\mathbf{v}) > \mathit{length}(\lambda)$
- If there exists $P \in \mathcal{P}_{\text{current}}$ with:
  - $\mathit{inlane}(P) = \lambda$
  - $\mathit{outlane}(P)$ has unoccupied first cell
  - $\mathit{outlink}(P) = \mathit{turn}(\mathbf{v})$
- Then associate $\mathbf{v} \leftrightarrow P$ (in this case we say P is marked)
- Else stop $\mathbf{v}$ at the end of λ

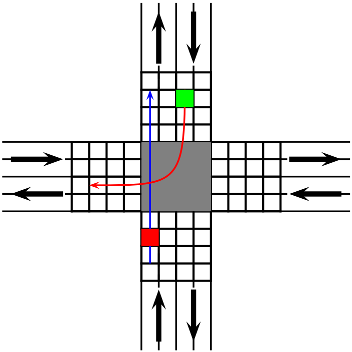
Clear paths

Consider each marked path P of each node n



Clear paths

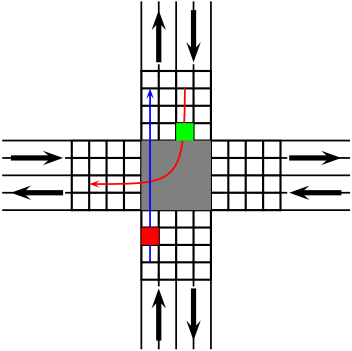
Consider each marked path P of each node n



- If P must give way to another marked path P' of n

Clear paths

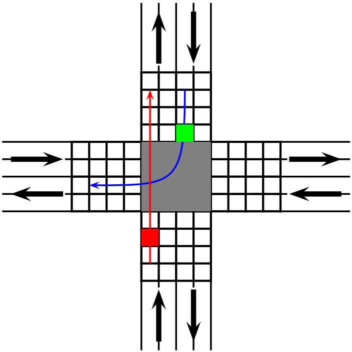
Consider each marked path P of each node n



- If P must give way to another marked path P' of n
  - Stop the vehicle $\mathbf{v} \leftrightarrow P$ on the last cell of *inlane*(P)

Clear paths

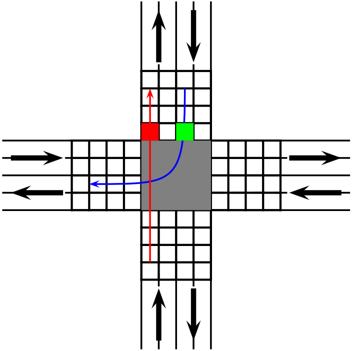
Consider each marked path P of each node n



- If P must give way to another marked path P' of n
  - Stop the vehicle $\mathbf{v} \leftrightarrow P$ on the last cell of *inlane*(P)
- Else move the vehicle $\mathbf{v} \leftrightarrow P$ to the first cell of *outlane*(P)

Clear paths

Consider each marked path P of each node n



- If P must give way to another marked path P' of n
  - Stop the vehicle $\mathbf{v} \leftrightarrow P$ on the last cell of $inlane(P)$
- Else move the vehicle $\mathbf{v} \leftrightarrow P$ to the first cell of $outlane(P)$