Figure removed: to not run into copyright issues, these figs have been deleted

V2X-based signal control

(V2X = vehicle to anything communication)

Peter Wagner, with Robert Alms, Jakob Erdmann, Yun-Pang Flötteröd, and Daniel Wesemeyer

German Aerospace Center (DLR) – Institute of Transport Systems

Transport Phenomena in Complex Environments 2019

Erice, Sicily, Italy

5 September 2019

DLR

So far, during this school…

Figure removed

- Most talks dealt with small things, and how they move
- This session deals with larger things (vehicles), moving in a yet larger structures: networks (this talk).
- I will only mention the modelling vehicles
- A network (links & nodes) needs simple vehicle objects:
    - TASEP has been mentioned several times, a good candidate for the dynamics of cars on a link
    - One may do it even simpler, by just counting: queue-models
    - Or more complicated, by doing a real vehicle dynamics $v_i(t + \Delta t) = \cdots$

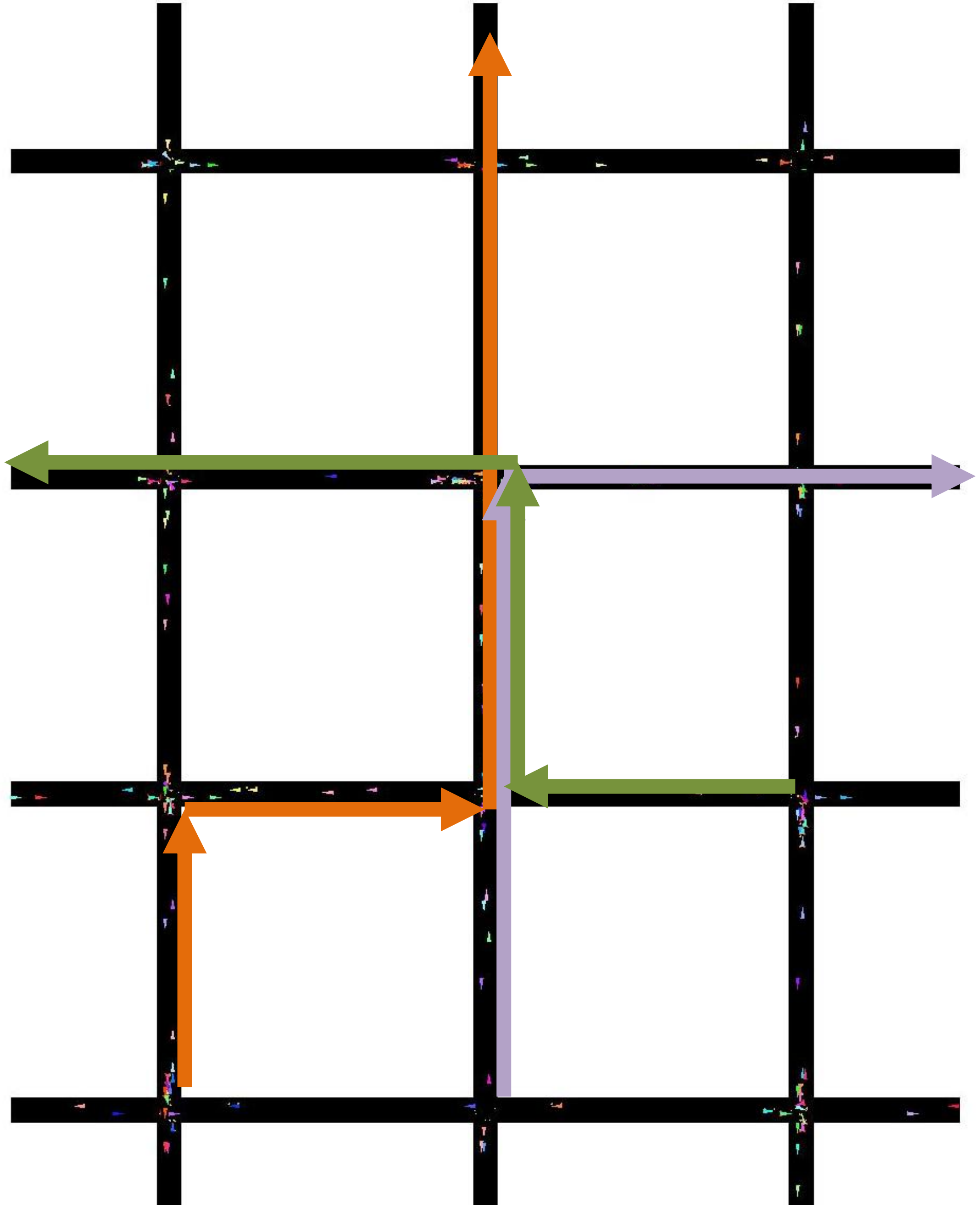
Figure removed

Figure removed

Real networks…

- Have something that we haven't meet this far: the objects follow real routes
- ➔ creates spatio-temporal correlations
- Which are important if it comes to the co-ordination of traffic signals in such a system.

- Apart from this, most of the presentation will be simple; I assume that there are also many experts in this room.
- Vehicle drivers.



This talk

1. Introduction
2. Local control
3. Networks
4. Conclusions

Revolves around:

- What can be gained in traffic signal networks?
- There is an important distinction between ideal and real networks/ objects.
- Simulation models catch some of this difference. Hopefully.

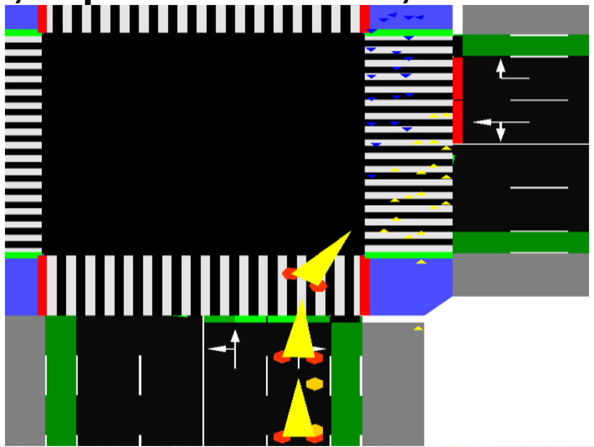
Question to the experts

- Any idea how real traffic signals in cities are organized?
- Physicists are good observers: If walking through a city, can you tell apart well organized from a badly organized signals?

Introduction for the audience not in traffic engineering

- V2X: Communication between vehicles (V) and anything else (X), especially traffic signals (TLS)
- Announced at least since 2005 (when I first became aware of it), still no large-scale implementation (to my knowledge)
- Traffic signals (TLS) are an important part of infra-structure in city traffic. Why TLS?
- TLS produce delay (delay: difference between real and ideal travel times)
- Finally: simulation. Will use here our own tool named SUMO; open source, and can simulate most traffic objects microscopically.
- See https://sumo.dlr.de

SUMO
SIMULATION OF URBAN MOBILITY



SUMO – a step towards (more) reproducible traffic science?

Figure removed: If you want reproducible science, the software needs to be open source

SUMO
SIMULATION OF URBAN MOBILITY

eclipse

DLR

V2X-based signal control – Why?

- Why is it interesting?
- Simple: this is the input needed to do it optimally.
- (Or close to optimal.)

- Vehicles communicate with the TLS controller via (4G), 5G, G5,…

➔ TLS can compute the best possible plan.

- One intersection; does it work with many?
- That is what we want to find out…

Source: DLR

Well – this did not work as planned

- There is always a danger with field experiments: they took longer.
- ➔ So, I can only report on simulations, and on older (sets of) field experiments that had just one intersection
- But: I use this opportunity to talk about the general framework
- Big question: what can we reach with traffic signal optimization in real networks?
- By what means?
- And, is it worth the effort?

- (My boss thinks not…)

Single intersections and small nets

Controlling TLS

- Very old: fixed cycle (1927)

Fixed cycle

- There is a well-known theory from 1958 (Webster, a physicist) that tells how to organize a traffic signal optimally in a fixed cycle manner
- He derived two approximations:
- Optimum cycle time c, it depends on demand q_i, more precisely on the ratio between demand and saturation s_i of all phases $y_i = \frac{q_i}{s_i}$, and the loss time L:
- $c = \dfrac{1.5L + 5}{1 - \sum_i y_i} = \dfrac{1.5L + 5}{1 - Y}$
- The green times g_i are then given by $g_i = (c - L)\frac{y_i}{Y}$
- Optimal (fixed cycle) for one intersection, constant demand, Poisson arrivals

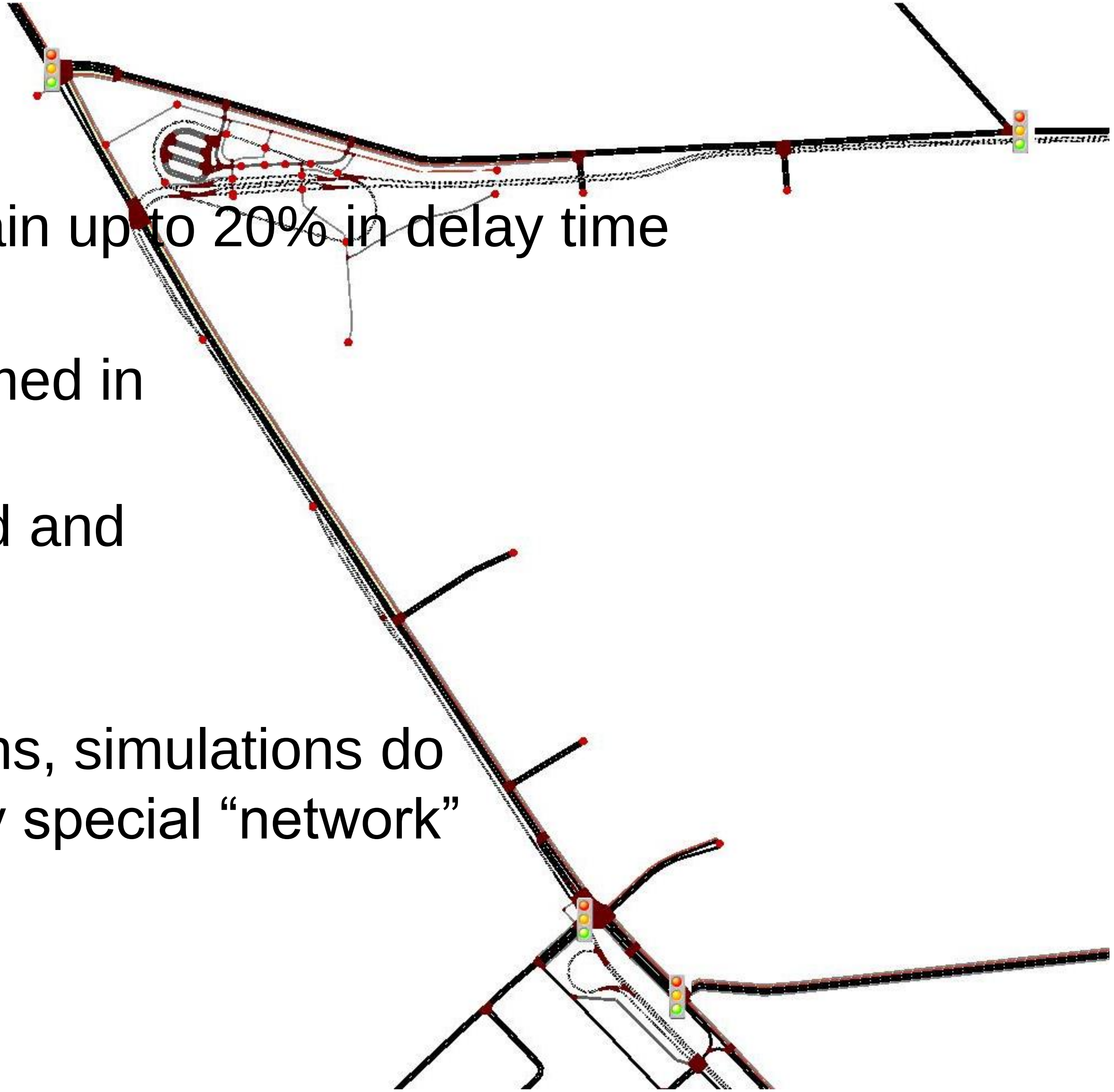
Controlling TLS

- Very old: fixed cycle (1927)
- Old: traffic controls TLS (1928 based on horn, 1952 like today)
- Actuated control: if the time since the last passing vehicle has grown too large, end this green phase
- Delay-based: communicating vehicles can tell TLS their speed
  ➔ $d_i = \Delta t\,(1 - v_i/v_{\max})$: when $\sum_i d_i = 0$, end green
- Make optimum plan based on communicated arrival times (dynamic programing) for the next ~60+ seconds. AGLOSA
- Update plan moving horizon (event-based, or any ~15 seconds)
- This last one is arguably the best, should be close to optimal. Robustness?

Figure removed

Photo: Charles Adler, Jr. Collection/ Archives Center/National Museum of American History, Smithsonian Institution
Blown Away: Adler's horn-activated traffic signal was quickly eclipsed by a pressure sensor embedded in the road.

Tested in simulation and field



- At one intersection, these two methods gain up to 20% in delay time
- In simulation, as well as in reality
- But: in one example, AGLOSA out-performed in simulation any other method
- In the field, the two methods (Delay-based and AGLOSA) have been about equal

- Preliminary results: With three intersections, simulations do not indicate large gains – but this is a very special “network”

- (And a lot of politics…)

Larger networks

Traffic signals in a network

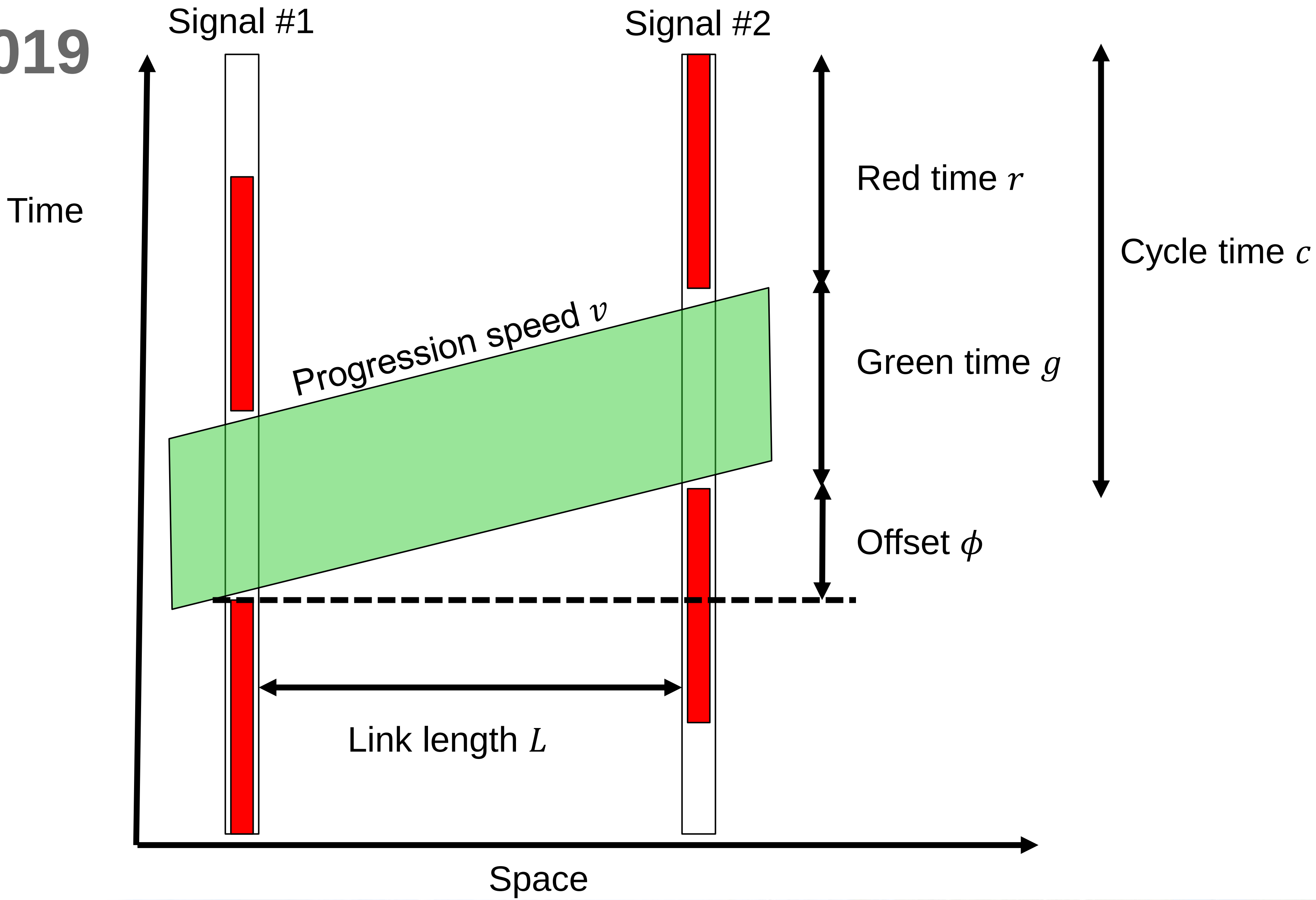
- Car-drivers: traffic signals do always display red when I arrive there
- To remedy this, traffic signal co-ordination (progression) is attempted
- Most famous: the green wave
- Easy to understood: in a space-time diagram, a platoon of vehicles progresses from one traffic light to the next

Figure removed

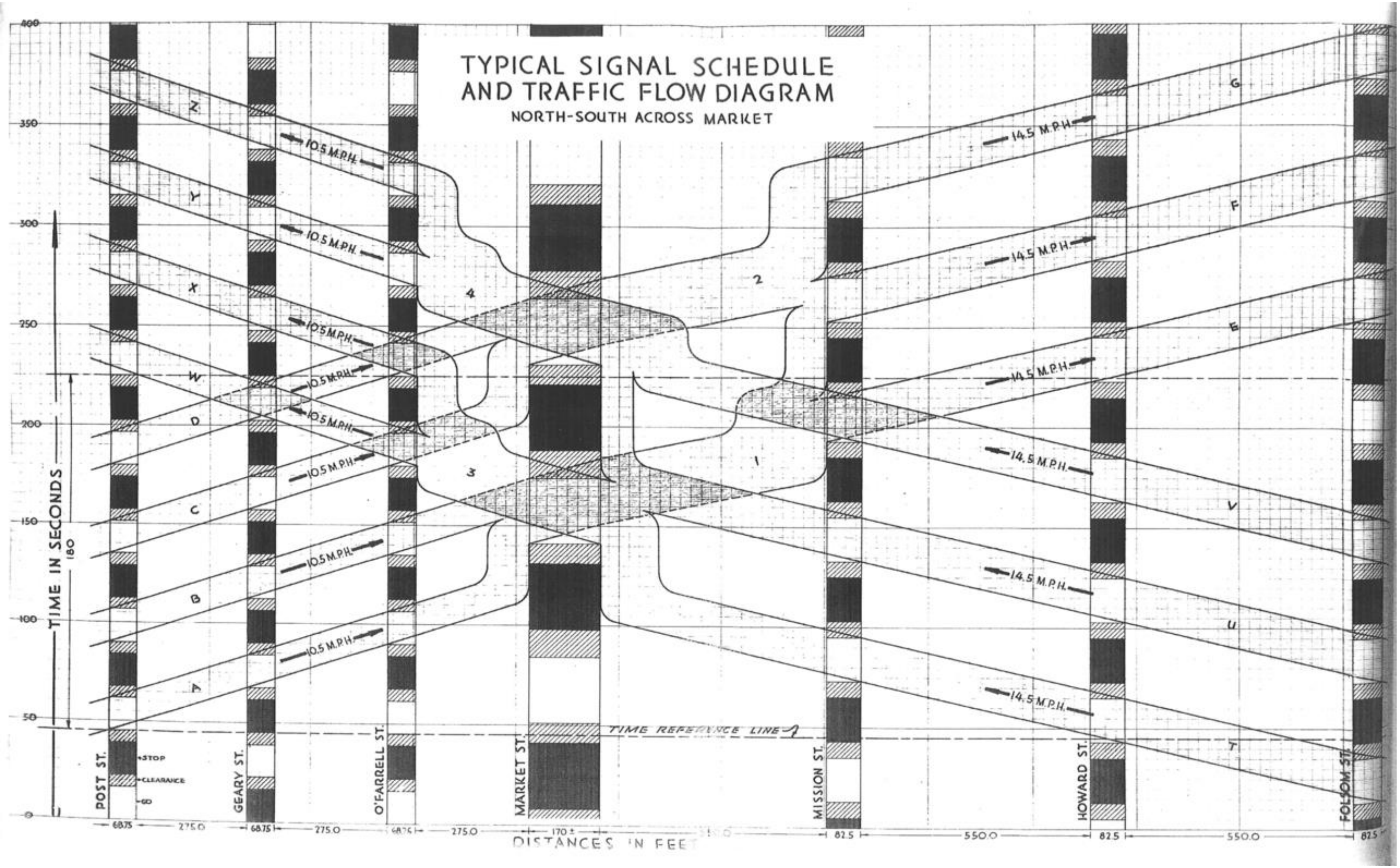
https://www.fhwa.dot.gov/publications/publicroads/02janfeb/timing.cfm

A green wave 2019

- Note the variable offset ϕ; the phase difference between each two “oscillators” (traffic lights) that run with phases φ_i, φ_j
- $\phi_{ij} = \varphi_i - \varphi_j$
- Clearly, in the best of all worlds $\phi = T = \frac{L}{v}$
- T is travel time



1929:



By City of San Francisco - Public domain (via Eric Fischer), CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=34715929

Introduction: Traffic signals in a network

- Car-drivers: traffic signals do always display red when I arrive there
- To remedy this, traffic signal co-ordination is attempted
- Most famous: the green wave
- Easy to understood: in a space-time diagram, a platoon of vehicles progresses from one traffic light to the next
- And: you may achieve the optimum: delay = 0 ☺ (makes a fine test case, will resort to this several times)
- Unfortunately easy to understood: one may think that doing the same in networks is simple, too.
- Not true, of course

Figure removed

https://www.fhwa.dot.gov/publications/publicroads/02janfeb/timing.cfm

Extension to a network…

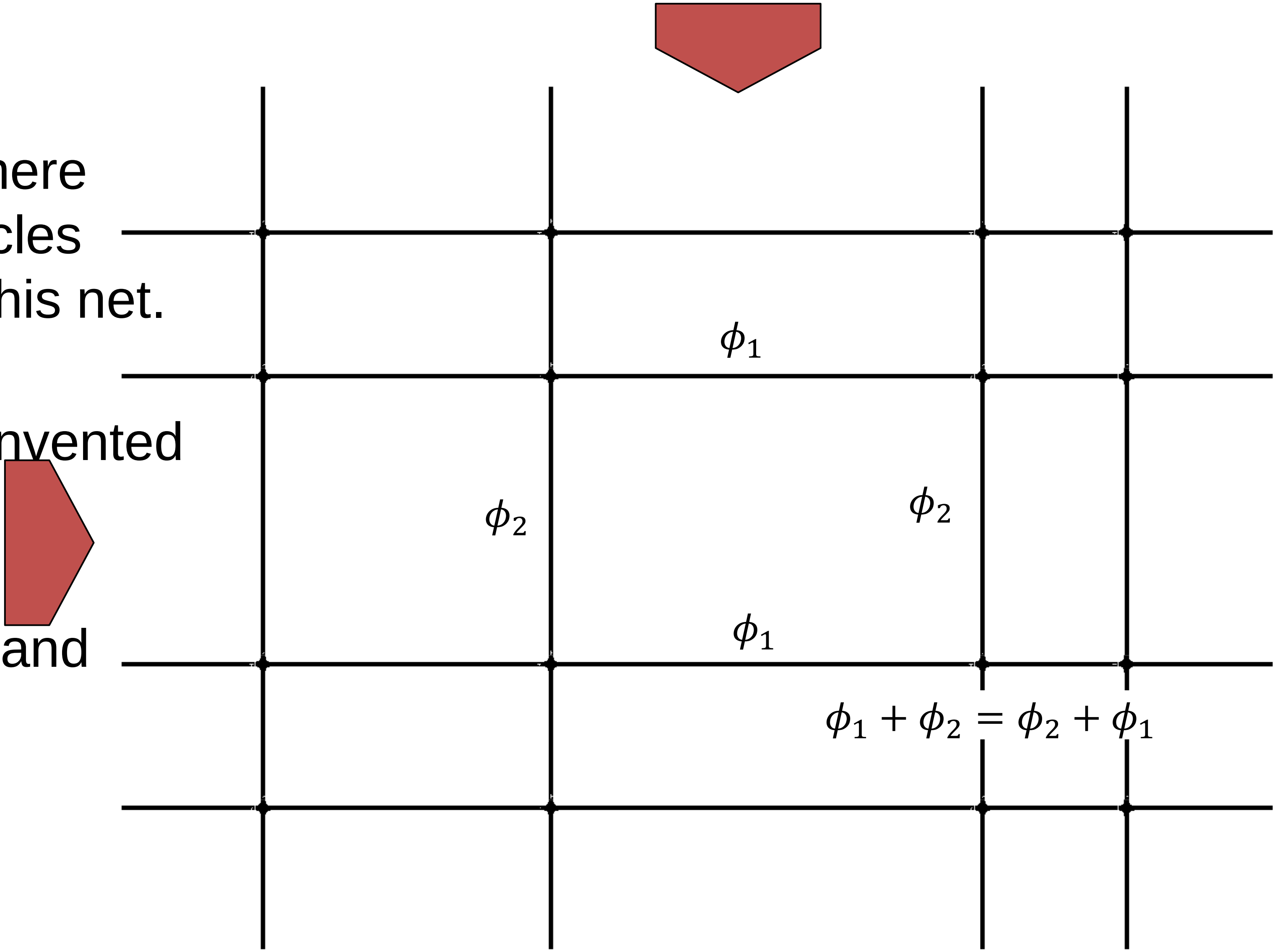
- Is complicated, only in rare special cases (regular grid networks, other preliminaries) this can be done in a simple manner
- (Even a green wave in both directions is generally not possible)
- In real networks, this runs into a fairly complicated optimization problem which is, as far as I have understood, NP-complete to solve
  (Little, 1966), (Gartner, Little, & Gabbay 1977)
- In 2004, Carlos Gershenson started a hype with the idea of a self-organized traffic signal system (SOTL)
- There is a lot of additional work on this
- Idea is: let these signals alone, together with the appropriate control mechanism they will find some self-organized optimum

Some kind of irony

Figure removed

- Carlos used one of those theory-things that especially physicists love: grid city, traffic flows in two directions only
- System can be open or closed (periodic boundary)
- SOTL is in essence:
- When red, do cumulative count n of vehicles on link
- If $n > \theta$ then switch (and reset $n = 0$) (provided minGreen has been reached)
- Funny: has an exact optimum solution! Not sure that he was aware of this, at least the paper does not mention it, but:
- Directions are independent; even inhomogeneous grids always have a set of offsets so that a perfect green wave can be established in both directions.

- Except at the input edges, where delays are unavoidable, vehicles can run unimpeded through this net.
- SOTL is similar to a method invented by Dunne & Potts in 1964
- D & P counted delayed cars, and were concerned with single intersection only



The Great Plan

- SOTL draw criticism. Nicely put by Bernhard Friedrich where he challenged
- The “jungle principle” with “The Great Plan”
- A Great Plan is charming, too: such a plan (similar to a bus schedule) forces traffic flow into a pattern of platoons for which down-stream traffic signals can be timed optimally
- Traffic is organized by the plan laid out by the traffic management center

Figure removed

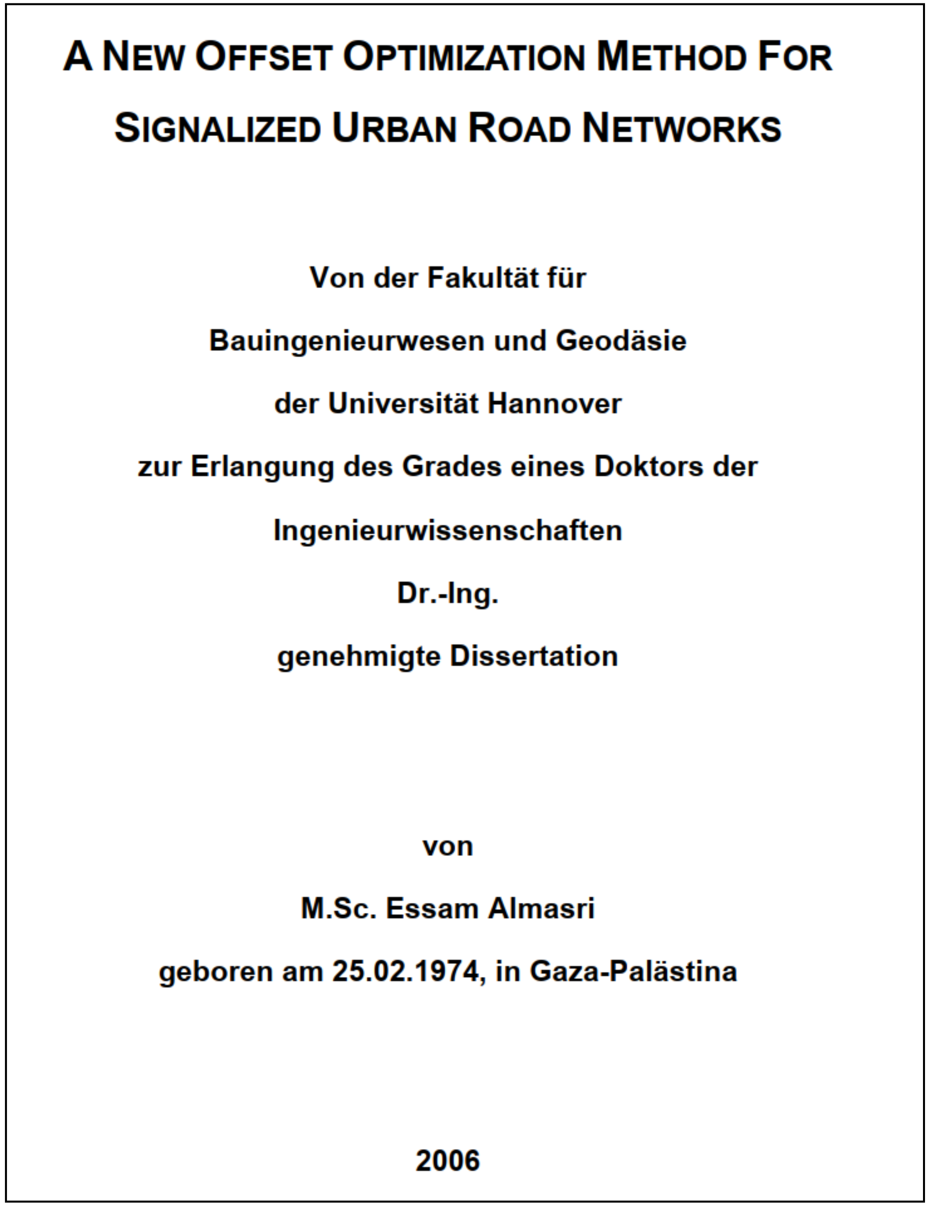
From: https://www.athenstransit.org/lines-5-6/

The Big Question

- What is better?
- Or, once more: what can be achieved?
- And under which conditions/ circumstances?

Do you know Essam Almasri's PhD?

- To find the optimal solution in a network one needs to find the optimal set of offsets ϕ_i
- This is a nasty optimization problem
- For small networks, brute-force is a temptation:
- System with 6 intersections has 5 offsets; a cycle time $c = 90\ s$ and test with 5 s granularity: $\left(\frac{90}{5}\right)^5 \approx 2M$ simulations
- That he did, with a CTM



A NEW OFFSET OPTIMIZATION METHOD FOR SIGNALIZED URBAN ROAD NETWORKS

Von der Fakultät für
Bauingenieurwesen und Geodäsie
der Universität Hannover
zur Erlangung des Grades eines Doktors der
Ingenieurwissenschaften
Dr.-Ing.
genehmigte Dissertation

von
M.Sc. Essam Almasri
geboren am 25.02.1974, in Gaza-Palästina

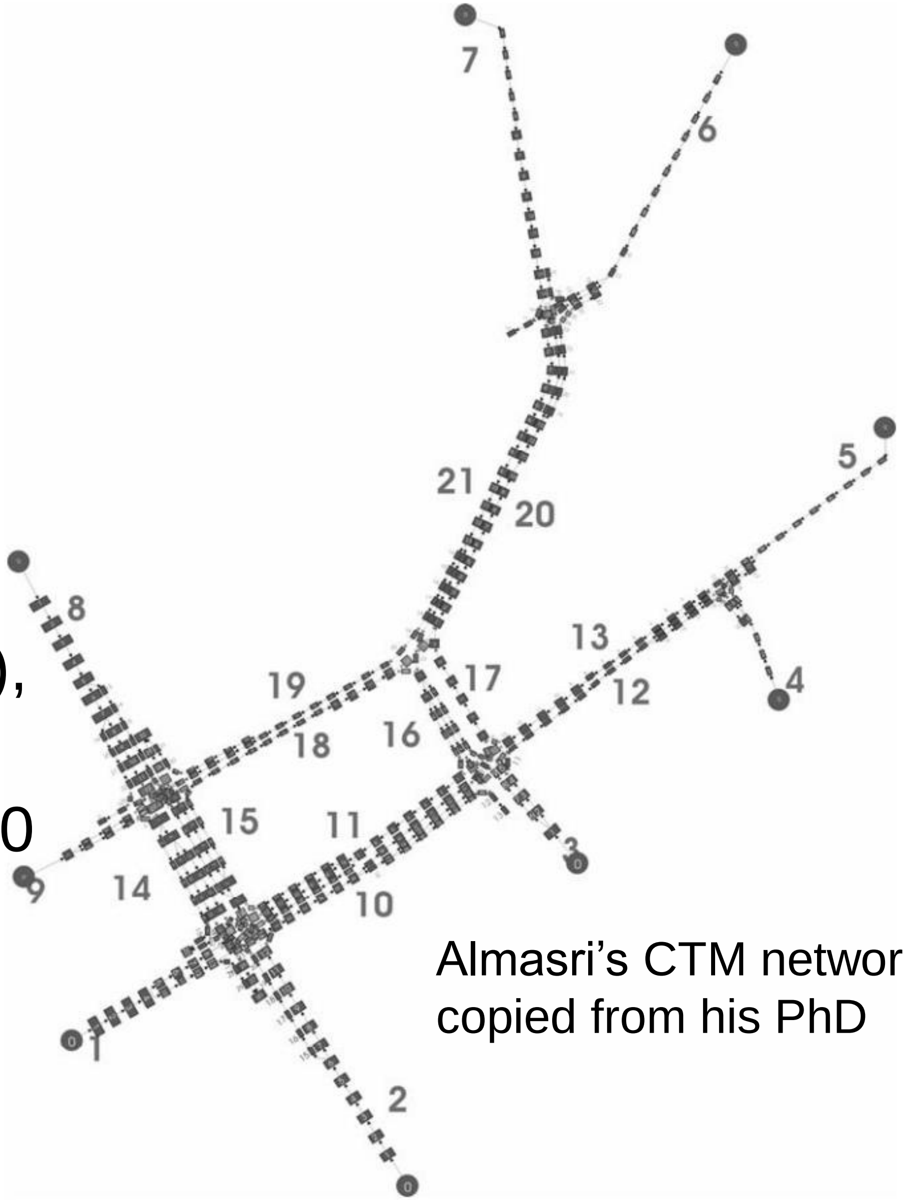
2006

Slightly less brute force

- Bad: this was for one demand only ☹
- In n-D this type of brute-force is not the best to integrate higher dimensional functions

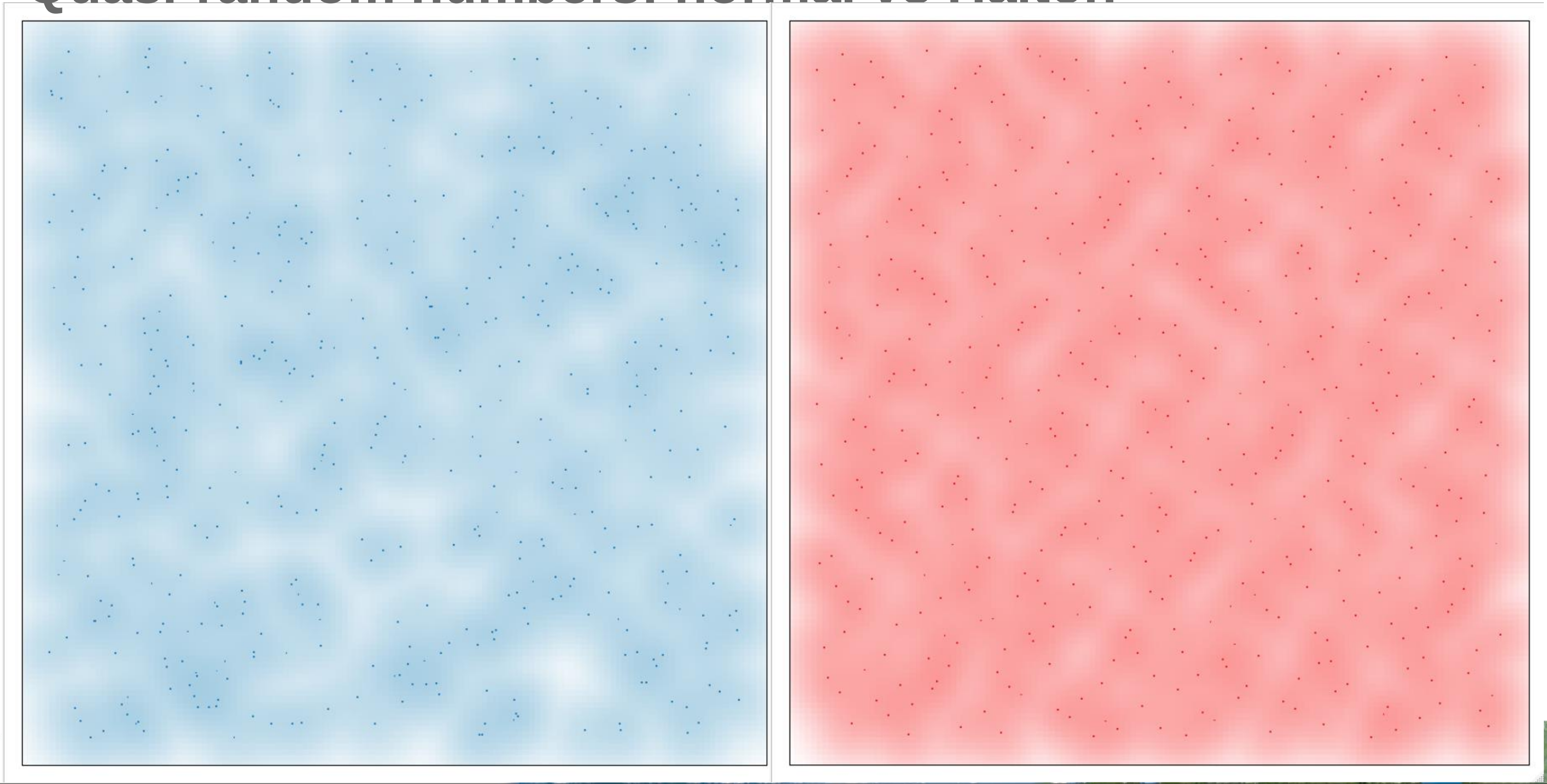
➔ Quasi-random numbers are a better way to do it

- Cover n-D spaces with minimal holes (discrepancy), and therefore, one has a better level of control
- Are scalable: if you have computer time to run 1,000 simulations, then you just compute 1,000 quasi-random n-D-tupels for your problem.
- That is what we have done



Almasri's CTM network, copied from his PhD

Quasi-random numbers: normal vs Halton



Simulation speed

- What is the fastest way to compute such a scenario? I.e., a small network with a
  - Given demand pattern
  - Cycle time
  - Set of offsets; green-times are computed from demand, they are not variables
- Almasri did it with the CTM; there is a believe that this is the fastest possibility
- But: no OD and trips, CTM has to run with $\Delta t = 1s$ to use traffic signal control

Also fast:

- Queue-model,
- A real microscopic model,
- Truly hard-core: single-bit coding of TASEP

Simulation speed

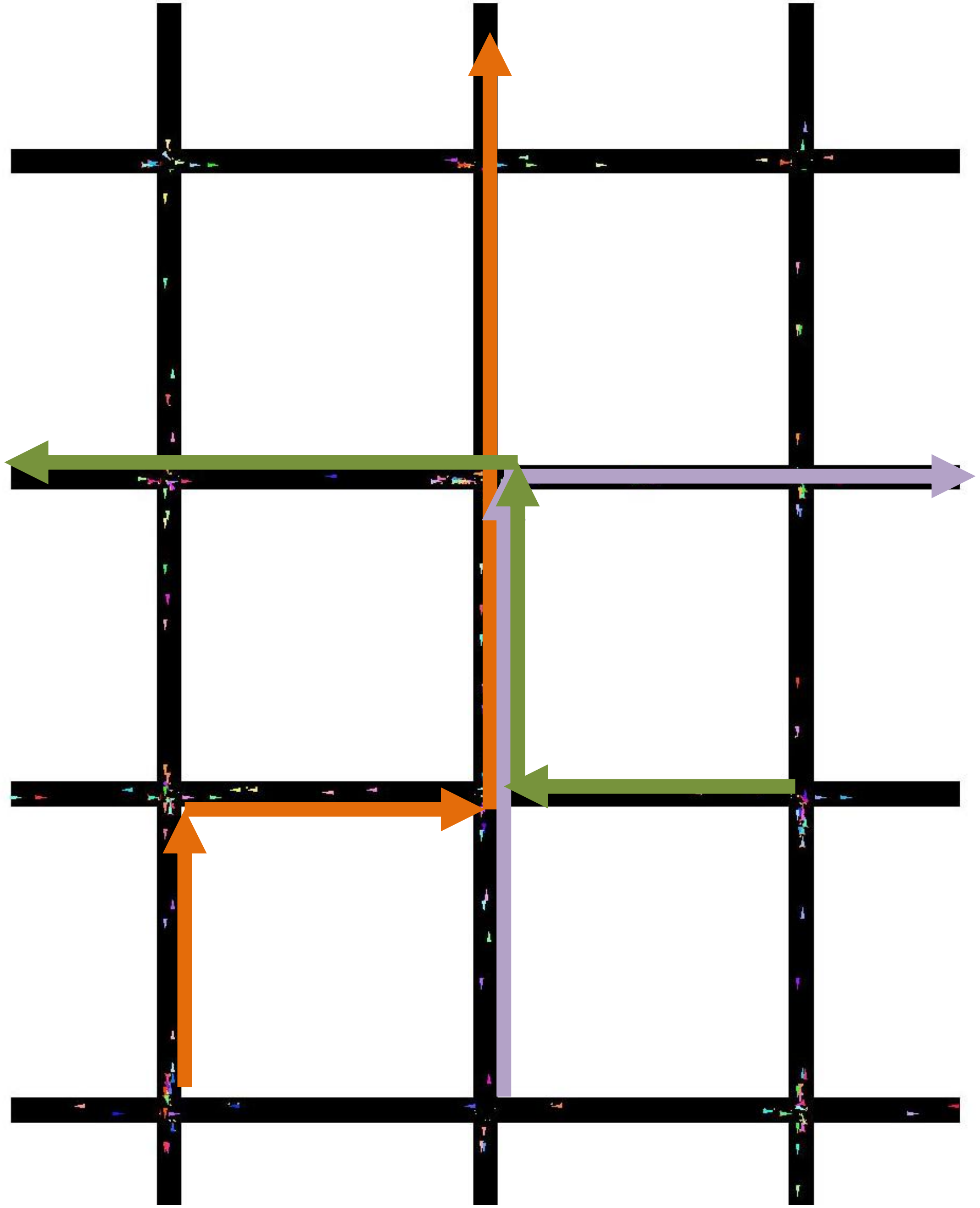
- SUMO with Almasri’s network:
  3h real time = 1 s sim time, about 1,000 trips/s
- This is the metric for comparison: trips/s
- A microscopic implementation tweaked for speed tops at 1 M trips/s
- Queue-models can do 20M trips/s; a serious implementation is close to 6M trips/s and is on par with the CTM
- Finally, the single-bit coding is still faster, but a pain to work with

- (Even programming Almasri’s network is stupid monkey work)
- Cannot be generalized…

More systematically, less thorough

Simulated Worlds (Parts of Cities)

- Real networks have both directions
- Have cars, and not green-bands
- These cars have different speeds ➔ platoon dispersion
- And: they have real routes, which interfere with Great Plan



The most import things last

- The networks have two lanes in each direction, that was done intentionally
- Cars are identical, but their preferred speed is drawn from a distribution with `speedDev = 0.1`
- Vehicles drive stochastically, parameter sigma of the SK model is at SUMO’s default value (0.5)

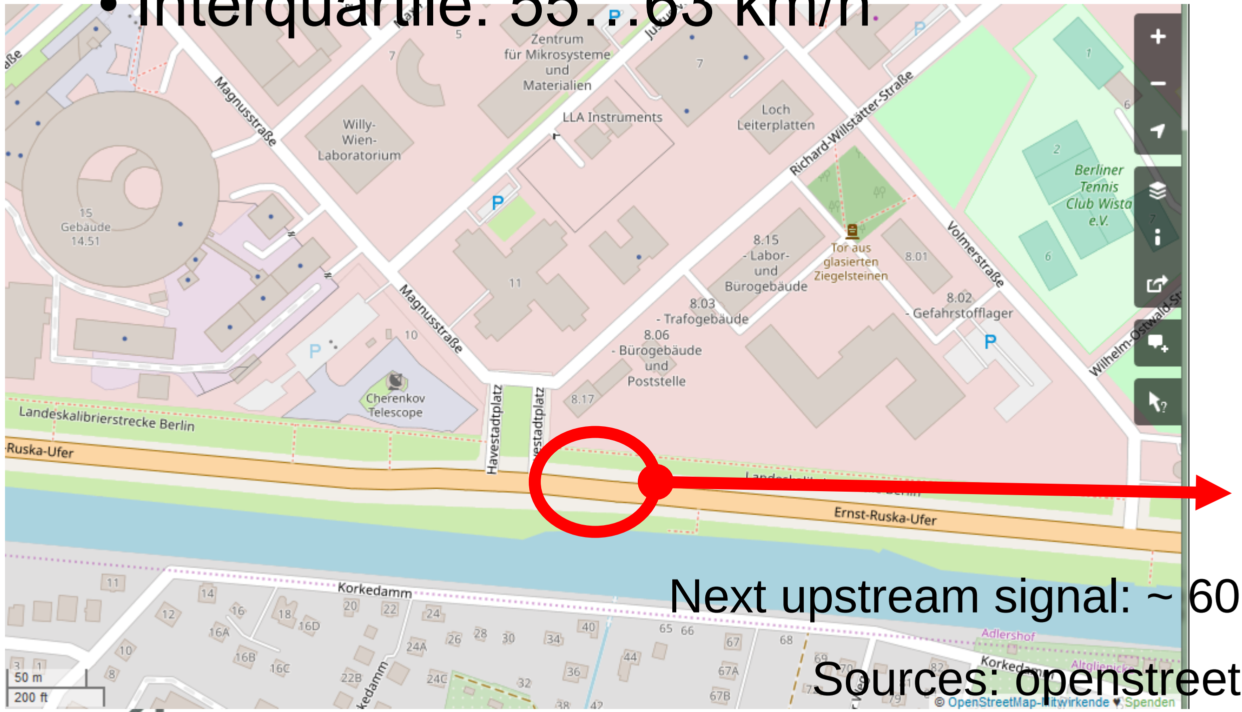
➔ Strong platoon dispersion, not unrealistic:

Figure removed

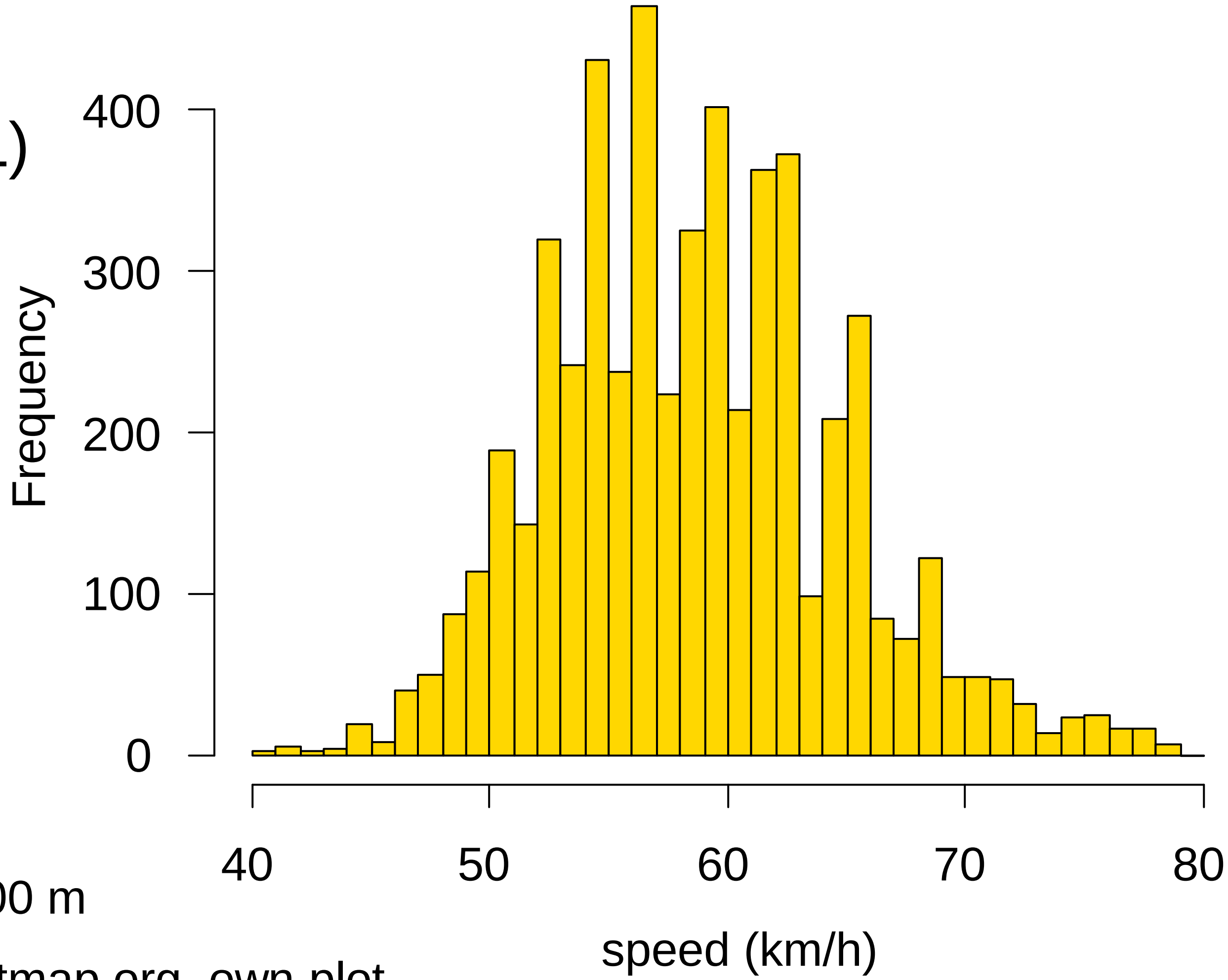
Taken from Gartner, Little, & Gabbay

Real-life speed distribution (Ernst-Ruska-Ufer, 2015)

- Data between 20…80 km/h (138 max!)
- Mean = median = 59 km/h (50 km/h SL)
- Sd = 6 km/h ➔ `speedDev = 0.1`
- Interquartile: 55…63 km/h



Next upstream signal: ~ 600 m



Sources: openstreetmap.org, own plot

Simulated Worlds (Parts of Cities) II

- All intersections have traffic lights
- All scenarios are grid-based, but with inhomogeneous grids
- Three main methods:
  - The Great Plan (in three versions)
  - Local control only (two versions, delay and actuated)
  - Local control with prediction (AGLOSA)
  - None

“SOTL light”

Gershenson/ SOTL

- Metrics for demand and delay in networks with different sizes:
  - Demand is inserted vehicles / network size (usually a.u.)
  - Delay is in seconds per vehicle per kilometer

Great Plan

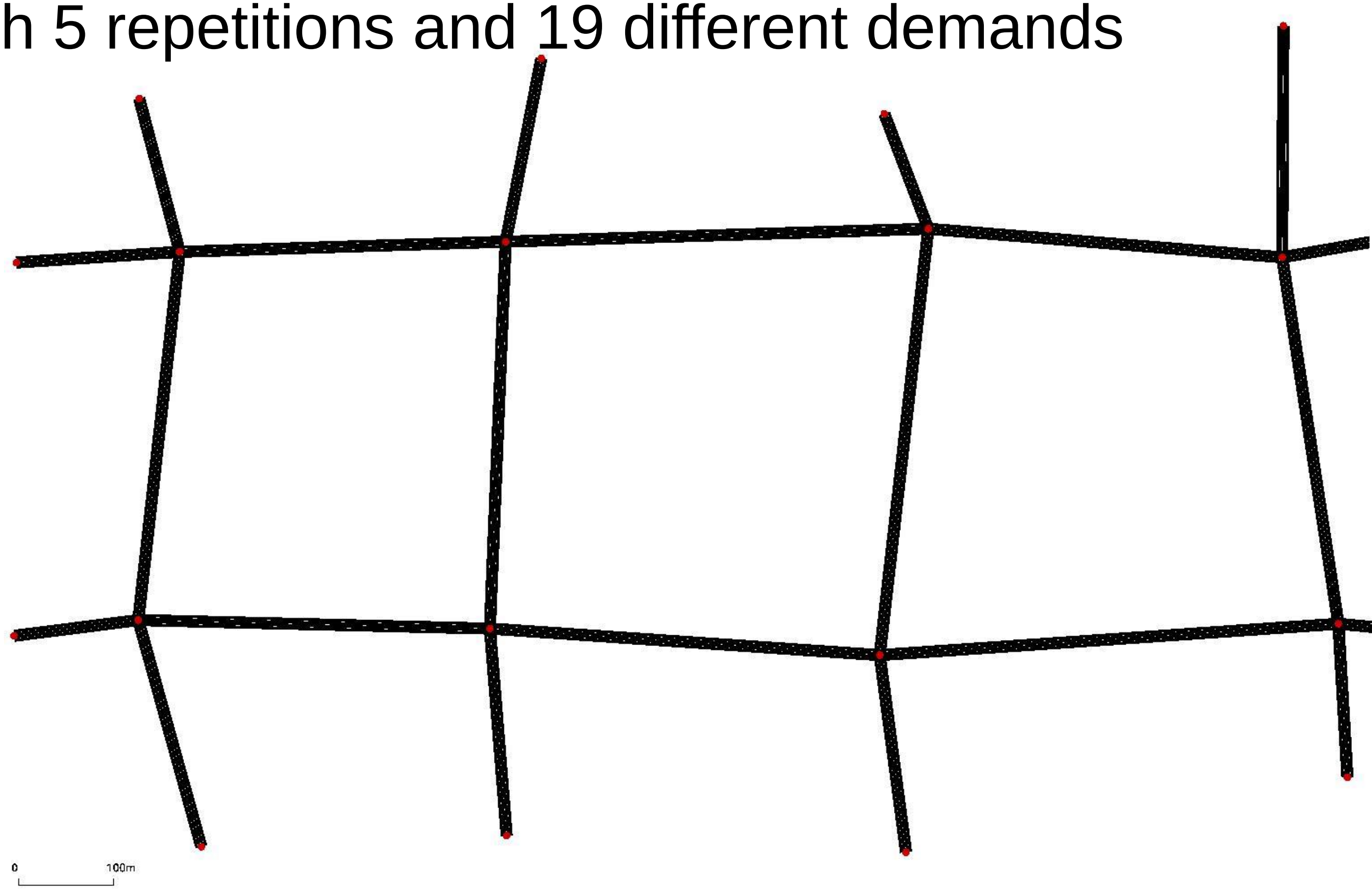
- All TL are fixed cycles:
- SC: compute optimal splits (green times) and cycle times for each intersection
- (based on Webster's theory)
- This depends on the demand at each intersection
- SCO: add co-ordination to this

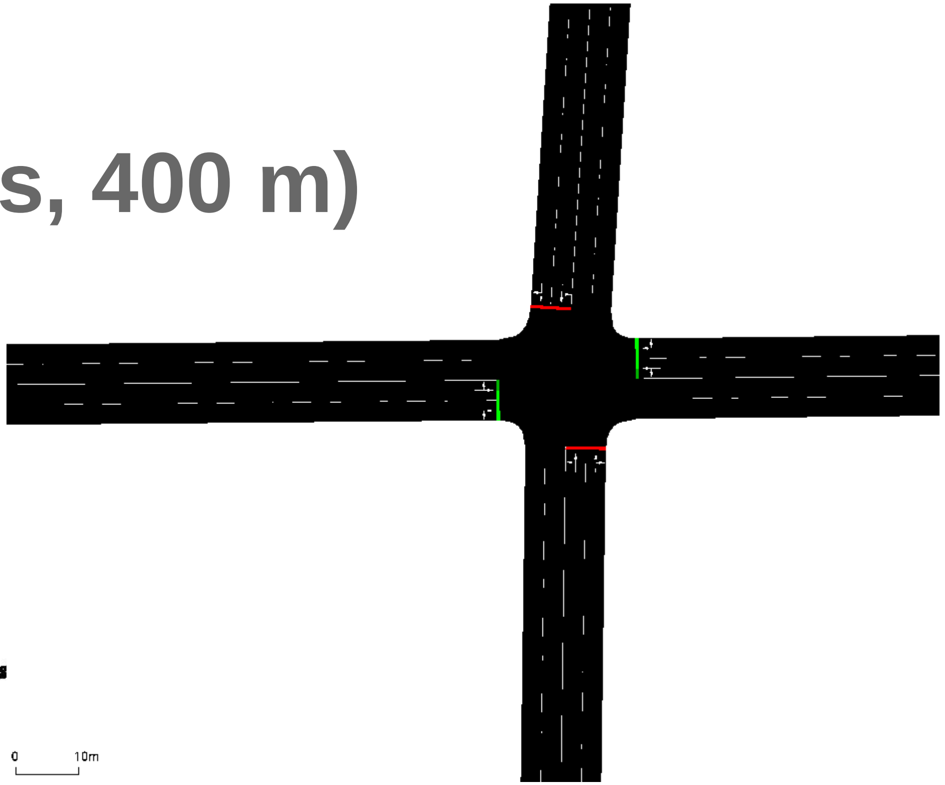
- Sometimes: use SUMO's default as comparison (it is a worse solution, since it does not know anything about demand)

Results

Example of a network (disturbed grids, 400 m)

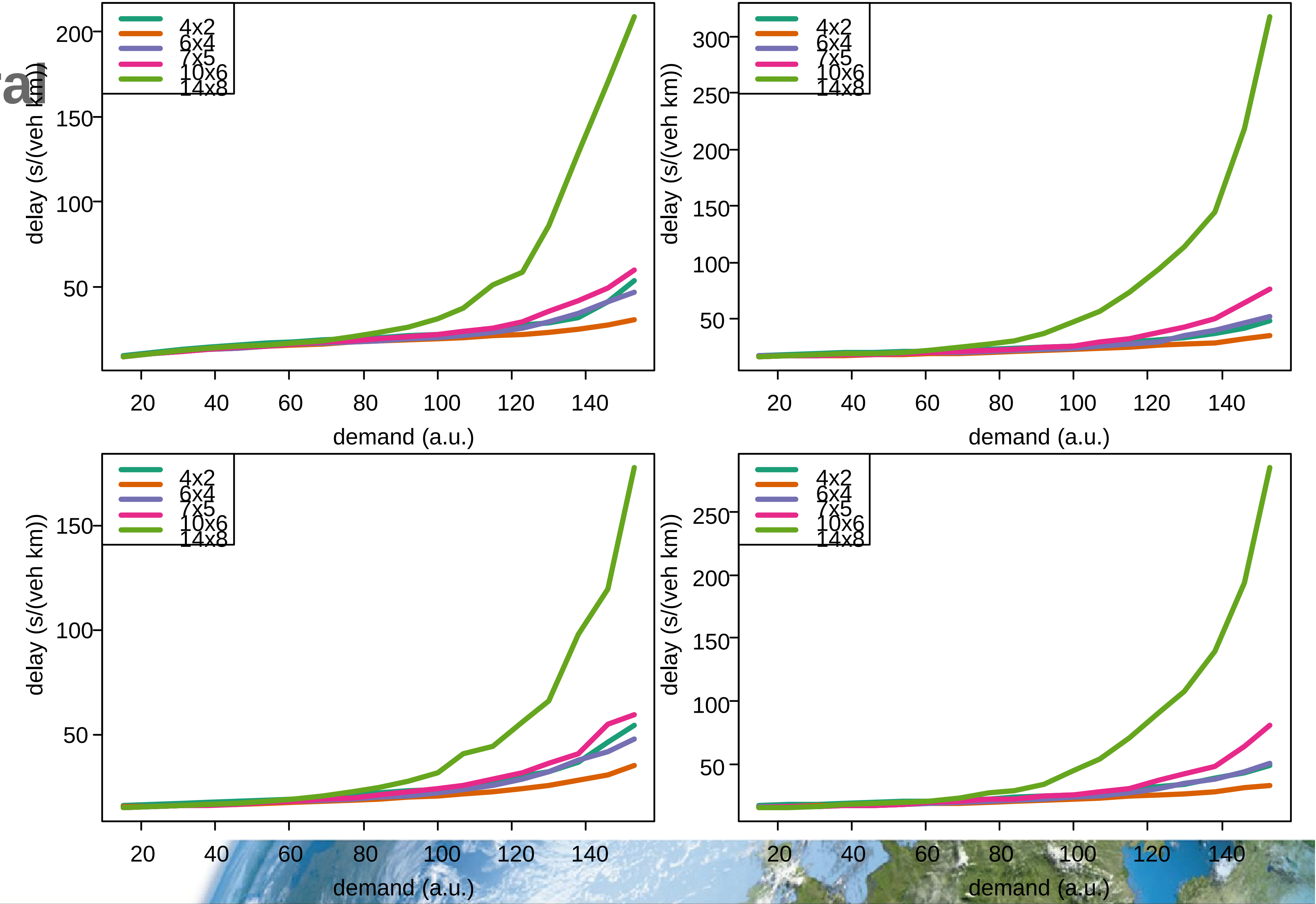
- 4×2, 5×3, 6×4, 7×5, 10×6, 14×8
- with 5 repetitions and 19 different demands





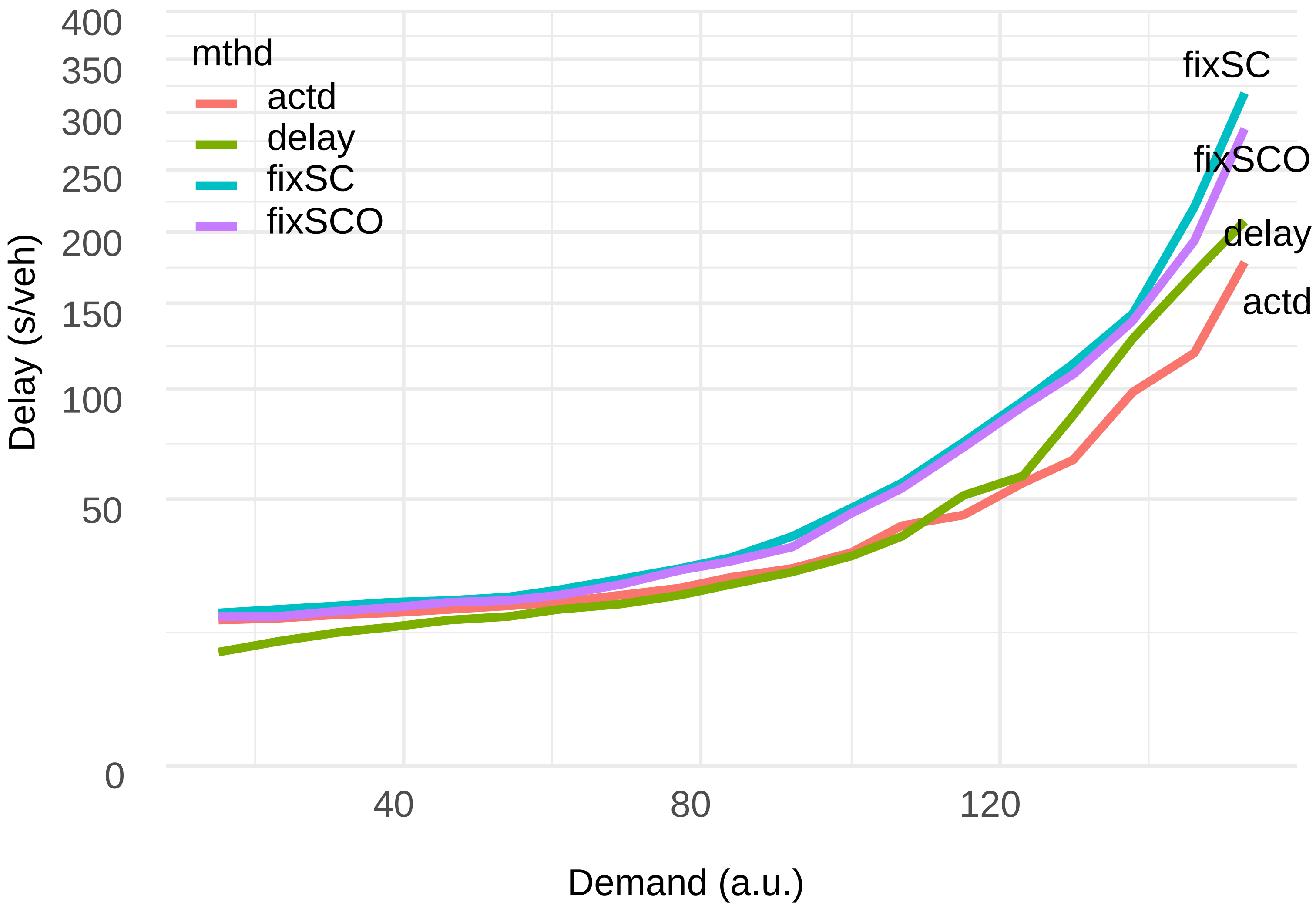
Most general

- All networks, all demands, all repetitions
- Too much information
- Pick largest net only



Most general

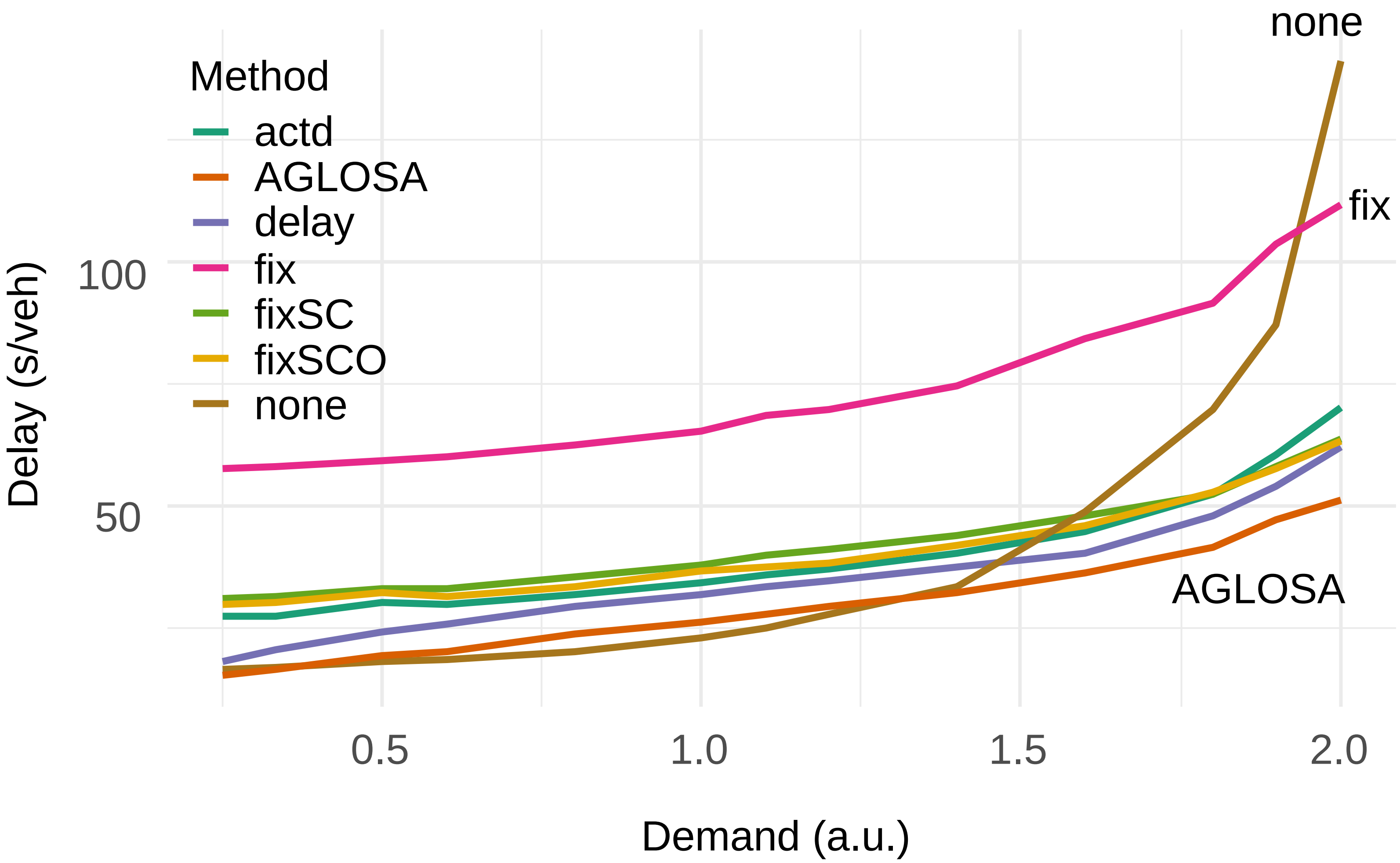
- Actuated (actd) and delay are single intersection policies “SOTL light”
- fixX are Great Plans, with or without co-ordination
- General: SC is a good idea
- Co-ordination give slight improvements
- SOTL methods better, for all demands



Digging deeper, 5 x 3 network, details

- Fix: SUMO's default (as worse as it gets)
- AGLOSA: is truly dealing with networks, too
- None: switch off all lights! ➔ a safety nightmare; a simulation deals with that easily.

- The rest



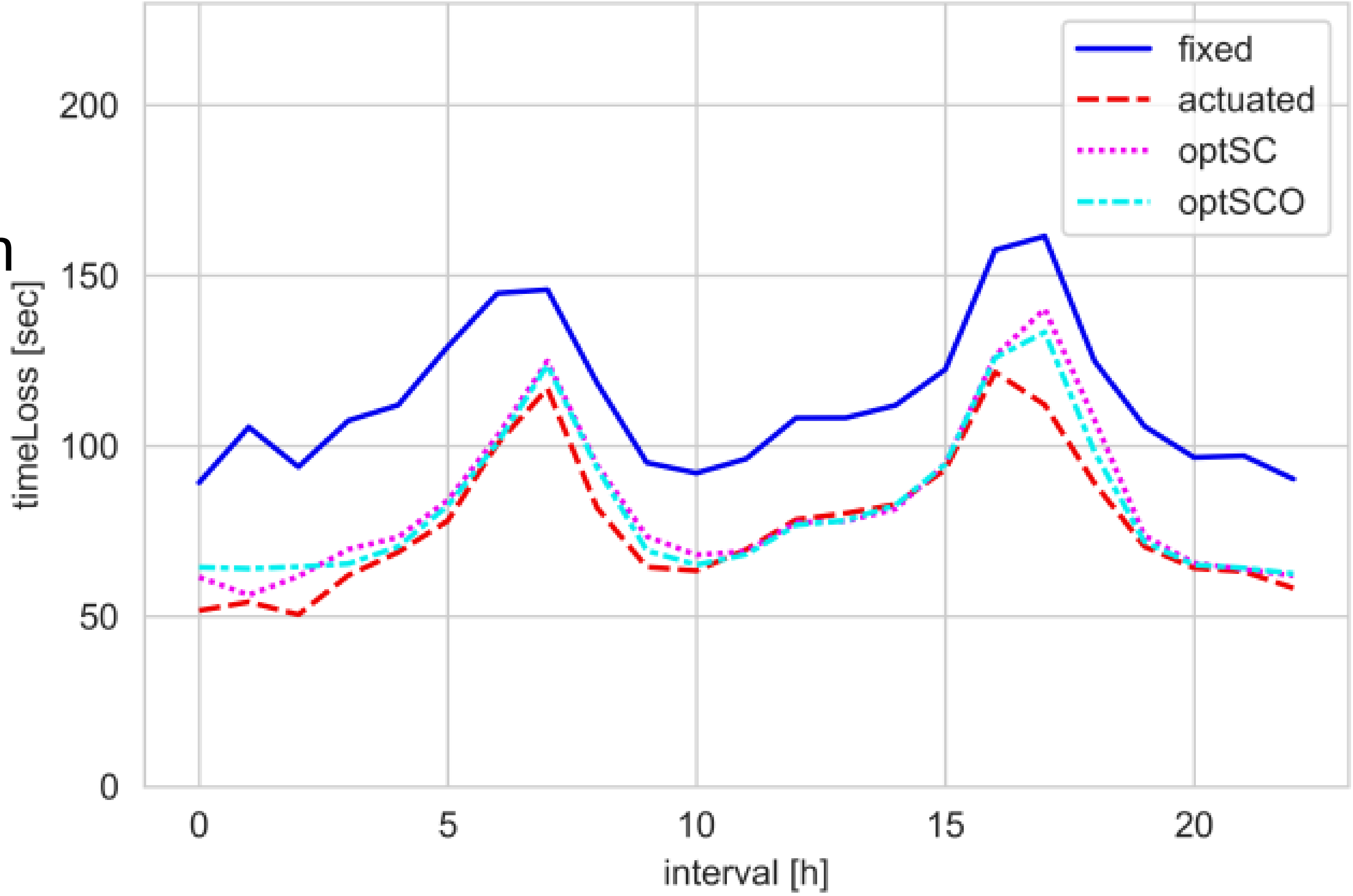
More real-world

Berlin Center

- Real-life network
- 120 traffic signals
- 242 km network length
- 190,000 trips,
- Real demand computed by an external tool
- Network is at the border of capacity
- 24 hour simulation, time-dependent demand

Results are similar…

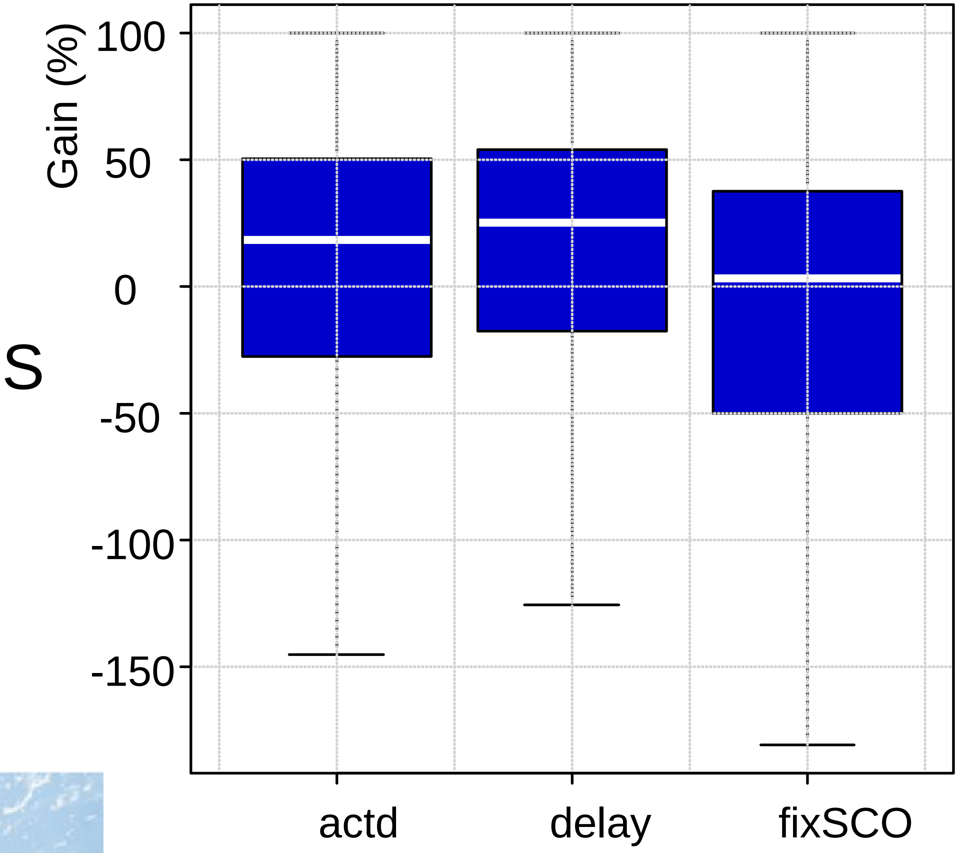
- But not the same.
- Difference between fixed and Webster larger
- Small gains with co-ordination
- Small gains with actuation



Conclusion & Outlook

Figure removed: Self-organization, a flock of birds

- Real world: the Great Plan seems to be underperforming (3% gain for co-ordination)
- Ideal case: w/o platoon dispersion, and highly idealized demand, it may have an edge
- If results apply to real life, then running all signals actuated yields smooth traffic in a city (18% / 25% with large dispersion)
- Can gain even more when using network-ready TLS like AGLOSA…
- Needs short-term prediction & planning & communication



Limitations / Remarks

- Each single scenario has one constant demand ➔ favors Great Plans
- The networks are topologically similar to real networks, but they lack their hierarchical structure
- There are better methods to optimize co-ordination, but most of them rely on the idealizations mentioned already
- Large networks are yet different, since they have to divided first in smaller ones

- Relation to this school: something in common with confined diffusion / diffusion in complex environments? Intersections are inhomogeneities. However, most examples I have seen here have a preferred direction; not exactly true for traffic.

Transportation planner’s curse

- But, you know: if you improve traffic signals, what will happen?
- You get even more traffic!

- **Thank you for listening!**

Figure removed

https://www.scienceabc.com/innovation/ready-steady-go-the-evolution-of-traffic-lights.html