COGA
Combinatorial Optimization & Graph Algorithms

Network Routing - Integrating Dynamic Network Flows and Scheduling

Rolf Möhring

Swiss Transport Research Conference
Monte Vertità, Ascona
24 April 2013

DFG Research Center MATHEON
mathematics for key technologies

STRC

Some applied projects

- Adaptive Traffic Control
- Routing of AGVs in the Hamburg harbor
- Constructing periodic timetables in public transport
- Coordinated traffic light control in networks
- Ship Traffic Optimization for the Kiel Canal

Adaptive Traffic Control

- Decentralized methods for optimizing traffic flows, *Sándor Fekete, TU Braunschweig*
- An integrated model for traffic assignment and traffic light optimization in traffic networks, *Ekkehard Köhler, BTU Cottbus*
- Optimized traffic guidance in large scale micro-simulation, *Rolf Möhring, TU Berlin*
- Planning and guiding traffic in megacities, *Kai Nagel, TU Berlin*
- Traffic guidance at great events and evacuation planning, *Martin Skutella, TU Berlin*

Bundesministerium für Bildung und Forschung

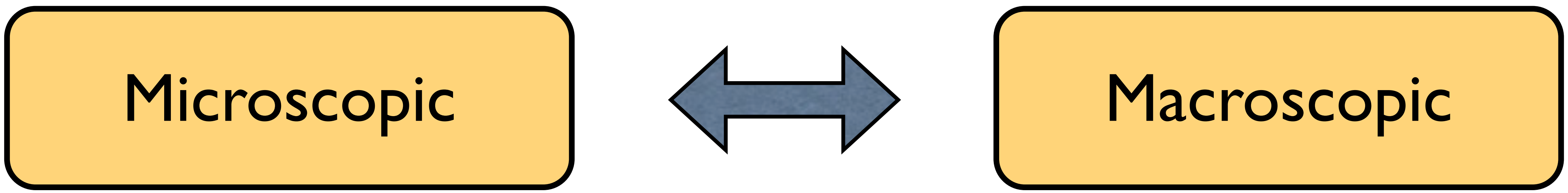
Technische Universität Braunschweig

Brandenburgische Technische Universität Cottbus

http://www3.math.tu-berlin.de/coga/projects/traffic/advest/

Central theme

[with Tobias Harks, Max Klimm, Ingo Kleinert]

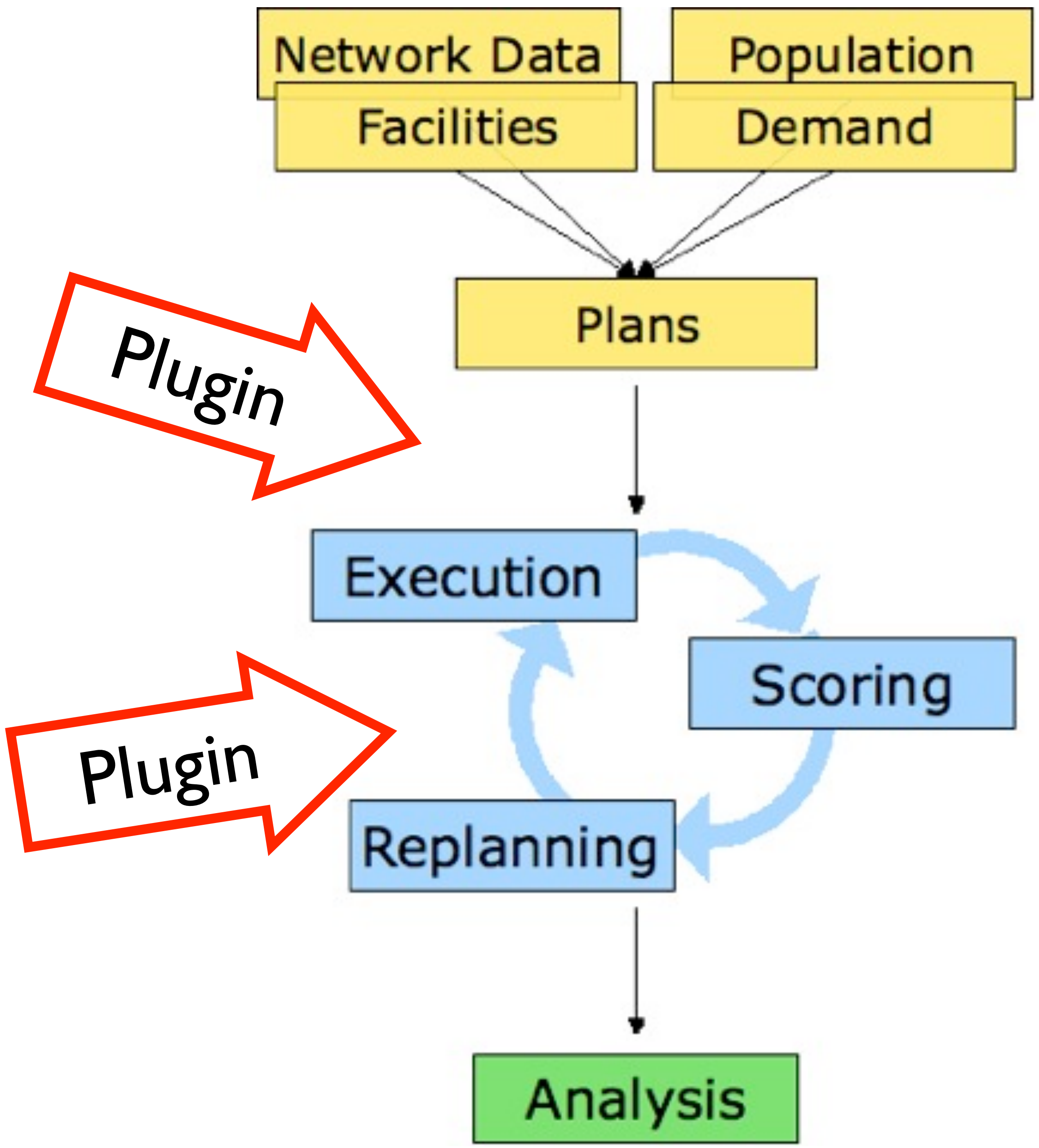


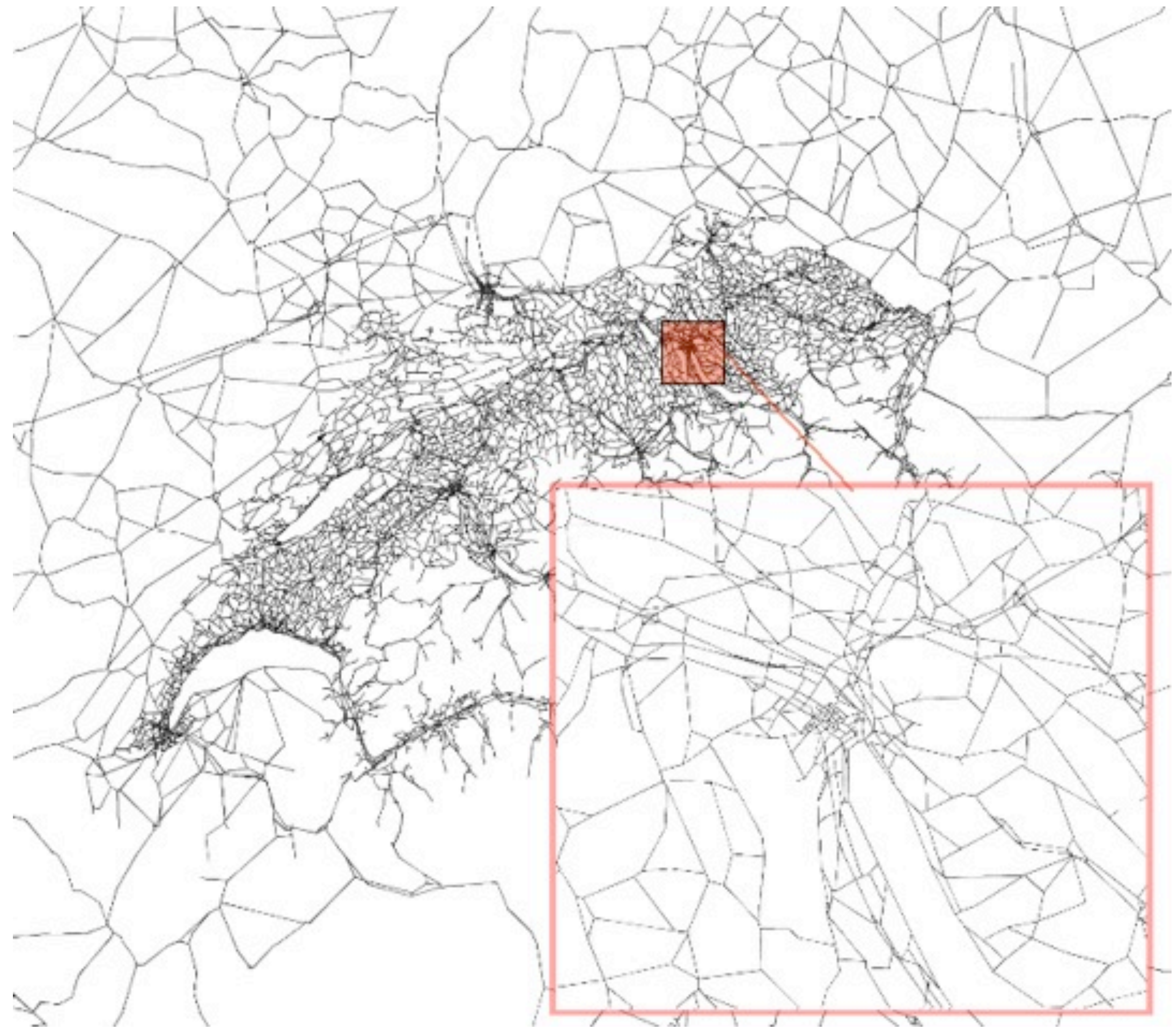
- Detailed simulation
- Study of many real life effects
- No optimization in the mathematical sense

- Coarse models
- Miss many real life aspects
- Optimization methods available or under development

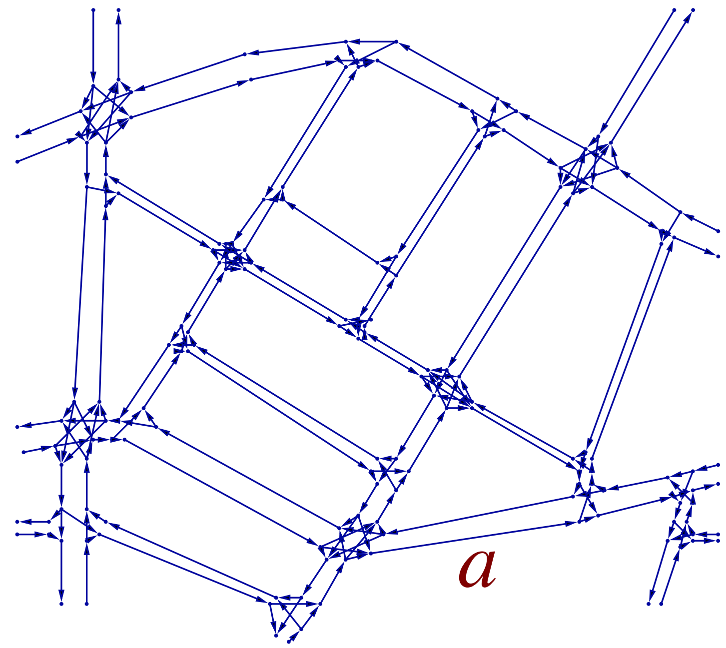
The MATSim microsimulation

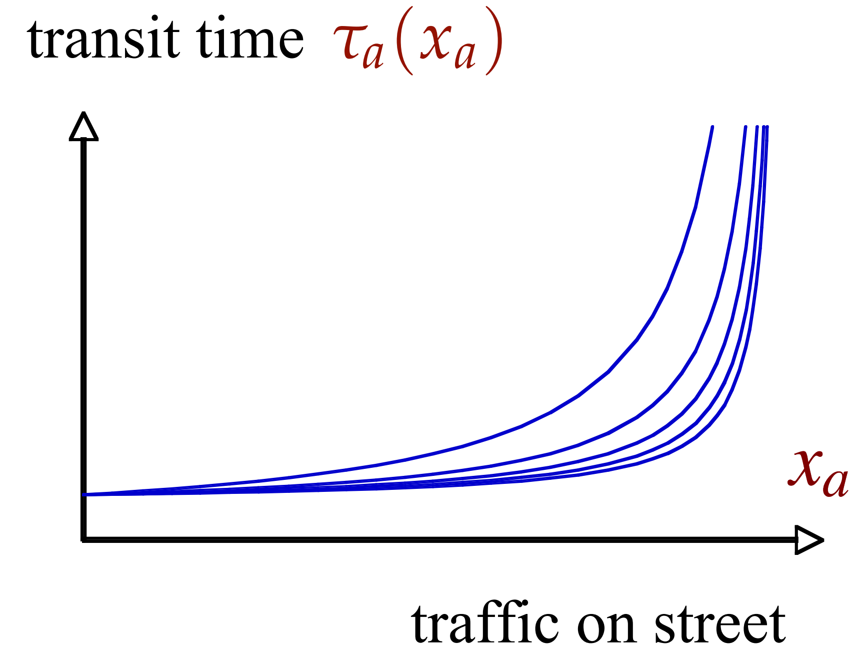
www.matsim.org [Kai Nagel et al.]





An optimization model for the rush hour





- Street graph with capacities and transit time functions
- Origin-destination demands for the rush hour
- Route the demand subject to the capacities such that the total travel time is "small" (system optimum)

$$\sum_{a \in A} x_a \cdot \tau_a(x_a)$$

Selfish routing leads to "user equilibrium"

Nash equilibrium

- Nobody can improve his route just by himself
- All alternatives take at least as long as the current route

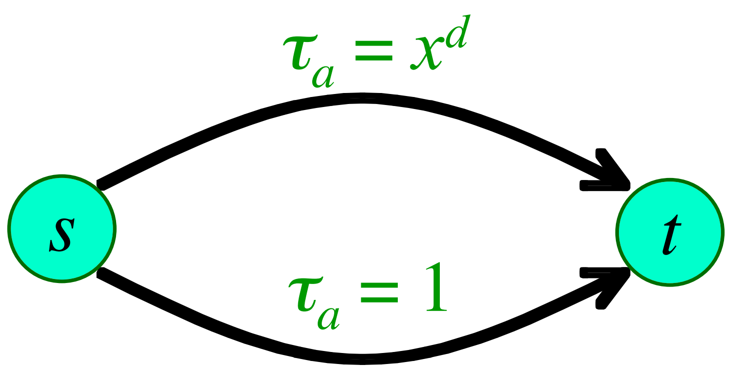
User equilibrium is paradox ridden

User equilibrium has higher network load than necessary
Price of Anarchy

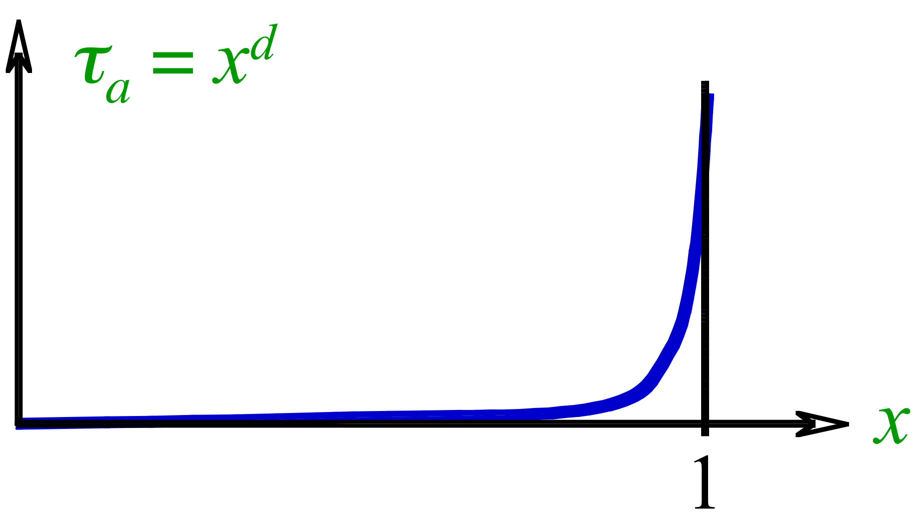
The price of anarchy

$$\text{Price of anarchy} := \frac{\text{Network load in user equilibrium}}{\text{Network load in system optimum}} = \frac{\text{UE}}{\text{SO}}$$

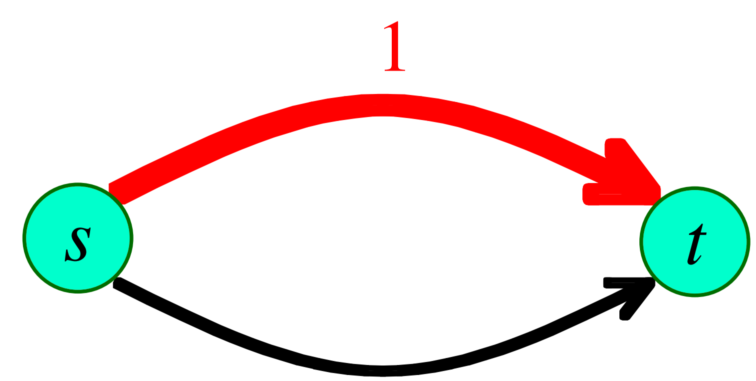
Pigou's example [1920]



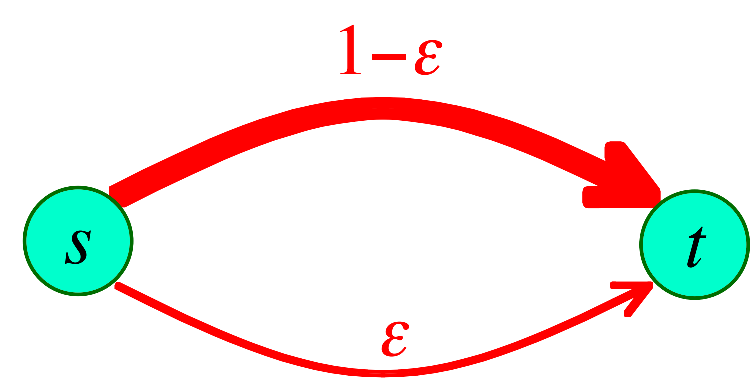
Demand b = 1



User equilibrium has value 1

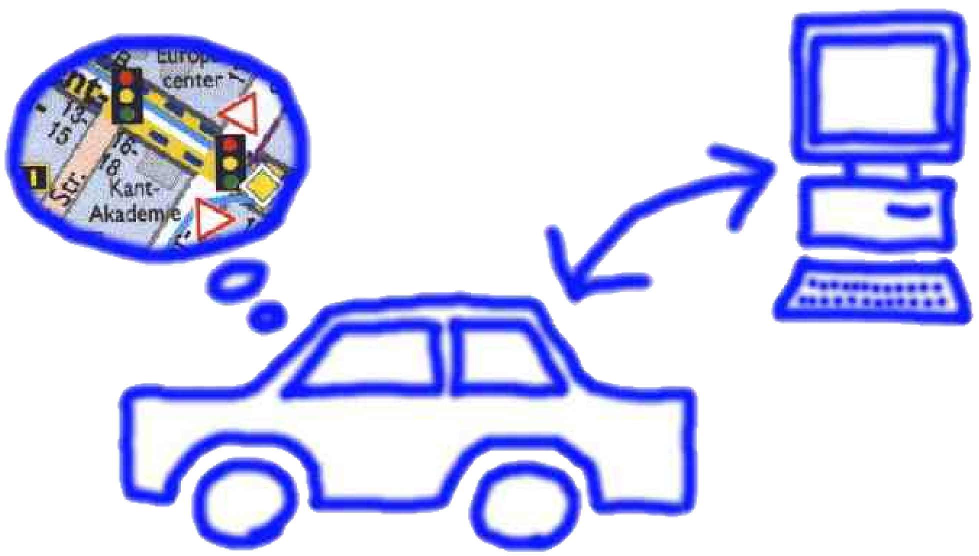


System optimum has value ~ 0



Centralized traffic management

Technological requirements



- exact knowledge of current position
- 2-way communication to a main server
- server has "complete" knowledge" of current state

- Can achieve the "system optimum"

But

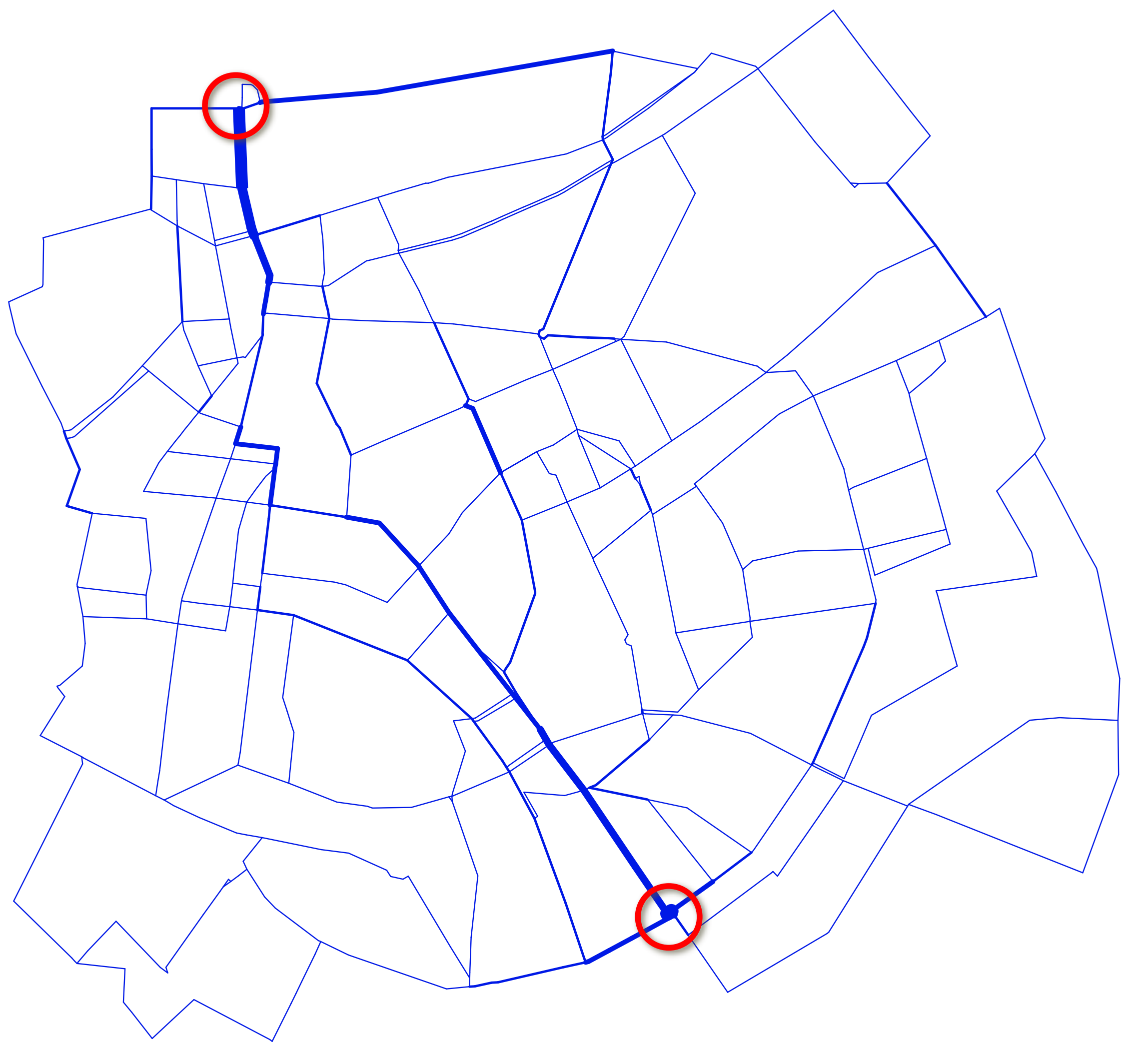
- Individual routes may be far too long!

Central assignment of routes

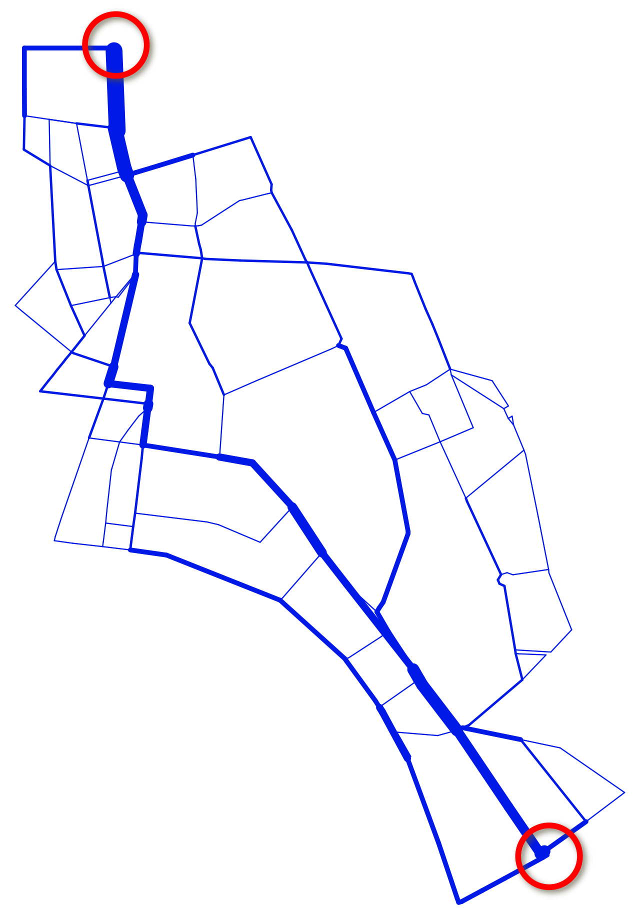
System optimum without ...

... and with fairness conditions

Central assignment of routes



System optimum without ...



... and with fairness conditions

Definition of fairness

Unfairness of a route guidance policy :=

$$\max_{\text{all OD pairs } i} \frac{\text{maximum travel time on a route for OD Paar } i}{\text{travel time in user equilibrium for OD Paar } i}$$

Unfairness (user equilibrium) $= 1$

Unfairness (system optimum) may be arbitrarily large

Our "fairness" algorithm CSO

[Jahn, M., Schulz, Stier 2005]

- Fractional multicommodity flow problem
  - convex separable objectice
  - side constraints on routes

Berlin in 20 minutes

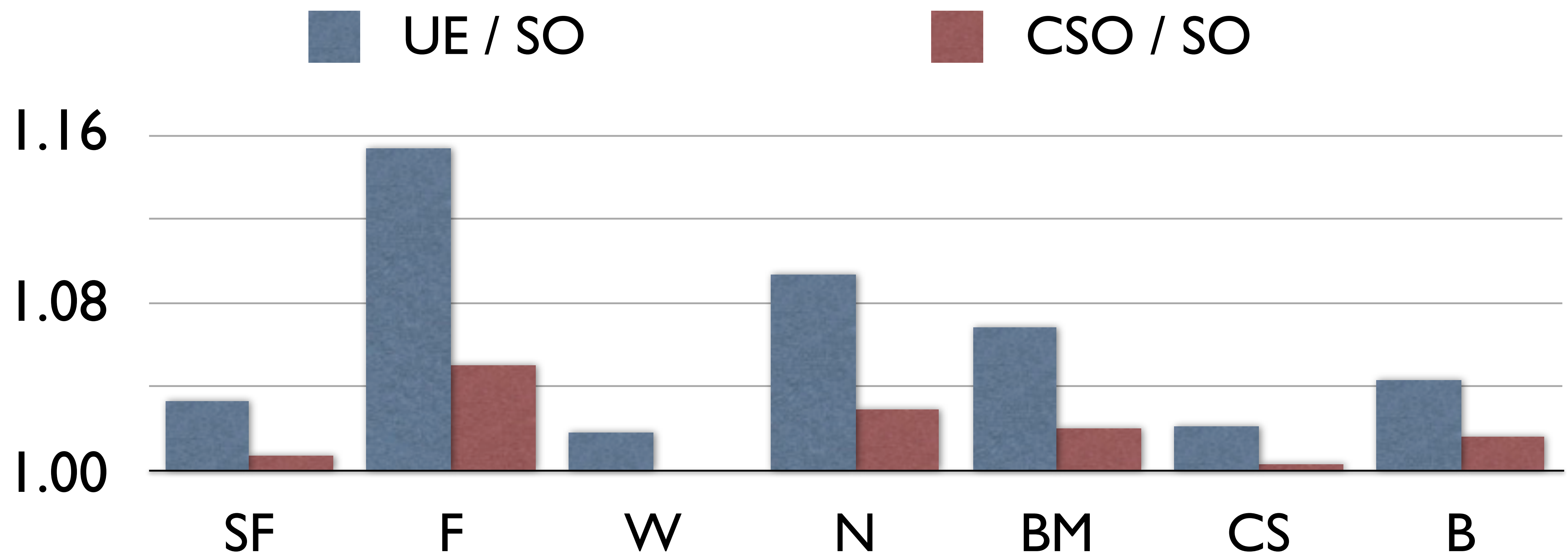
Gradient method (variant of Frank-Wolfe)

Simplex algorithm with column generation

Computing constrained shortest paths

Shortest paths (e.g. Dijkstra)

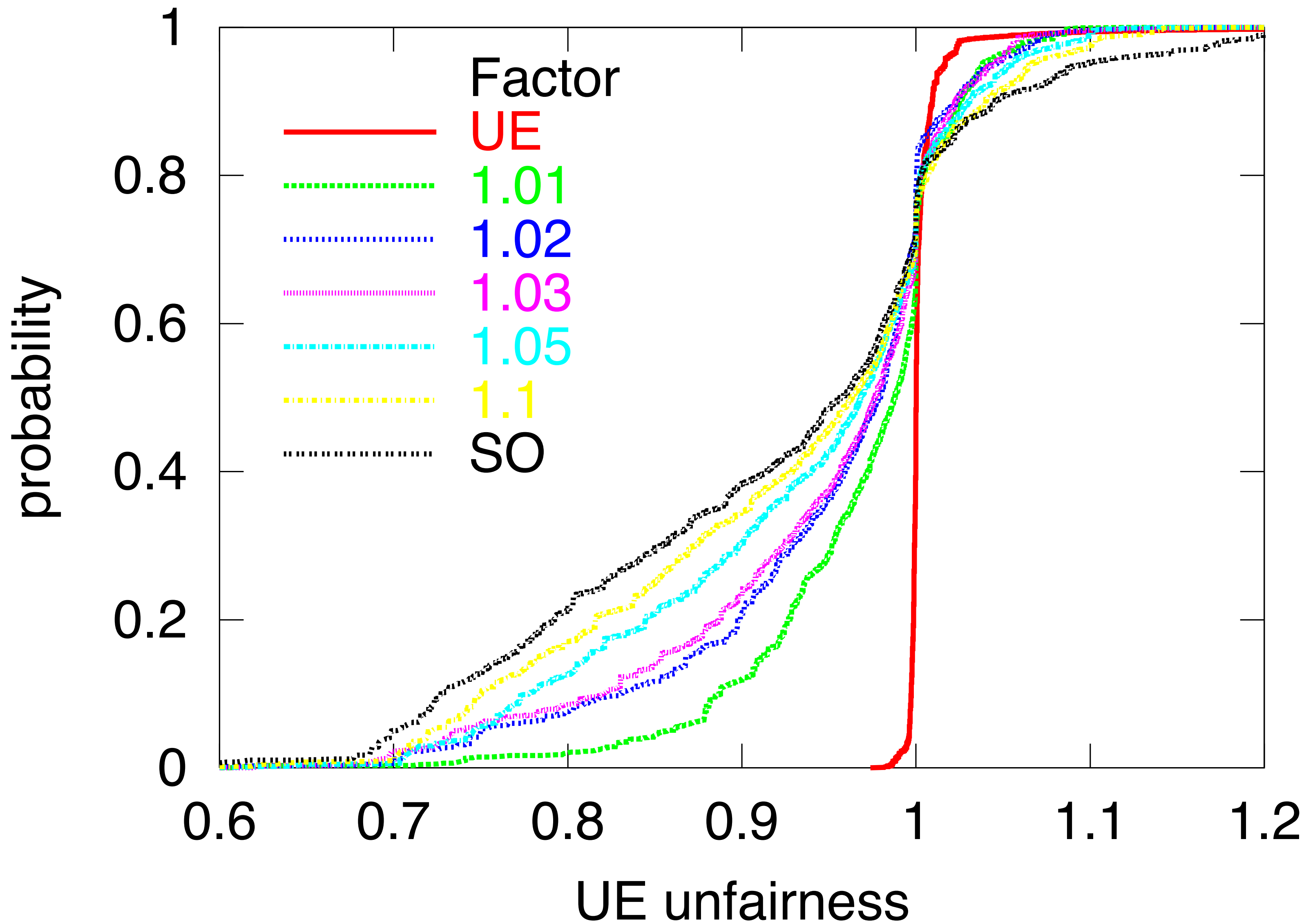
Results for some cities



SF Sioux Falls	F Friedrichshain	W Winnipeg	N Neukölln
BM Berlin Mitte	CS Chicago Sketch	B Berlin	

- initial theory [Roughgarden 03] $\frac{UE}{SO} \leq 2.151$
- improved analysis [Correa, Schulz, Stier 05] $\frac{UE}{SO} \leq 1.365$

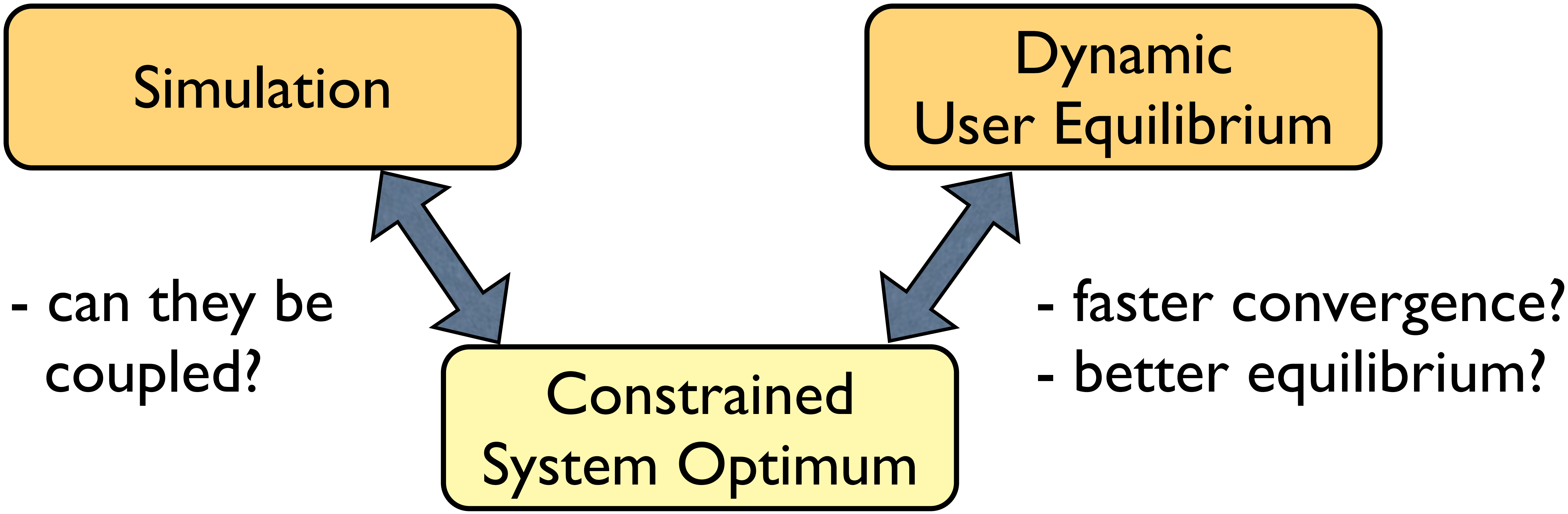
Analysis of fairness



- 75% of the users travel less than in equilibrium
- Only 0.4% of the users travels 10% more than in equilibrium
- For the system optimum, these are more than 5%

Questions and results

[Tobias Harks, Max Klimm, Ingo Kleinert, RHM 2011]



Stackelberg routing

„TomTom view"

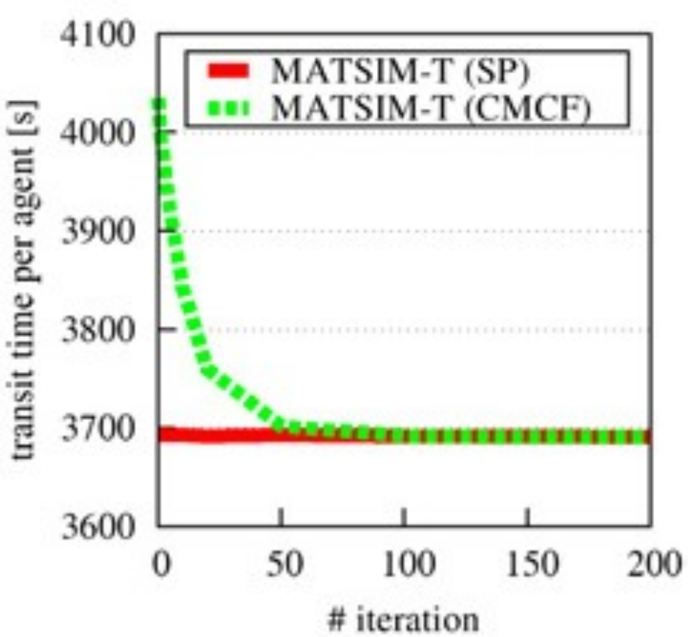
Congestion pricing

„Government view"

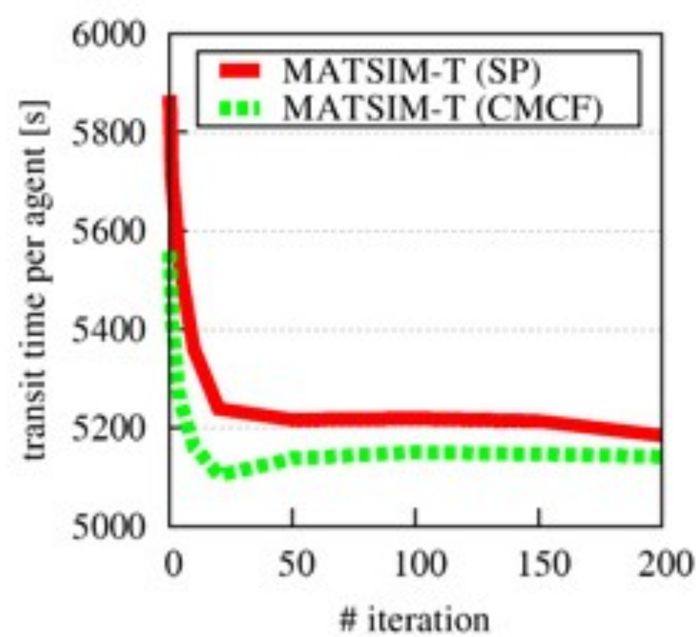
Optimization helps simulation

- Use optimal routes from constrained system optimum as start routes in simulation
- Tests with MATSim

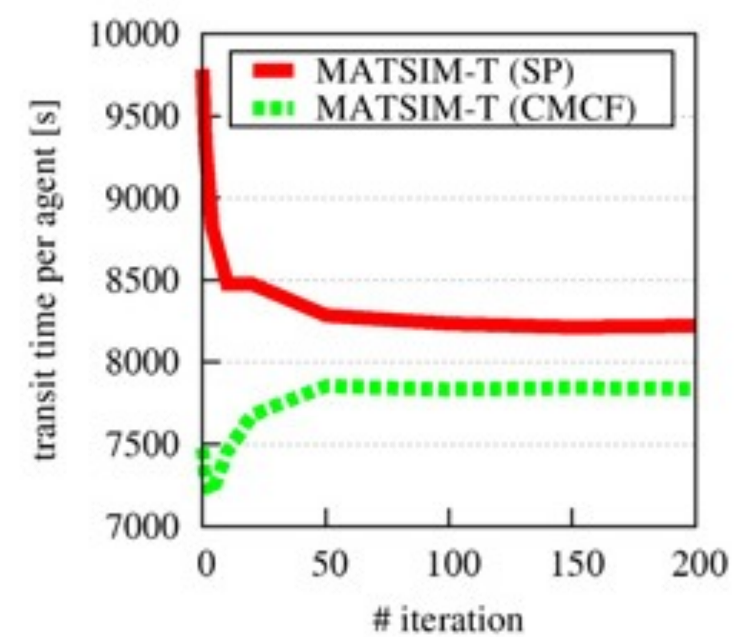
Results for Zurich



2,000 OD pairs

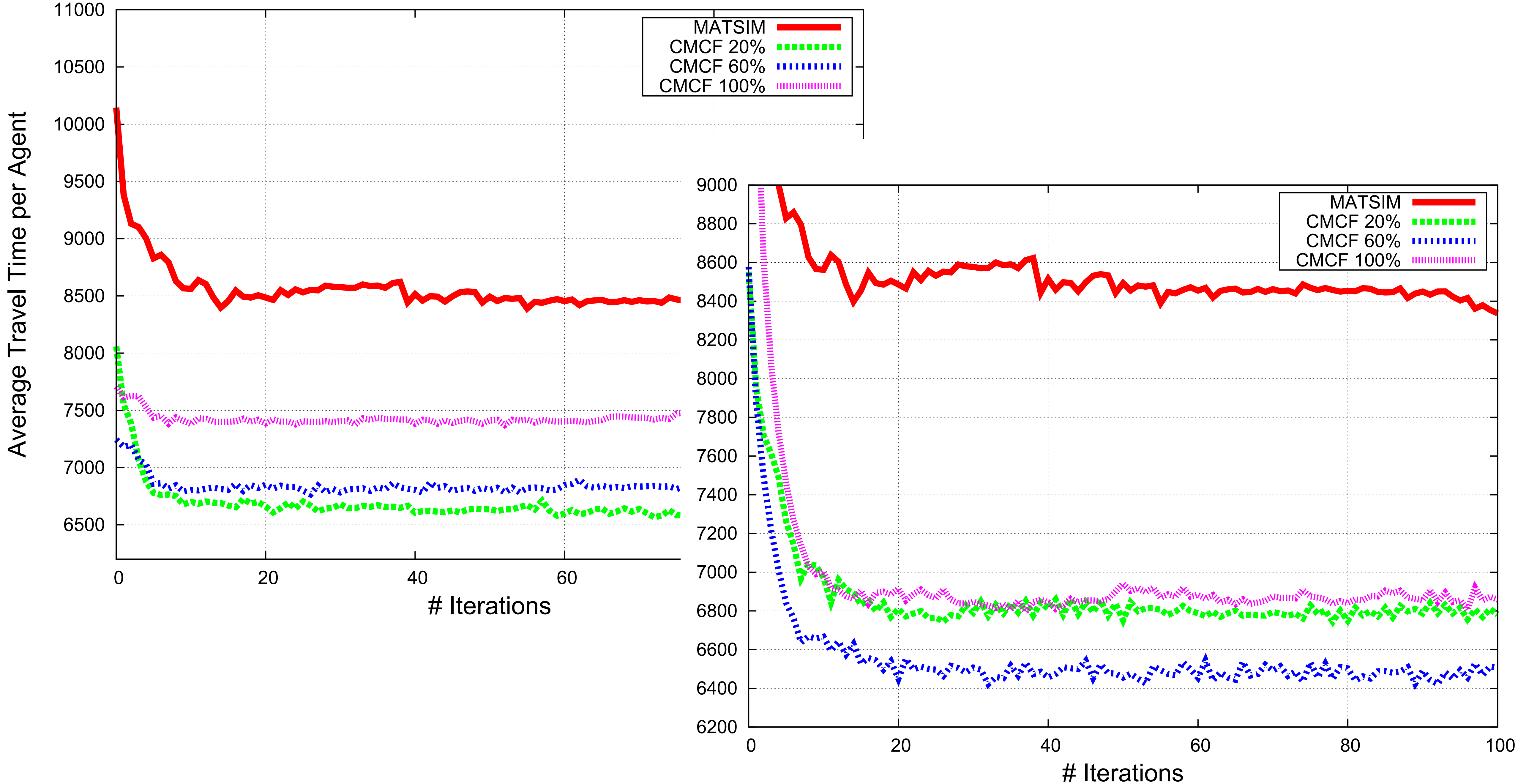


20,000 OD pairs



50,000 OD pairs

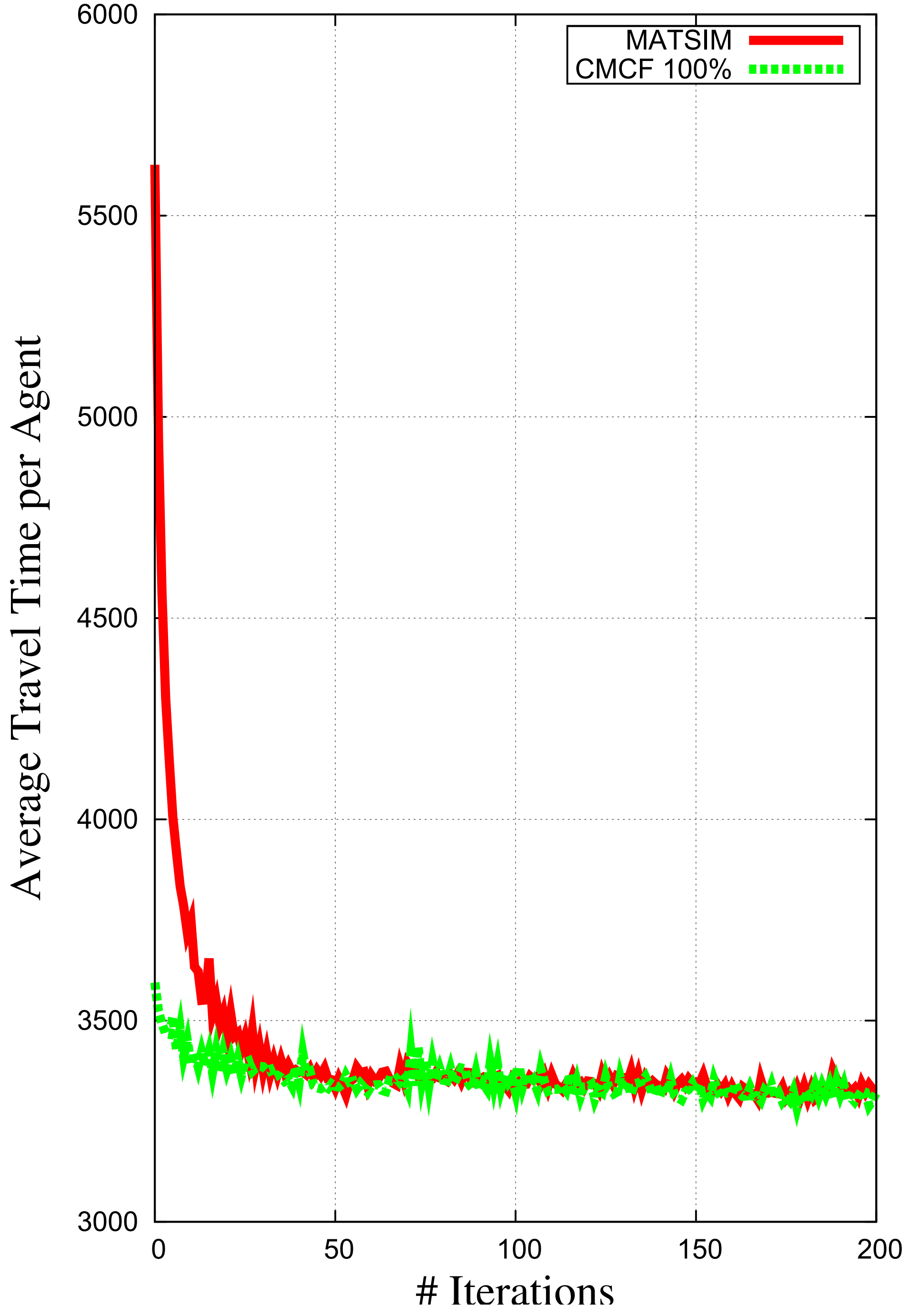
Looking more closely

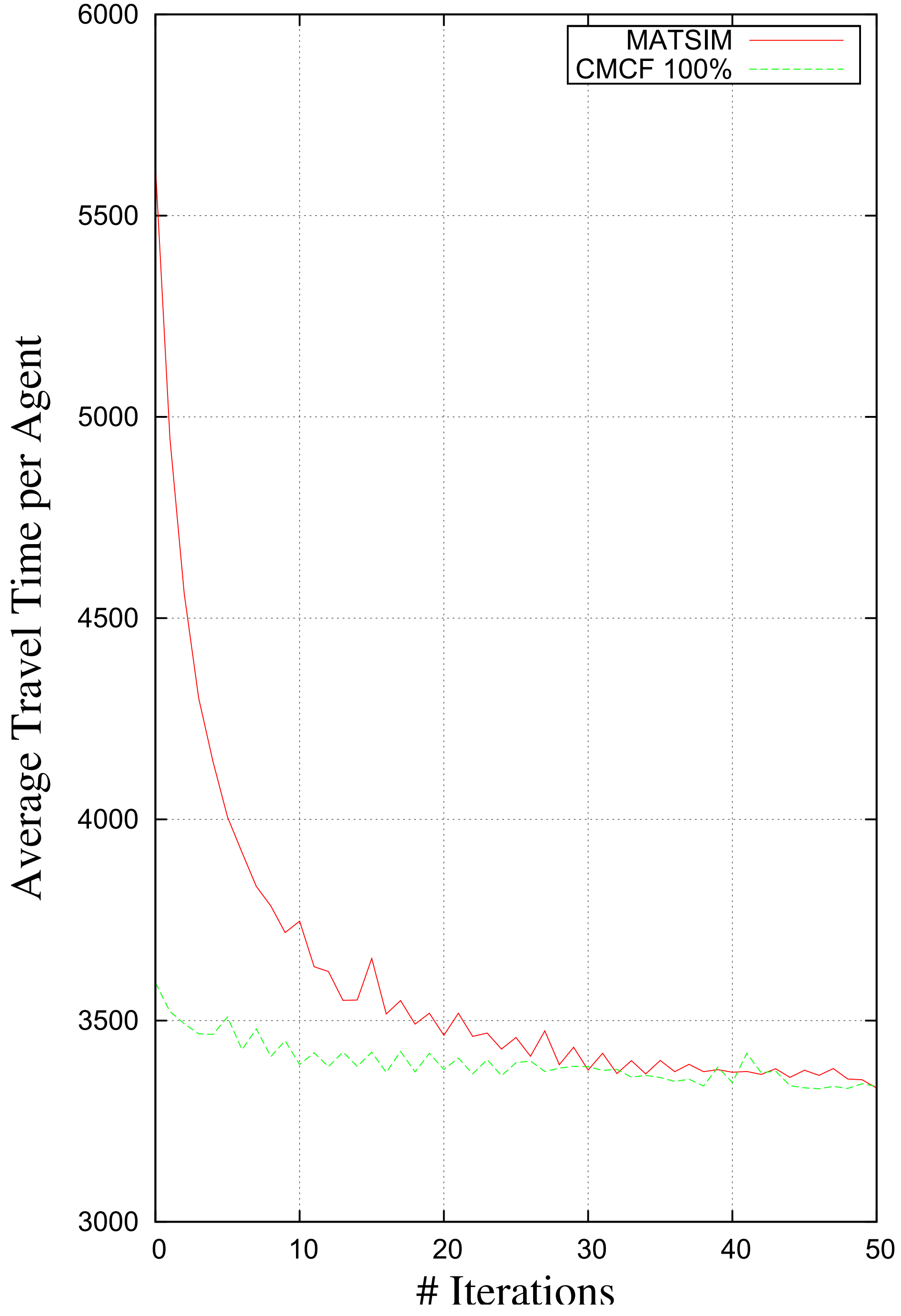


multiple dynamic equilibria for fixed departure times

Departure time choice helps

Still fits into the static model when the departure time interval is discretized





Summary coupling optimization and simulation

- Simulation converges faster
- Multiple equilibria exist, but are rare
- Variable departure times seem to lead to unique equilibria
- Optimizing over routes and departure times helps simulations

Congestion pricing

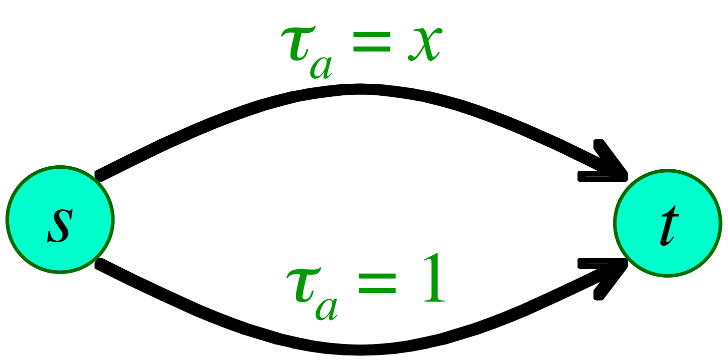
- Marginal cost pricing on all edges leads to the system optimum [Beckmann et al. `56]
  - f is SO w.r.t. $\tau_a(x_a)$ $\Leftrightarrow$ f is UE w.r.t. $\tau_a(x_a) + x_a \tau_a'(x_a)$

- tolls can „in principle“ achieve SO
  - need to choose arc cost as $\underbrace{\tau_a(x_a)}_{\text{time}} + \underbrace{x_a \tau_a'(x_a)}_{\text{toll, depends on flow } x_a}$

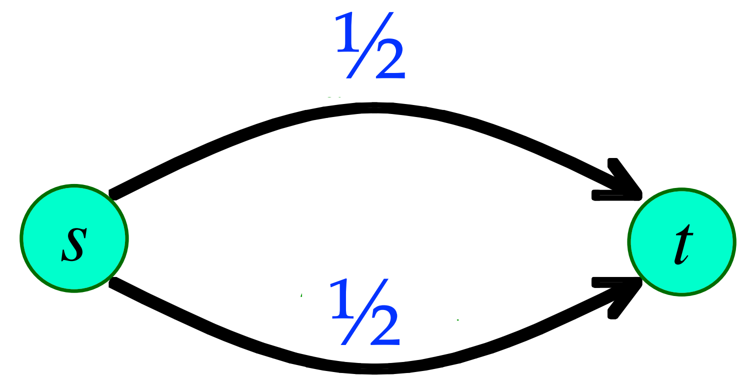
time

toll,
depends on flow x_a

Tolls achieve system optimum on Pigou's example



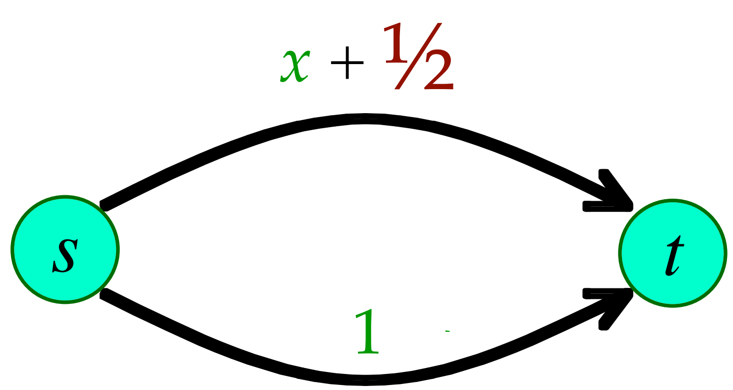
demand b = 1



system optimum

Tolls compute as $x_a \tau_a'(x_a)$ for the system optimal flow

$\Rightarrow$ $½ * 1$ on the top and 0 on the bottom



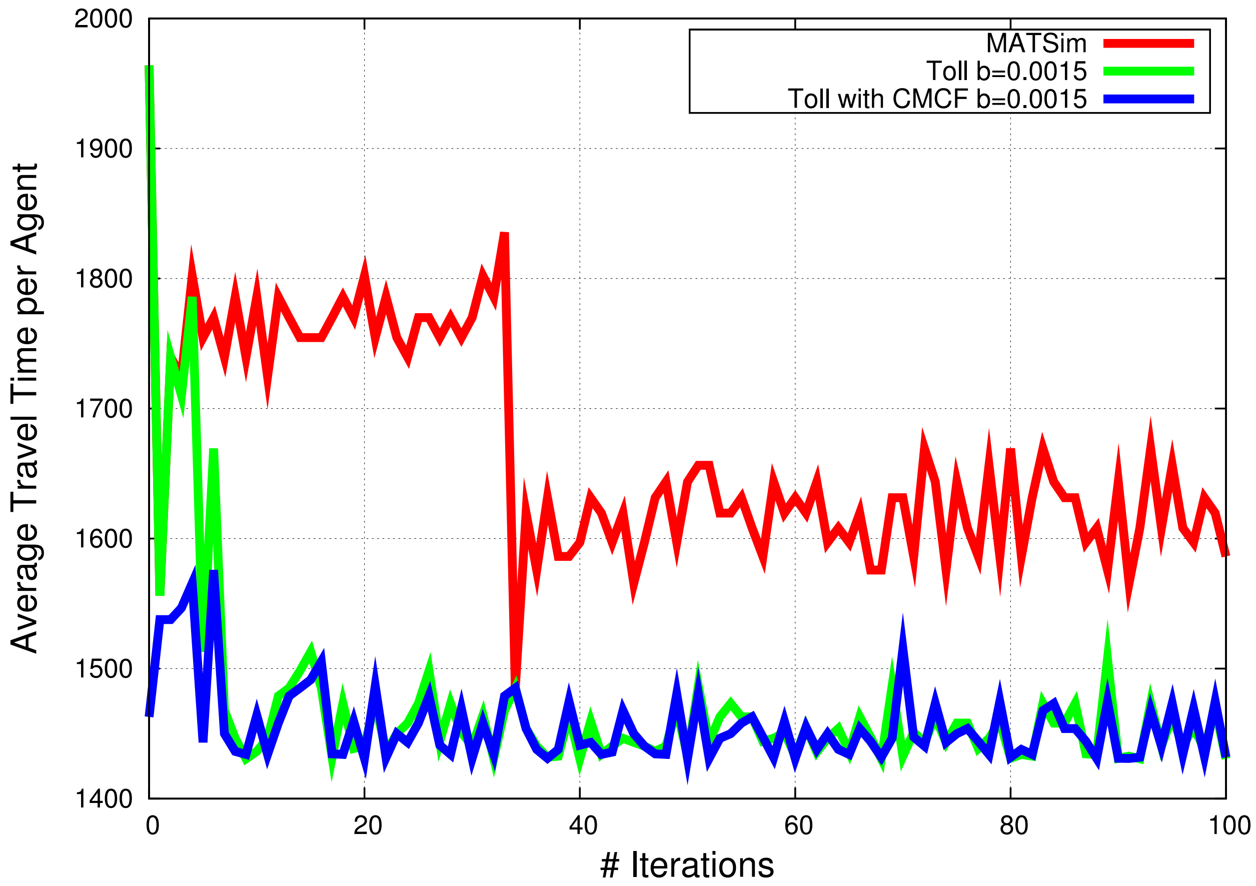
tolled travel times,
achieve system optimum at equilibrium

Known results on congestion pricing

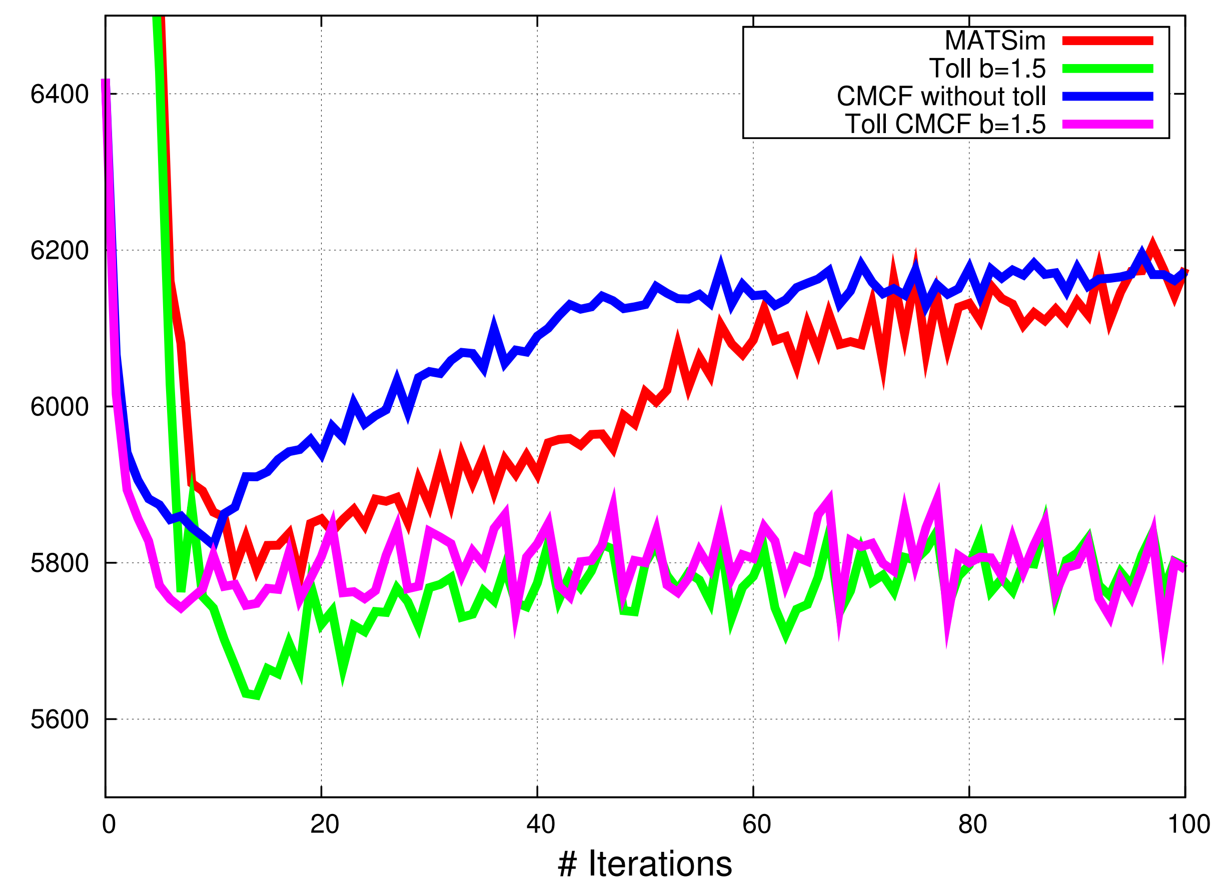
- Polyhedral description of opt-inducing tolls (path-based and arc-based via node potentials)
  [Bergendorff et al. (1997)] [Hearn and Ramana (1998)] [Larsson and Patriksson (1999)]
- Optimize toll-dependent objective function over opt-inducing tolls; in particular: min revenue, min toll-booth problem
  [Hearn and Ramana (1998)] [Dial (1999)]
- Heterogeneous vs. homogeneous users
  [Cole et al. (2003)] [Fleischer et al. (2004)] [Swamy (2007)]
- Characterizing (arc-based) flows that are enforcable by tolls
  [Fleischer et al. (2004)]

Arbitrary networks – arbitrary tolling

- calculate optimum inducing tolls via linear programming
- determine routes with CSO algorithm
- integrate them into microsimulation



Pigou's example

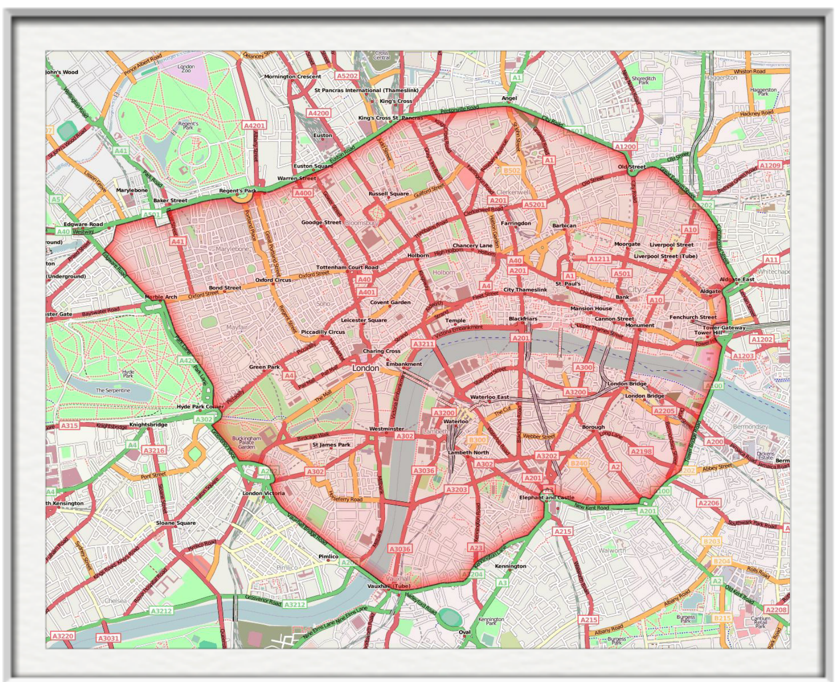


Switzerland

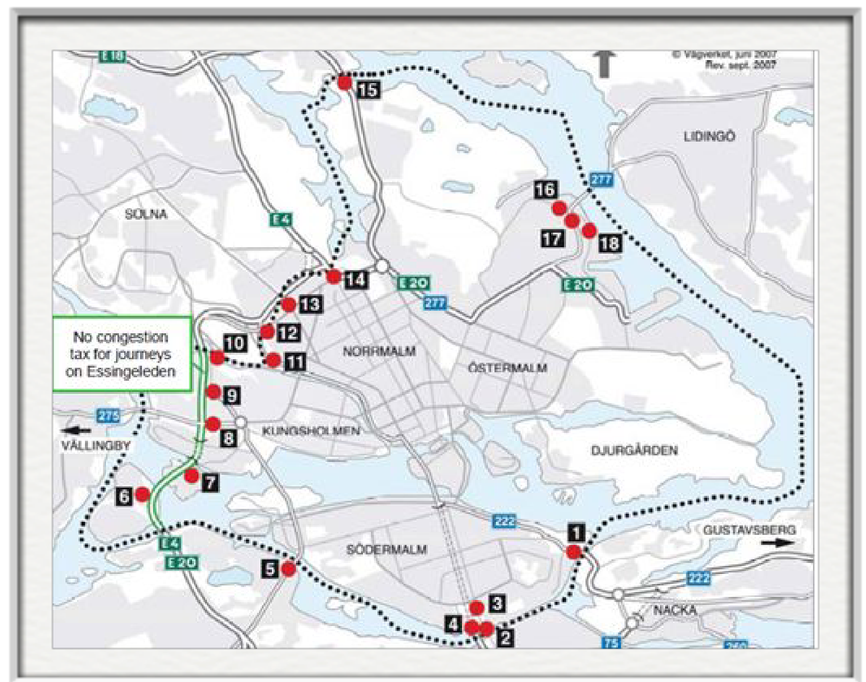
Congestion pricing in practice

- Marginal cost pricing on all edges not feasible in practice

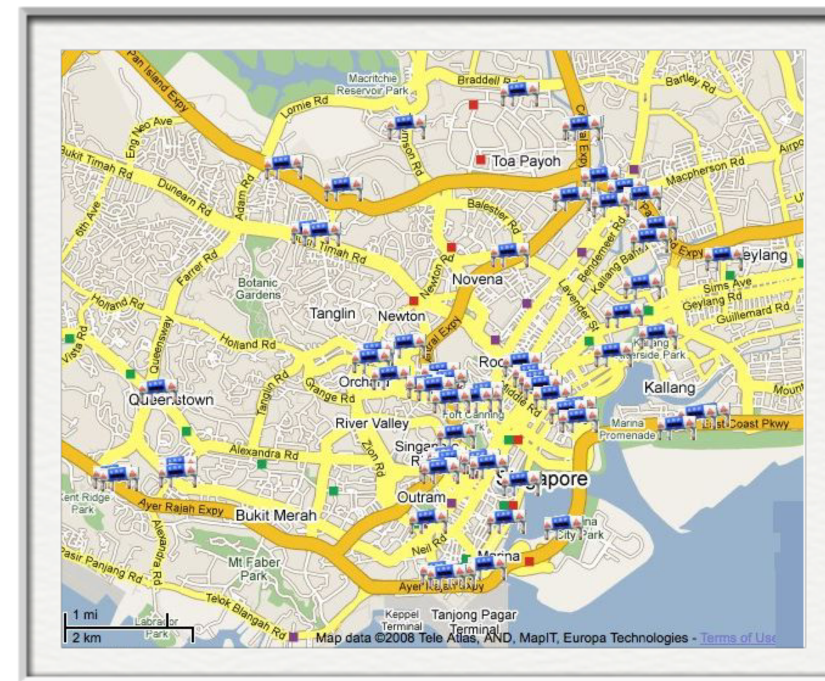
London



Stockholm

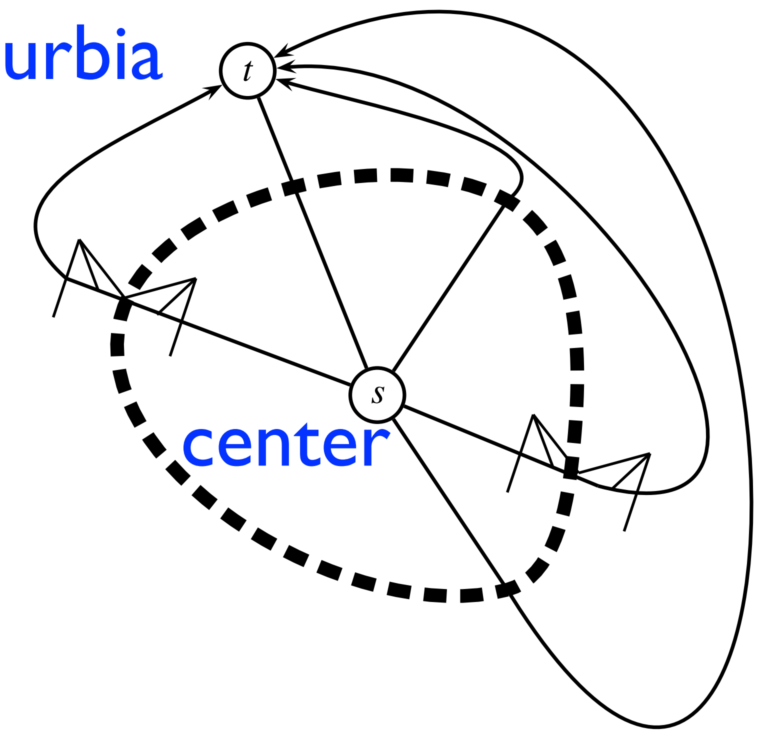


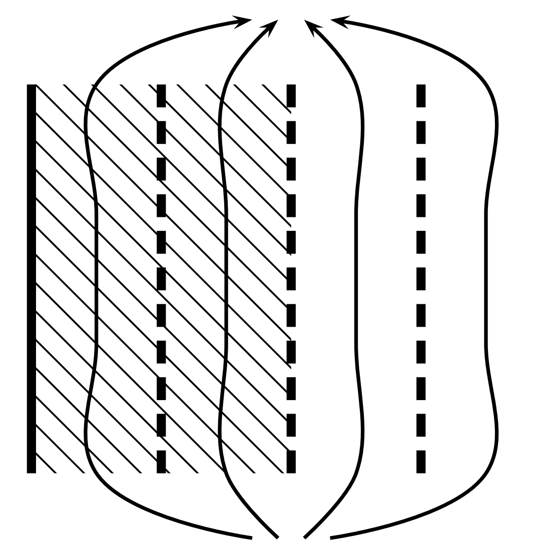
Singapore



- Only a subset of edges has tolls
- Need to study network toll problem with support constraints

Congestion pricing – a solvable case





- efficiently solvable for affine latencies [Hoefer et al. '08]
- Here: algorithm that minimizes the price of anarchy
  - within an additive error of ε in $poly(m, \kappa, d, 1/\varepsilon)$-time
  - with m = #edges, κ = Lipschitz-constant on τ, τ', and τ^{-1},
    d = total traffic demand
- polynomial algorithm when the objective function is convex

Best tolls with limited number of toll booths

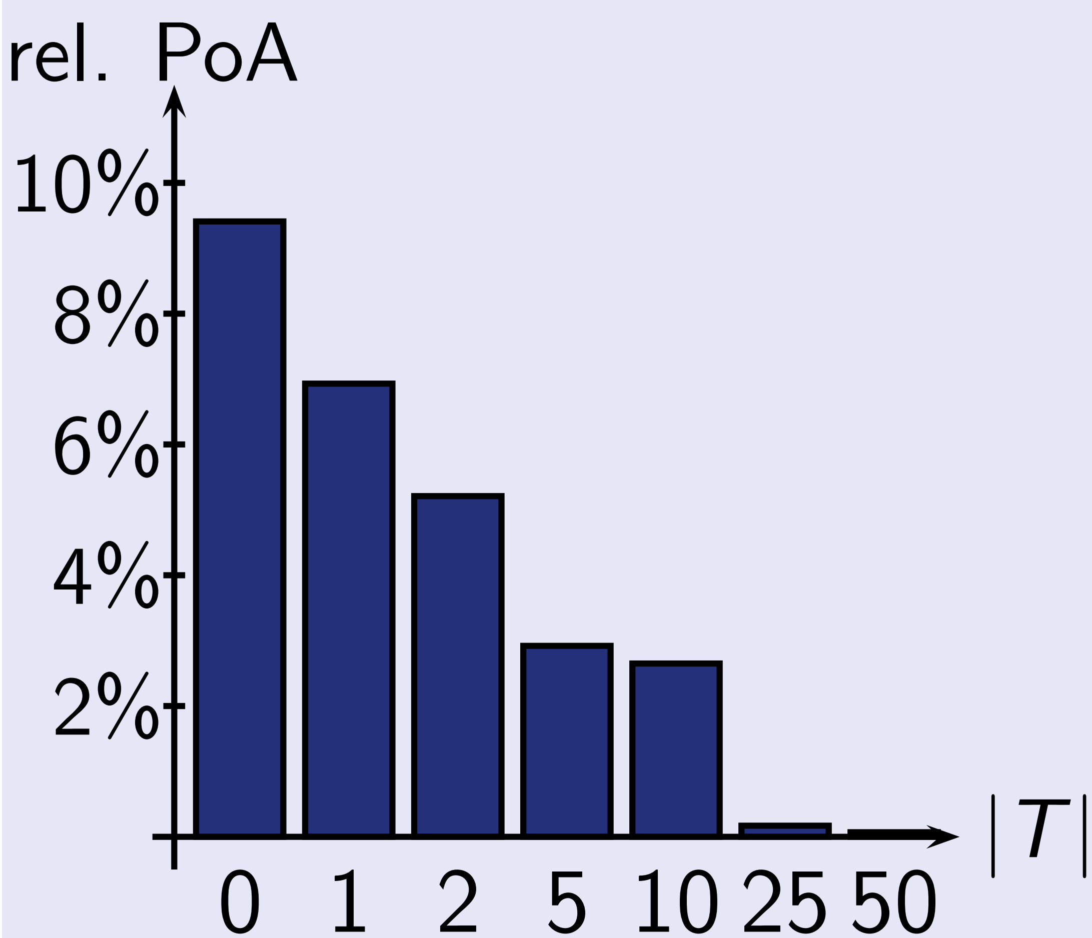
[Tobias Harks, Max Klimm, Ingo Kleinert, Rolf Möhring 2011]

- Is NP-hard, even for two commodities and linear travel times
  [Hoefer et al. 2008]
- No hope for exact solution
- Implemented and tested several algorithms
  - motivated by steepest decent approaches
- Tested on real-world networks

Few tollable edges suffice to reduce congestion

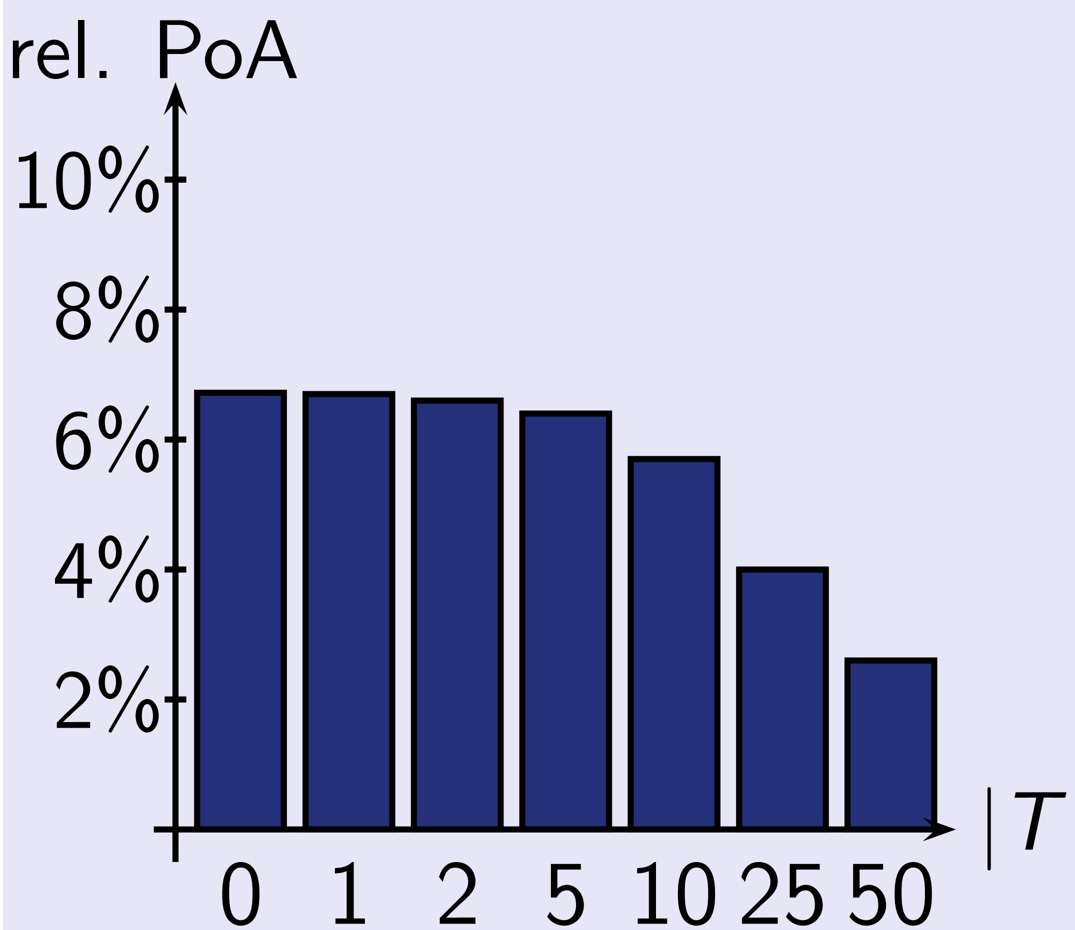
B.-F'hain

$n = 224$, $m = 523$, $k = 506$



B.-Mitte

$n = 1782$, $m = 2935$, $k = 29$



Routing AGVs in the Hamburg Harbour

Ewgenij Gawrilow, Elisabeth Günther,
Ekkehard Köhler, Rolf Möhring, Björn Stenzel

Container Terminal Altenwerder (CTA)

- most modern container terminal
- far reaching automatization on the logistic processes
- expanding at high rates
- here: Transport of containers between waterside and storage area
  - with 70 Automated Guided Vehicles (AGVs)

Overview of the harbor layout

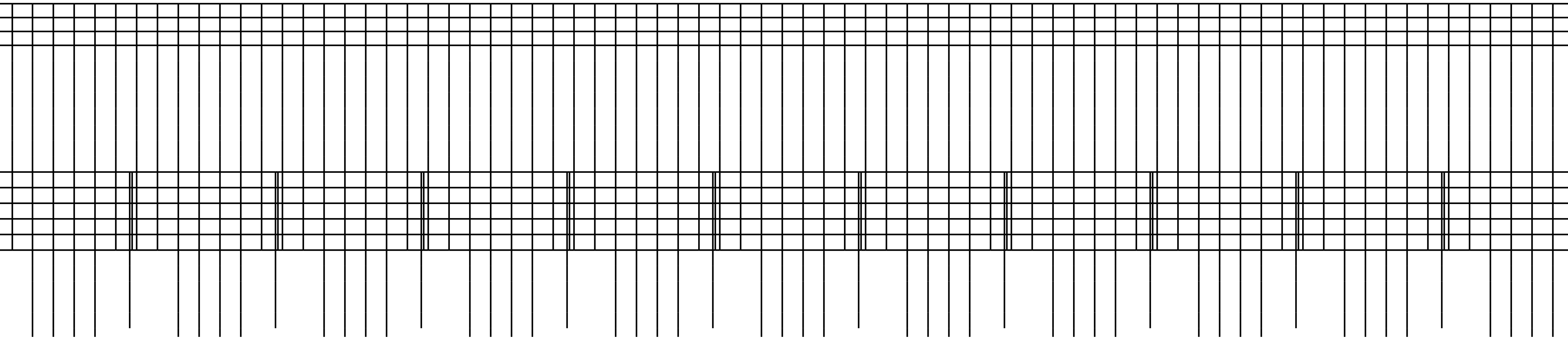
Overview of the harbor layout

Centrally controlled traffic

Optimization model

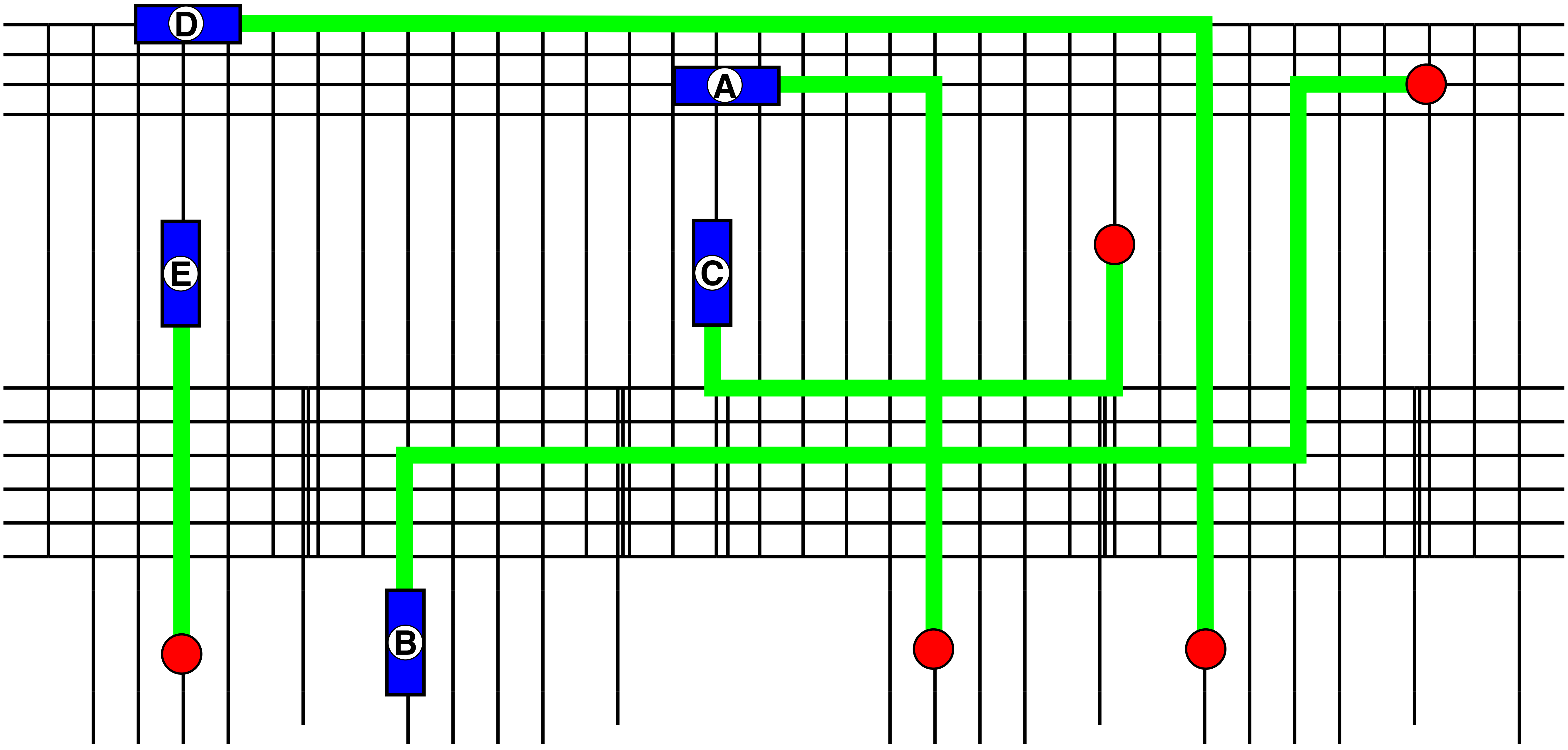
- Graph with 15,647 arcs and 5,445 vertices
- Travel times τ_a on arc a
- Sequence of routing requests with
  - start, destination, departure time
- Wanted:
  - collision-free routes
  - garanteed arrival times
  - high throughput at the bridges (= cranes)

water side

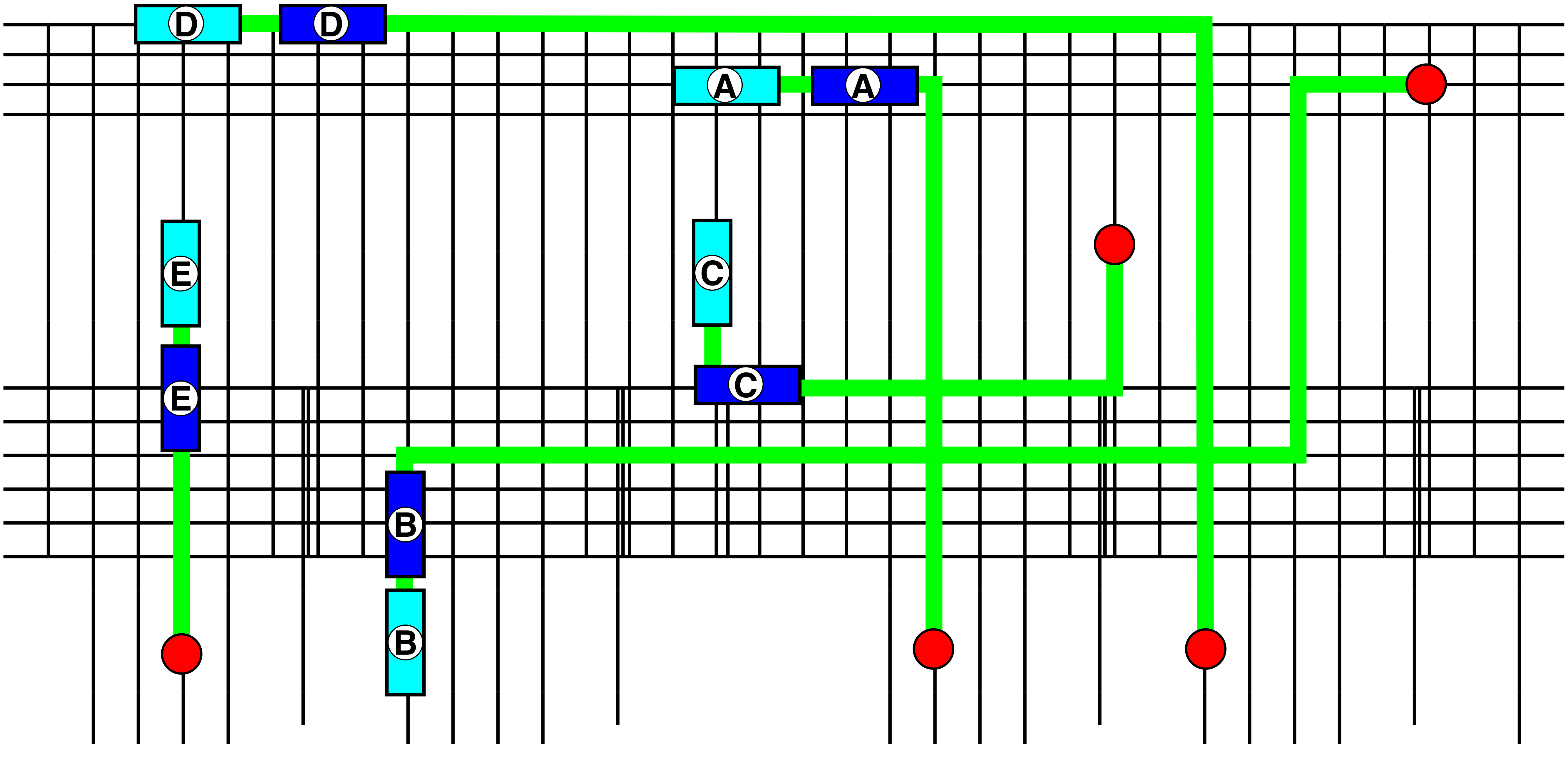


storage area

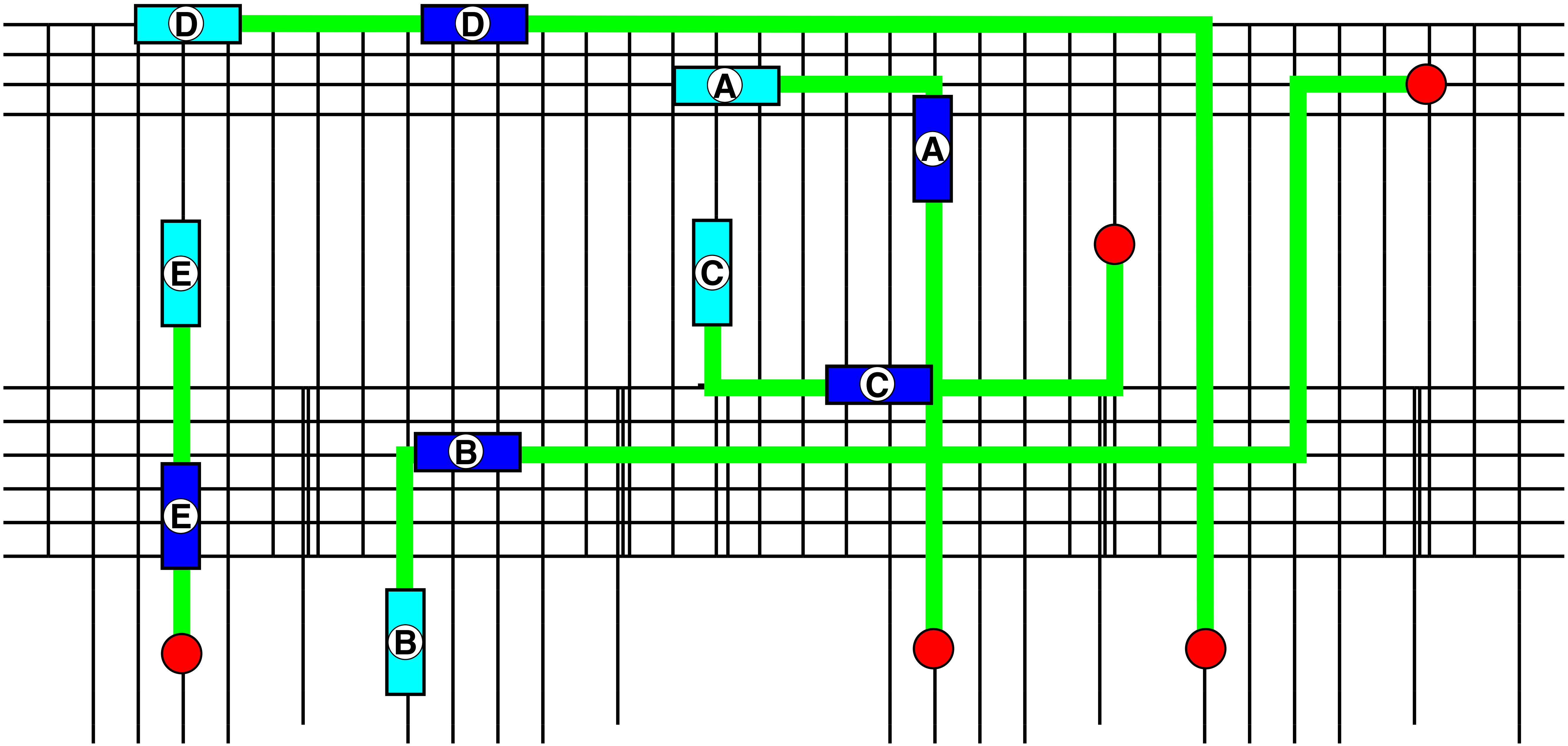
Example of a routing



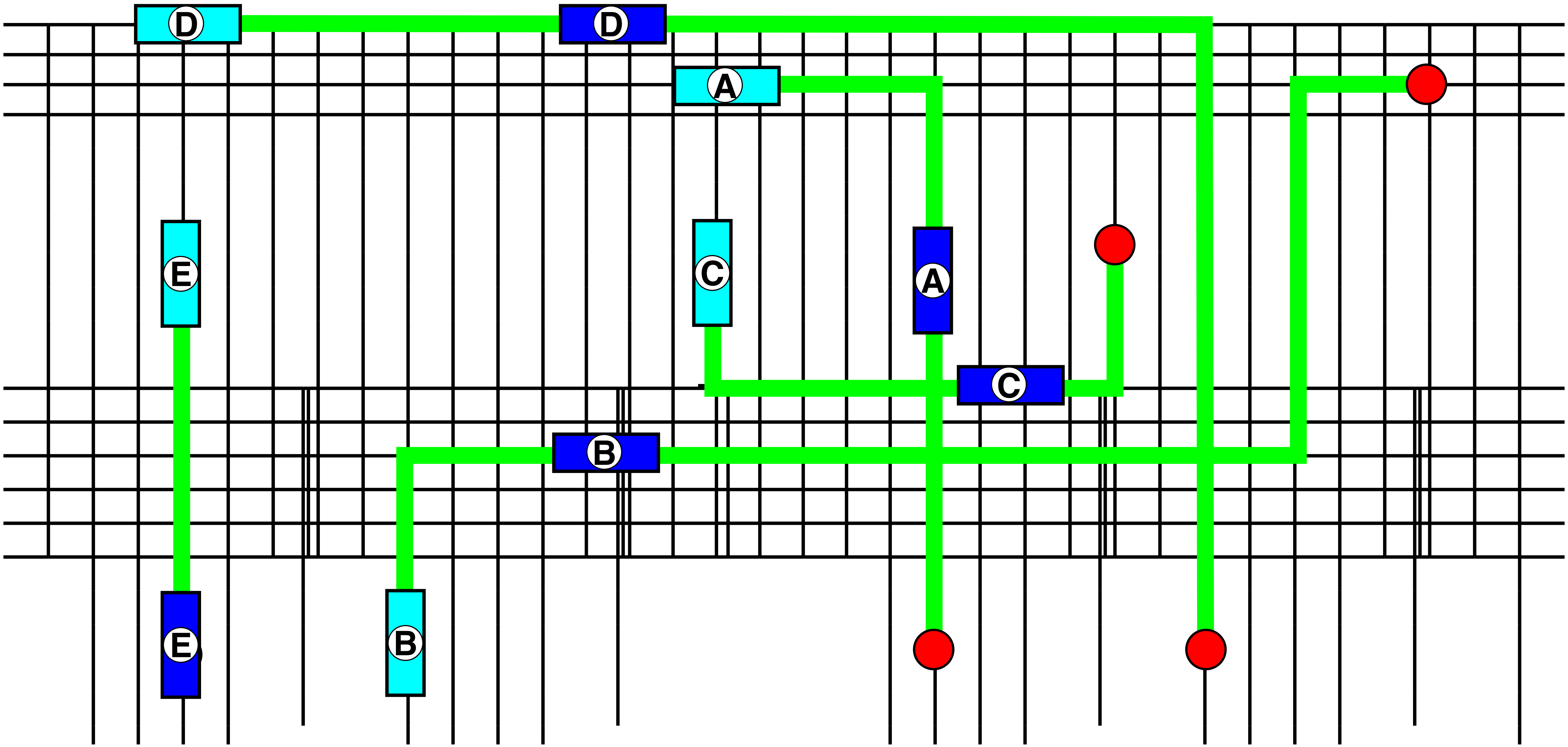
Example of a routing



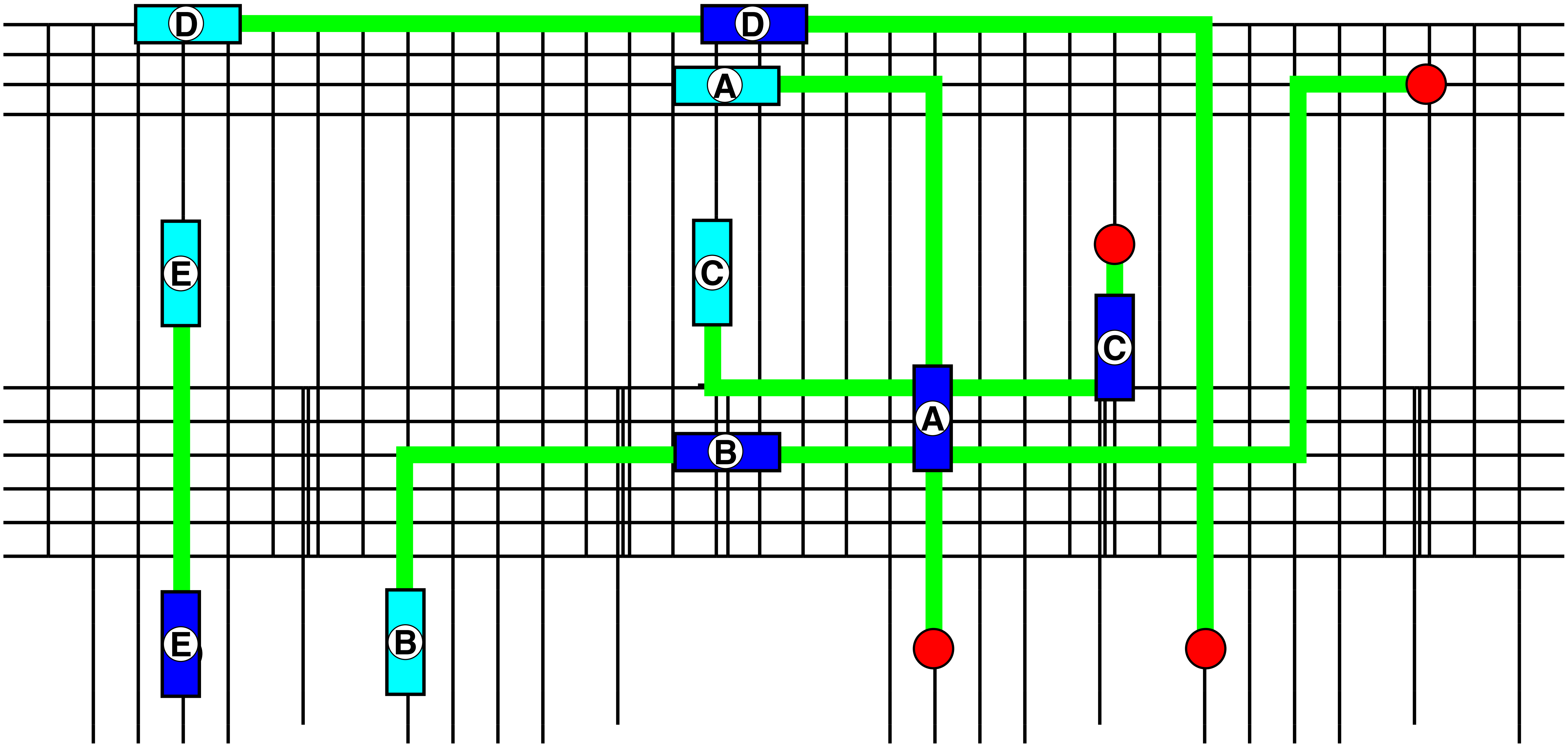
Example of a routing



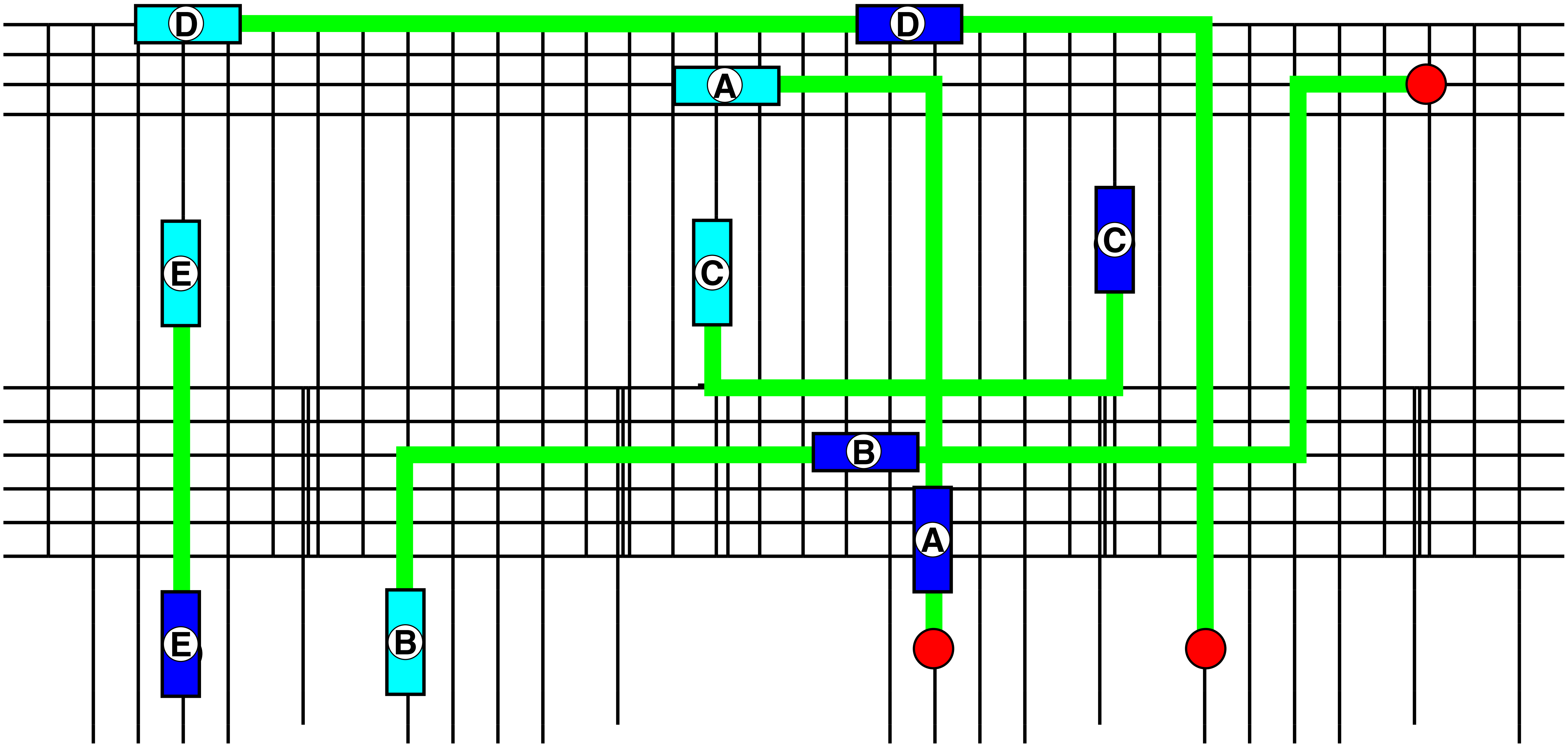
Example of a routing



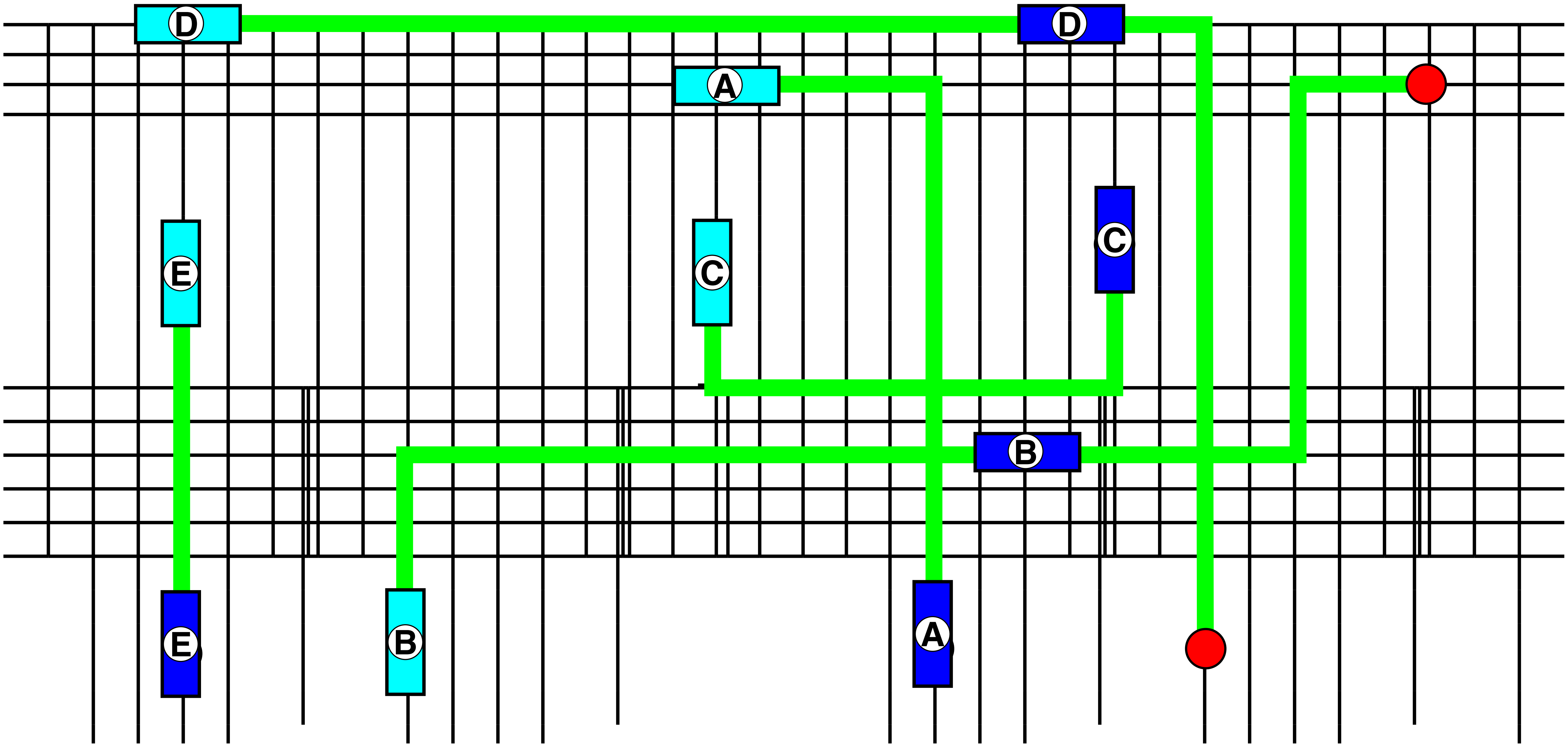
Example of a routing



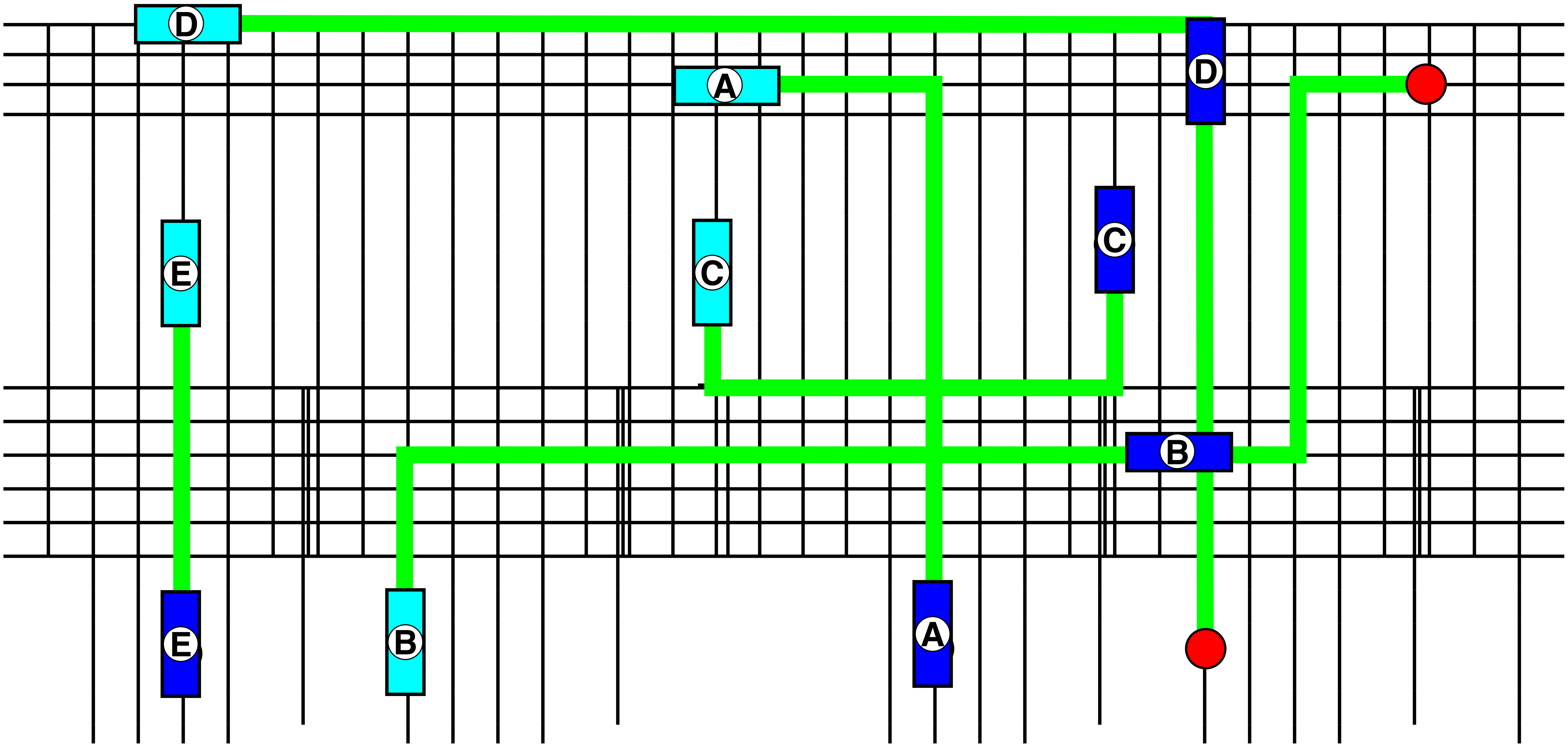
Example of a routing



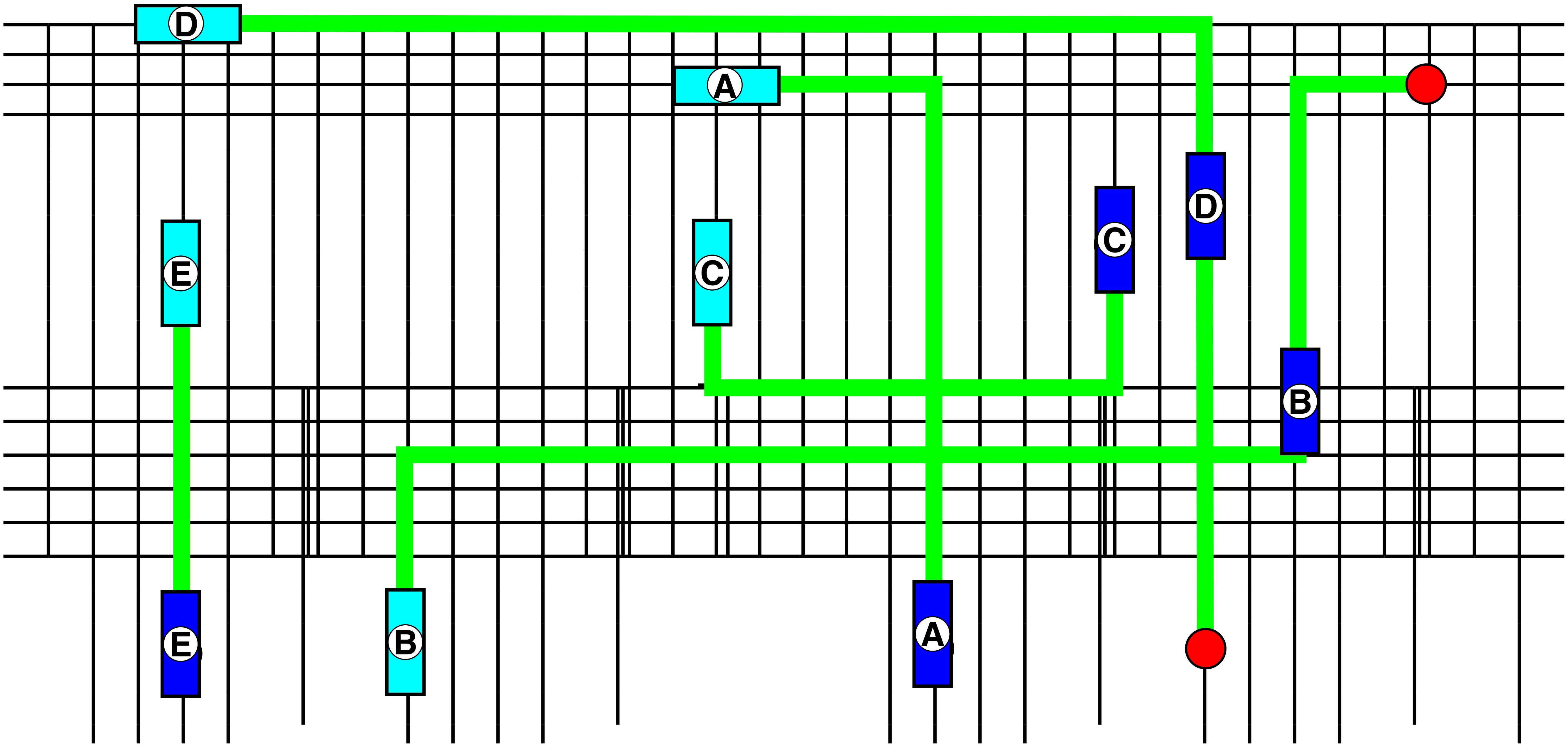
Example of a routing



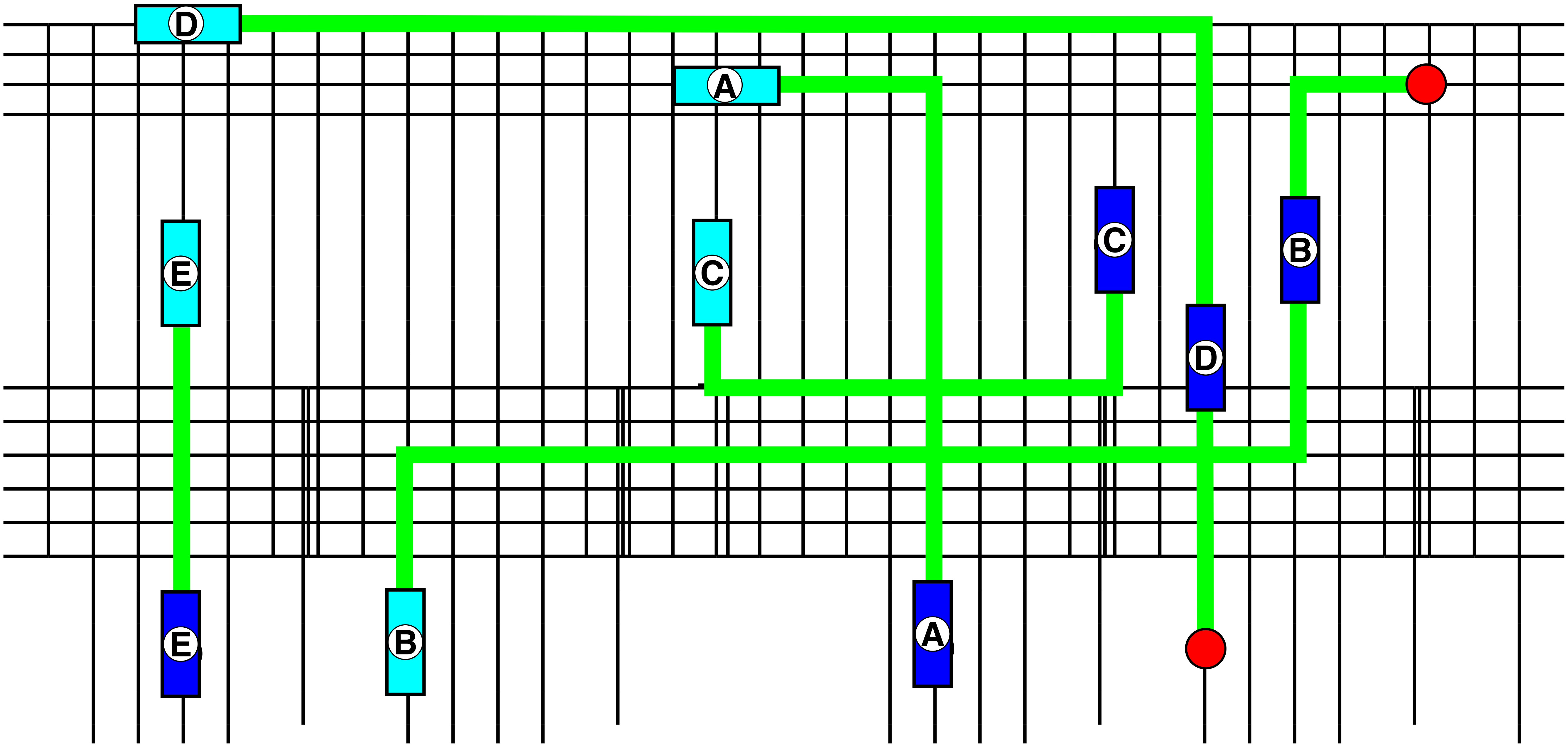
Example of a routing



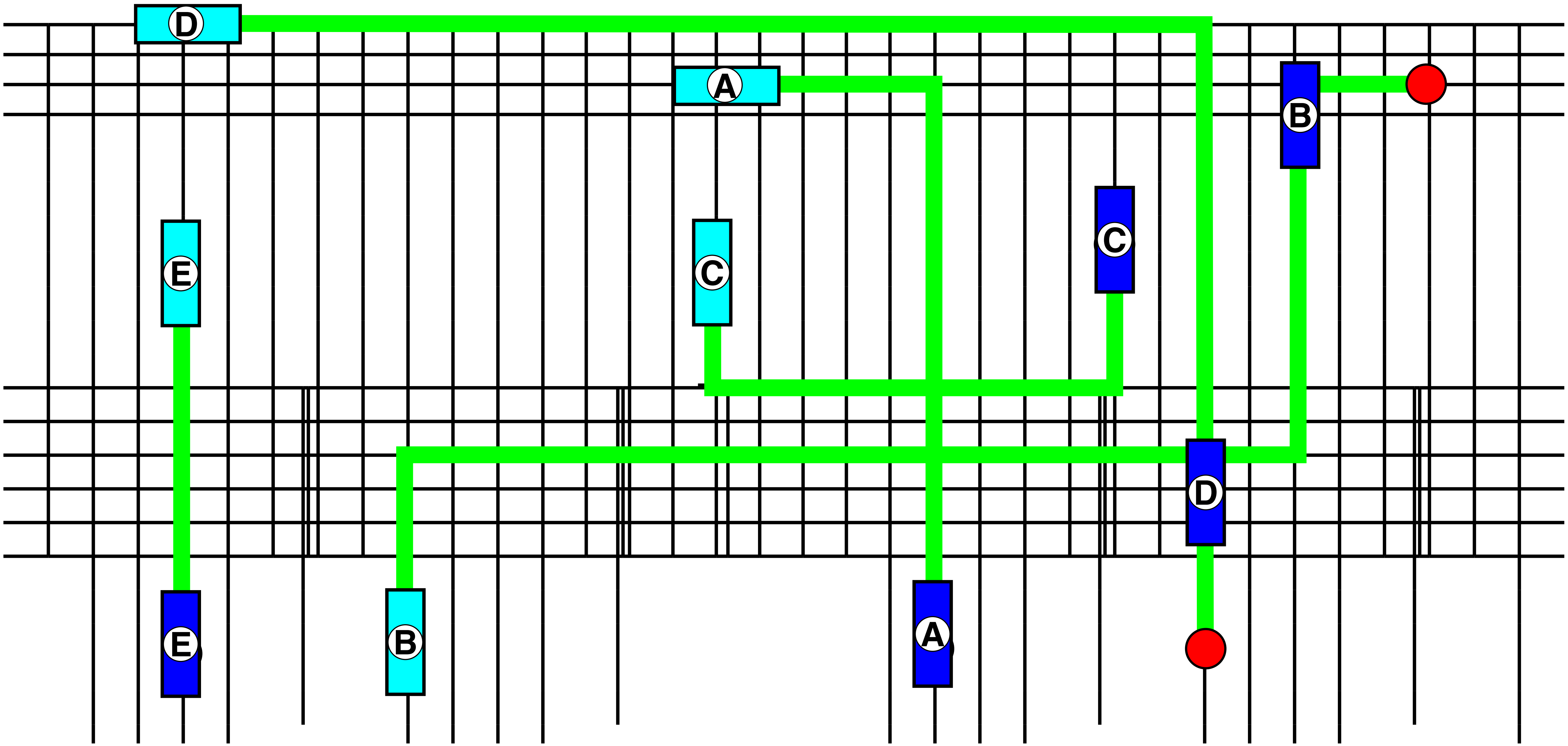
Example of a routing



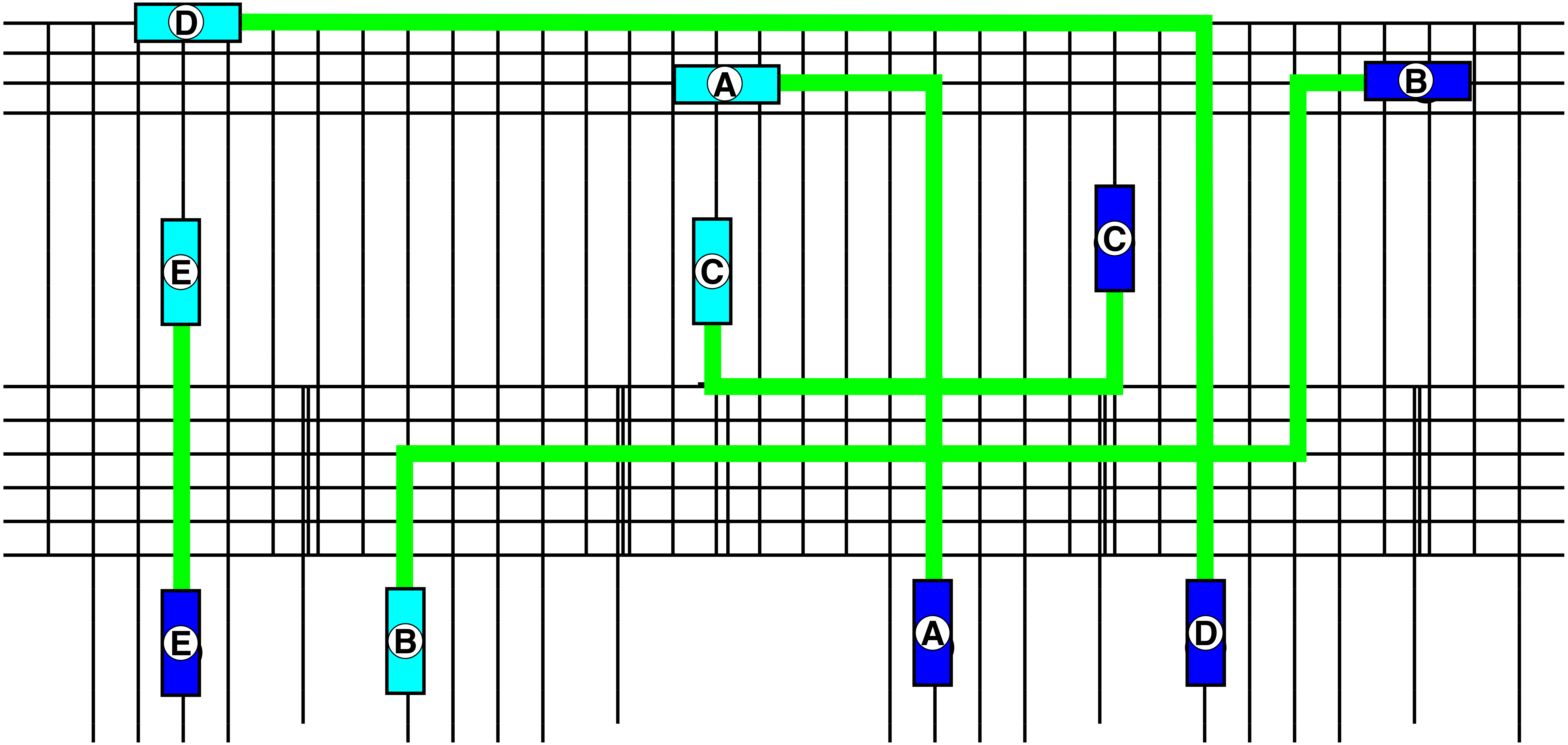
Example of a routing



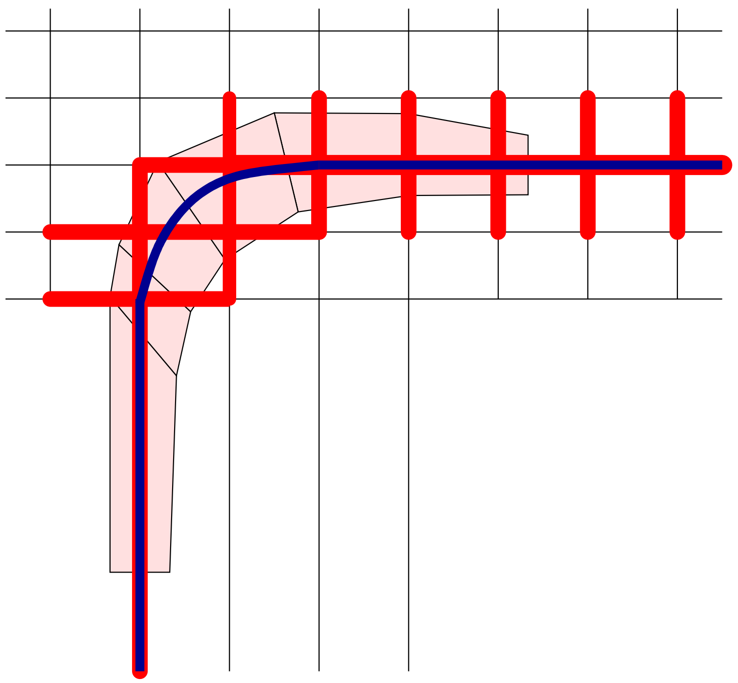
Example of a routing



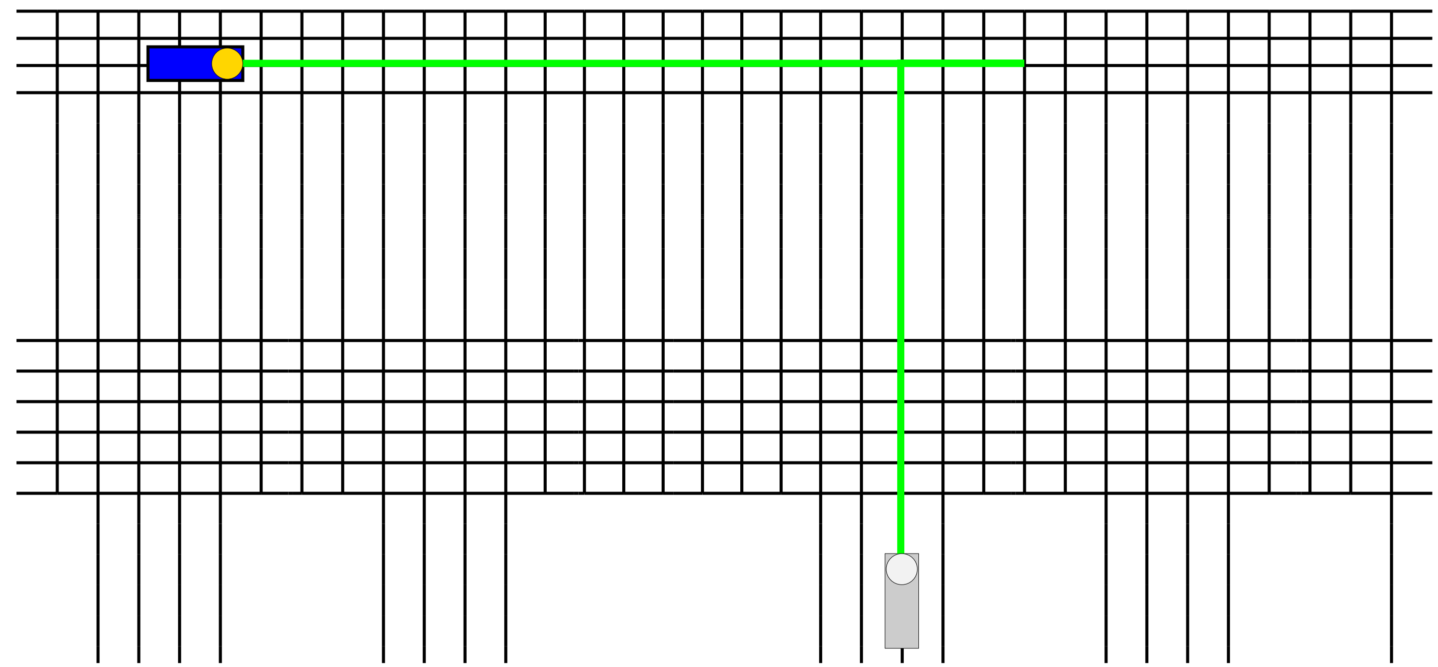
Example of a routing



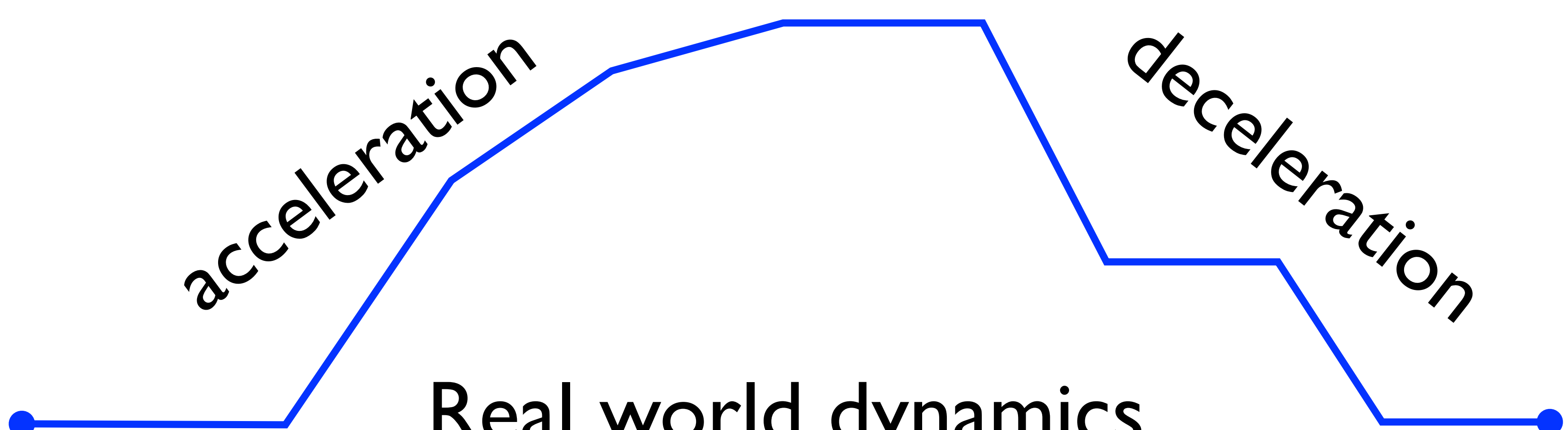
Complicating conditions



Turning behavior



Orientation at the destination



Real world dynamics

Overview

✓ Problem definition

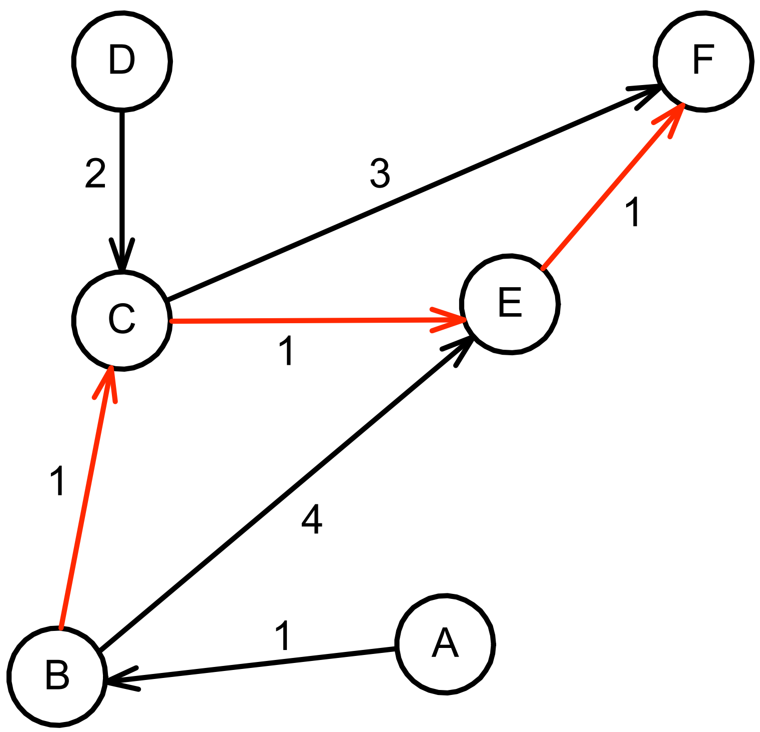
How to solve it?

It‘s easy for people

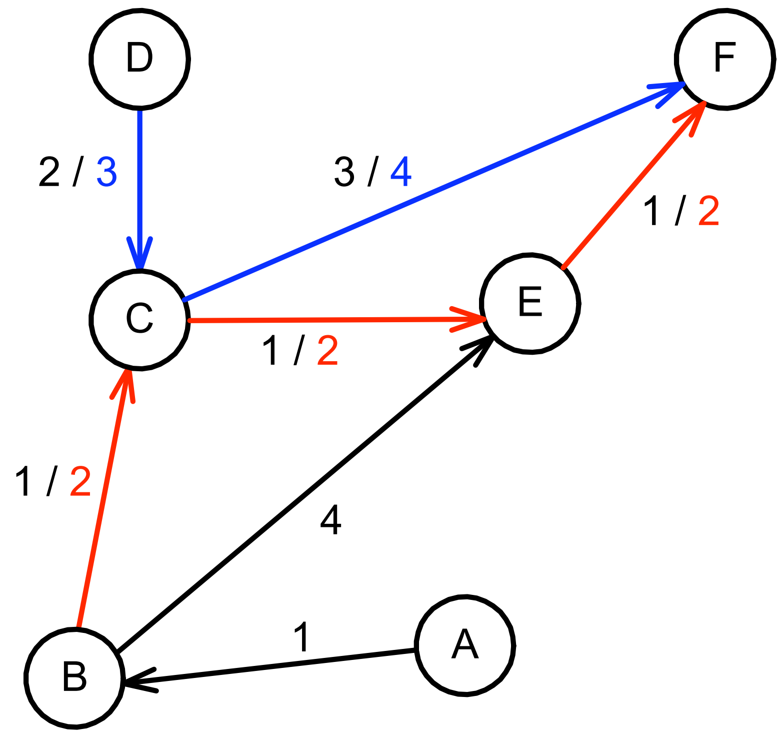
Overview

- ✓ Problem definition
- The current solution: static routing & deadlock avoidance
- Our approach: flows over time / dynamic routing
- Performance of our approach
- Theoretical foundation

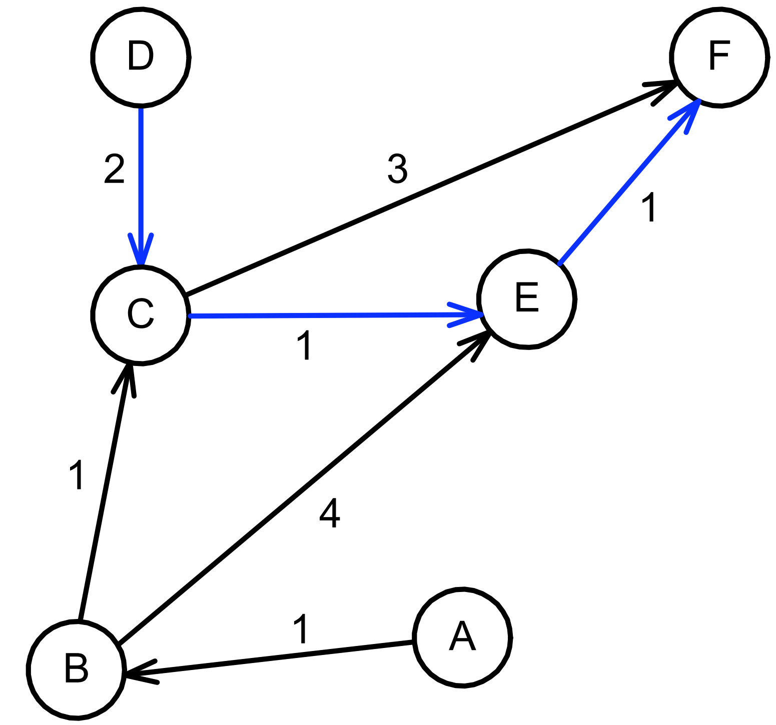
Used: static routing methods



compute shortest path in graph



penalize used arcs, compute new path

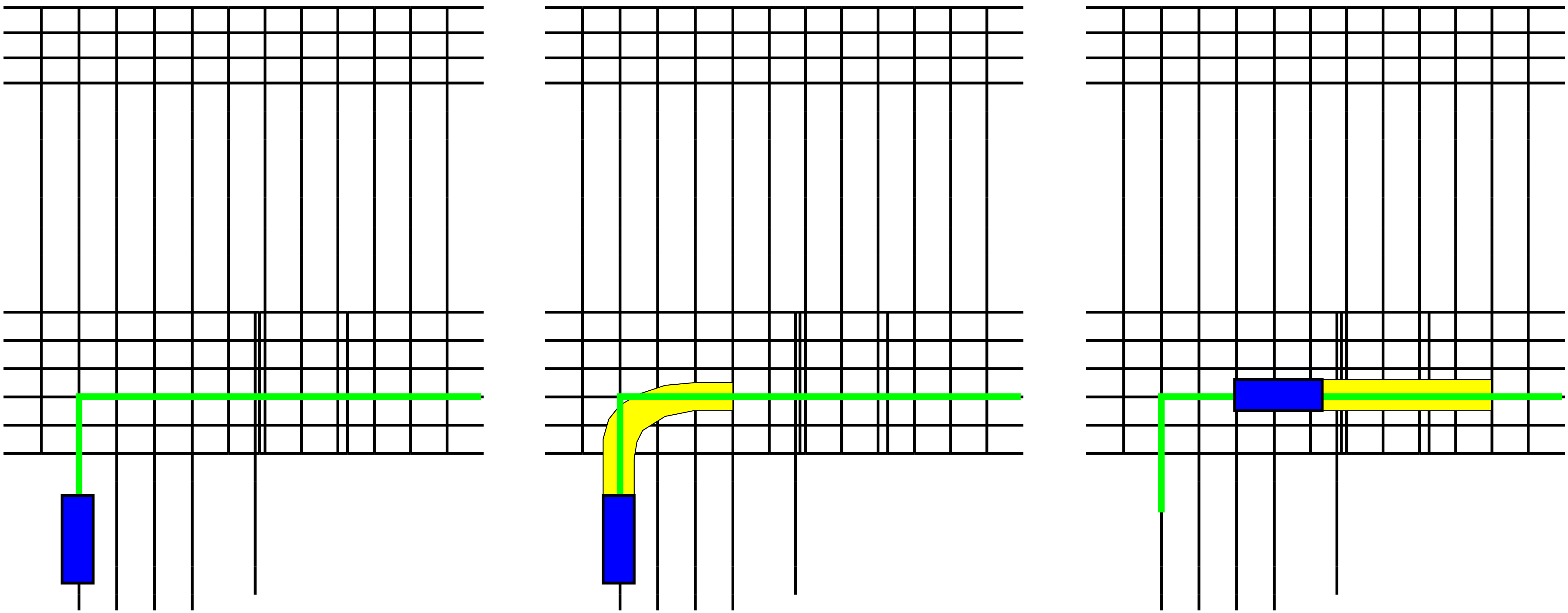


computed paths need not be shortest paths

Hope that this leads to only few collisions

Needs collision at run-time

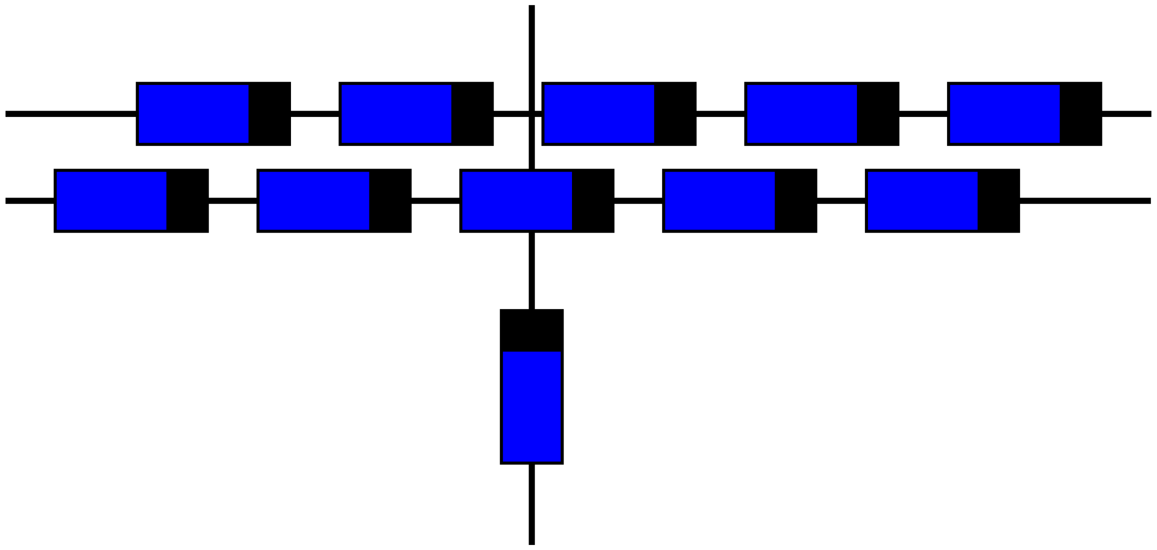
Reserve parts of a route exclusively for one AGV (Claiming)



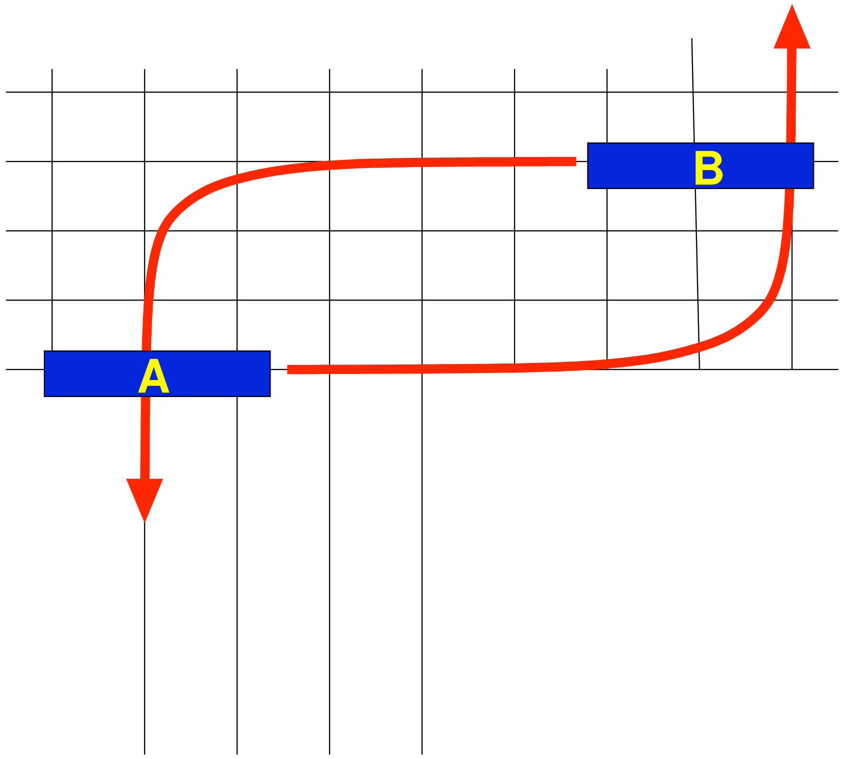
Compute route Claim 1 Claim 2

Disadvantages of claiming

1. no guaranteed arrival times



2. livelocks

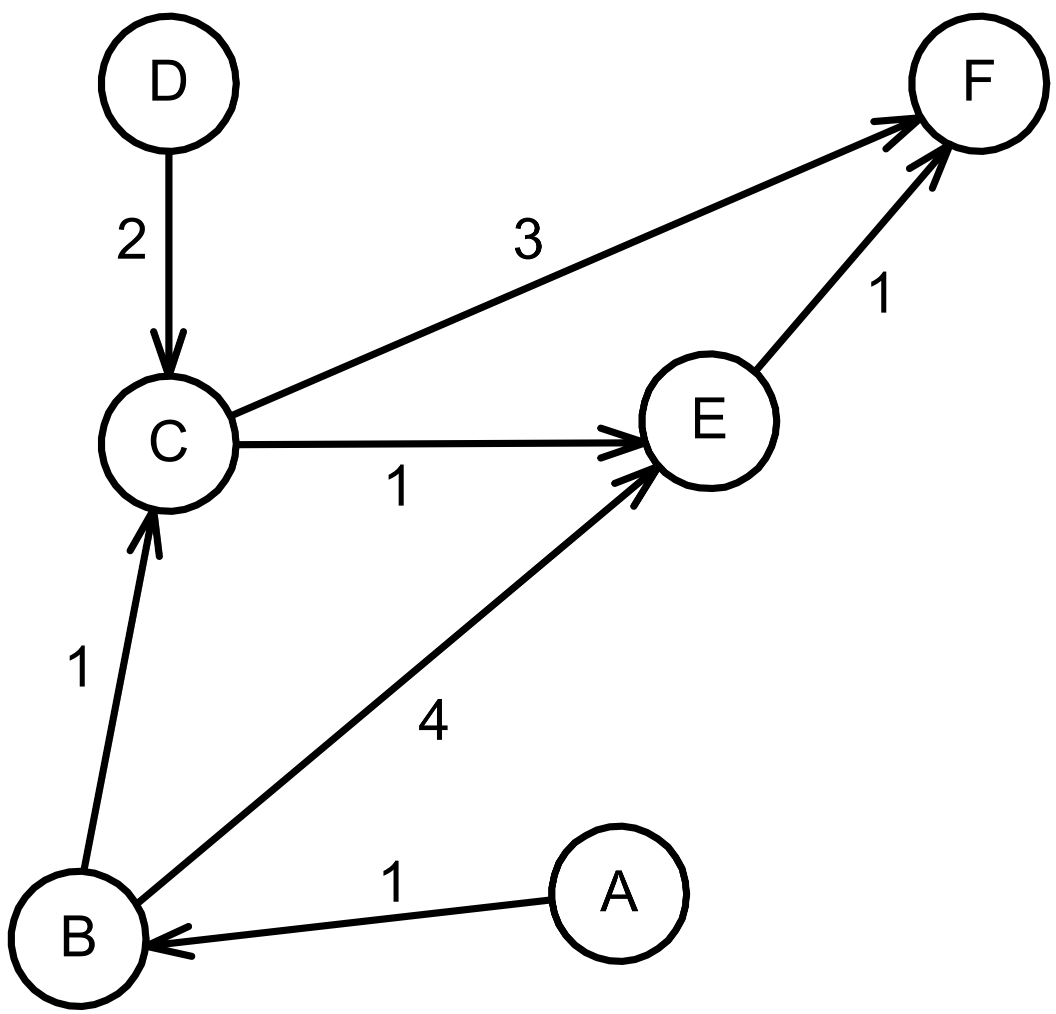


3. deadlocks

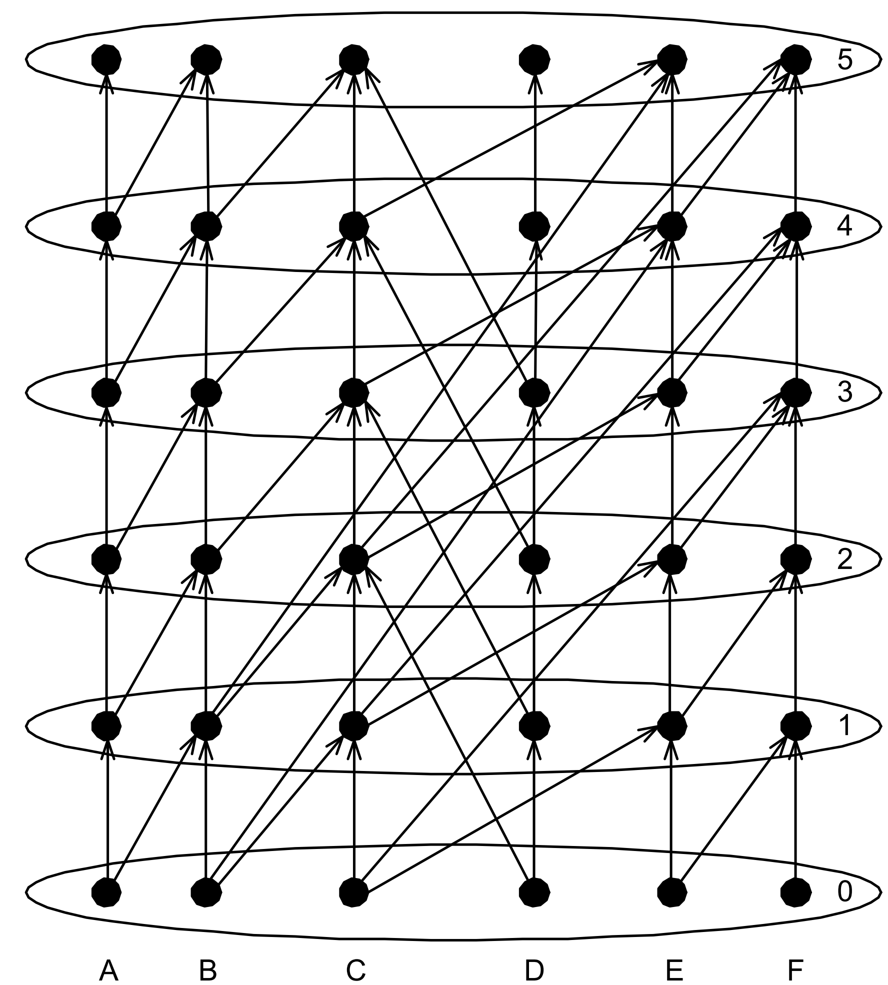
Overview

- ✓ Problem definition
- ✓ The current solution: static routing & deadlock avoidance
- Our approach: flows over time / dynamic routing
- Performance of our approach
- Theoretical foundation

Our approach: flows over time

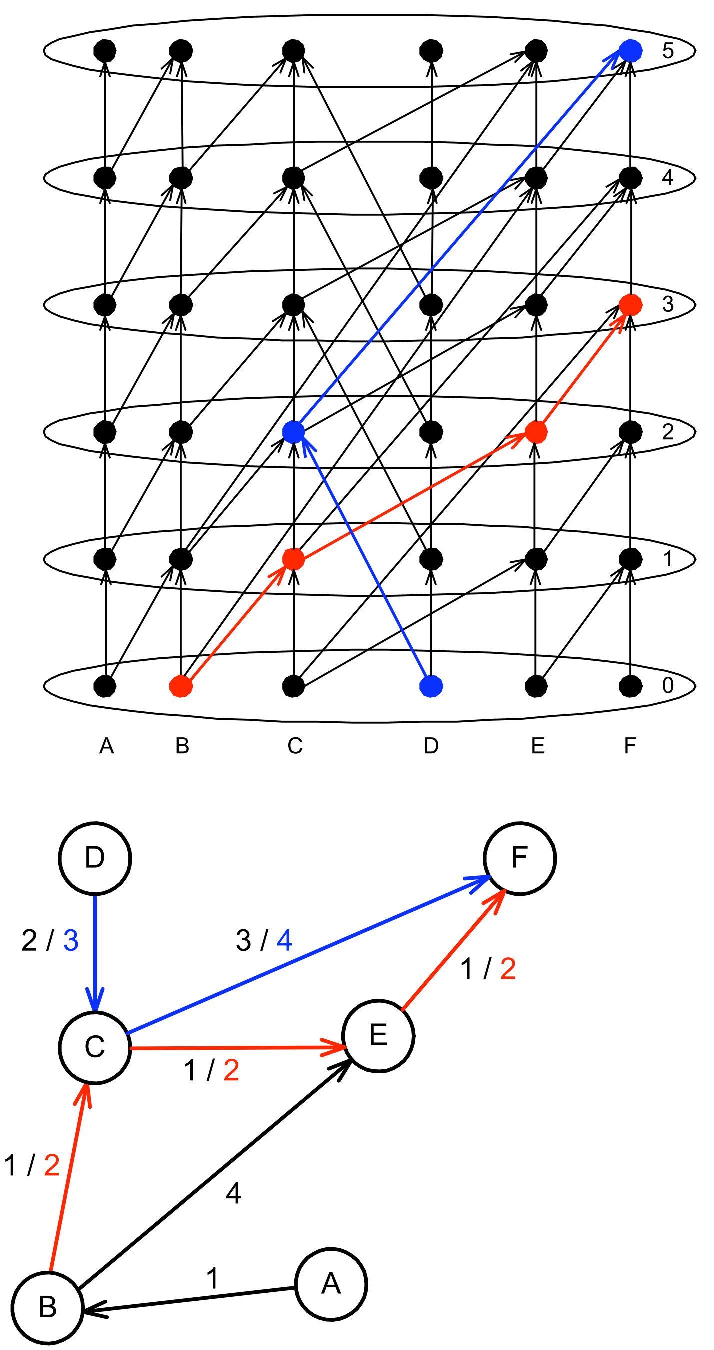


Graph

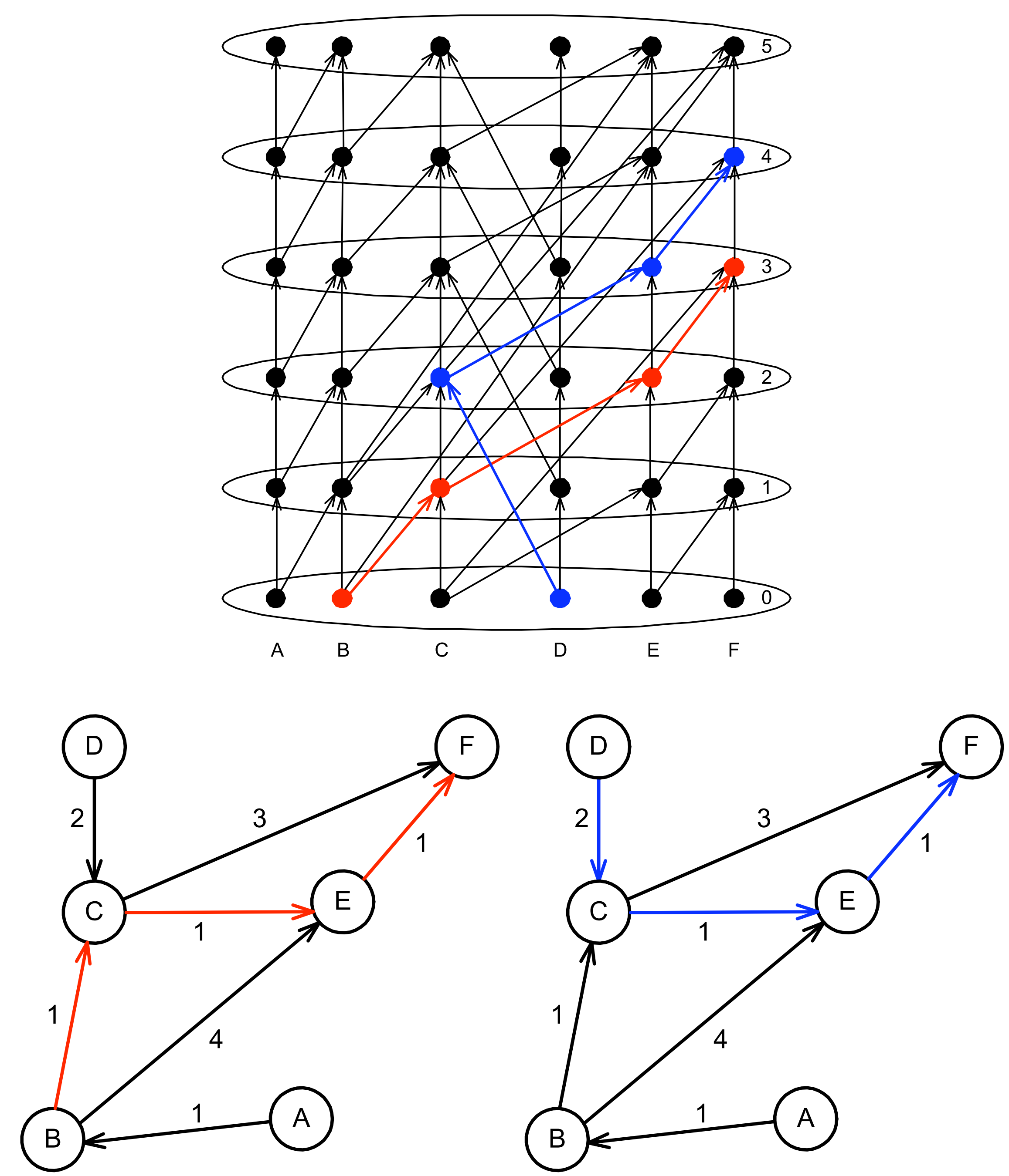


Time-expanded graph

Time expansion permits collision control

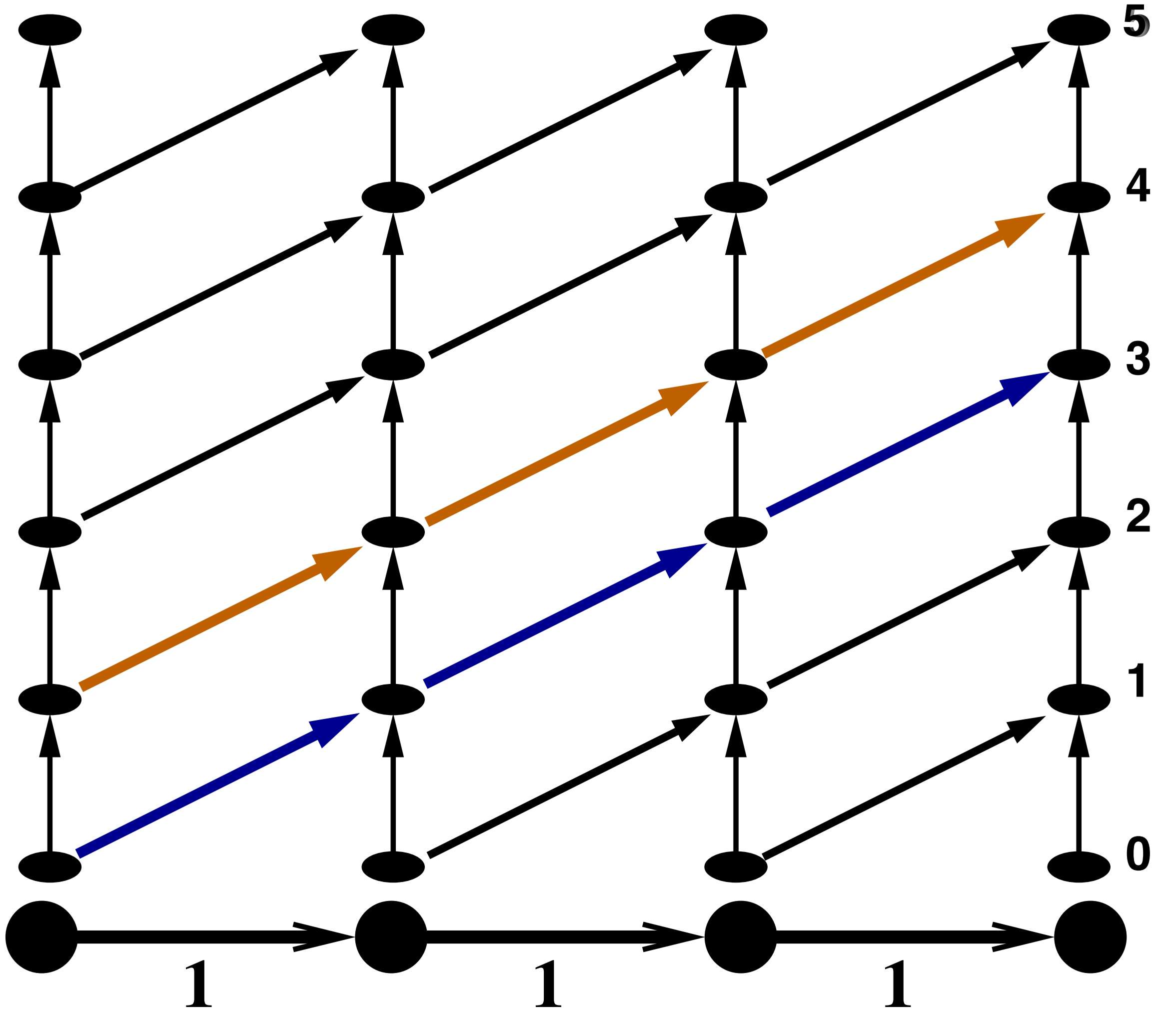


static routing, $T = 5$

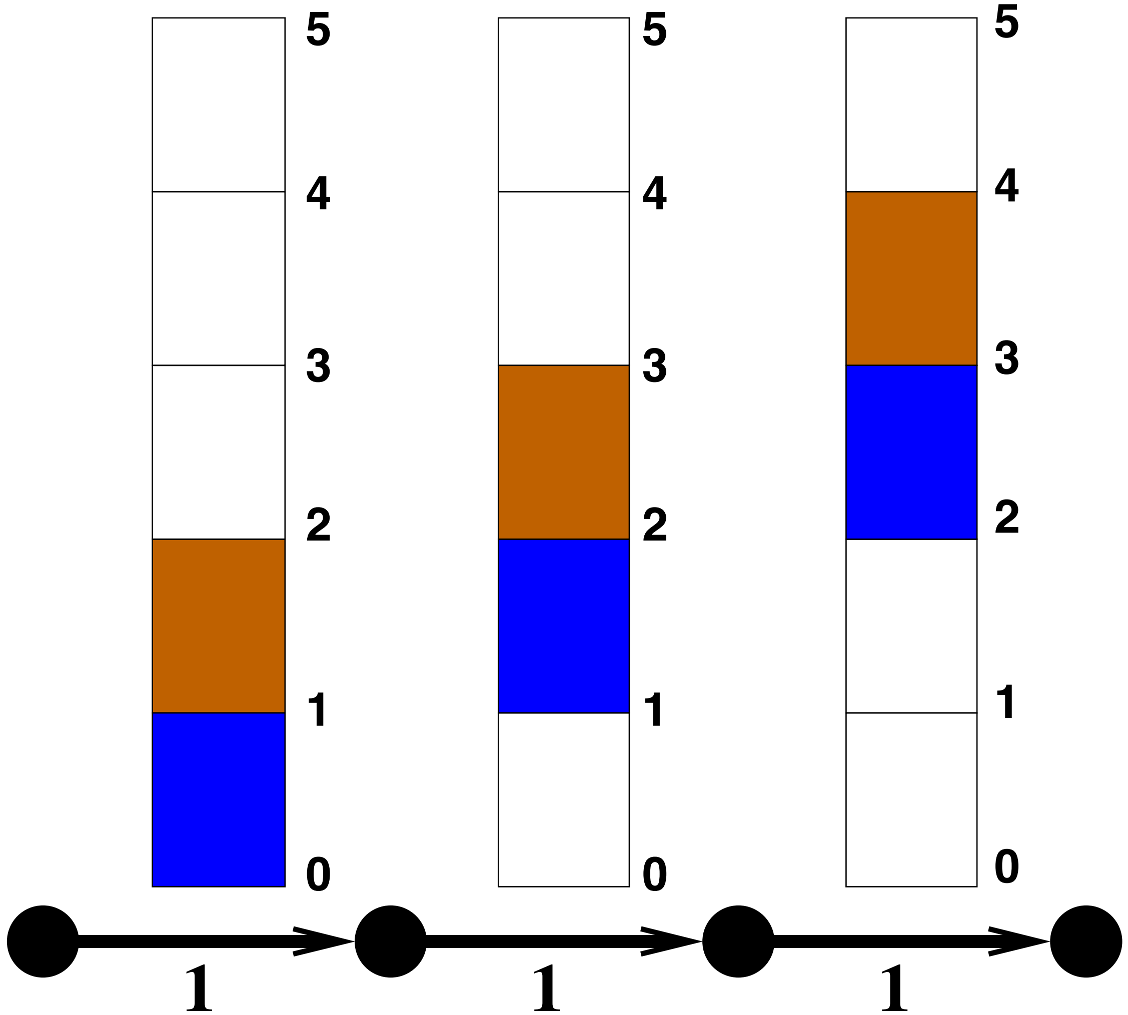


dynamic routing, $T = 4$

Modeling the time expansion



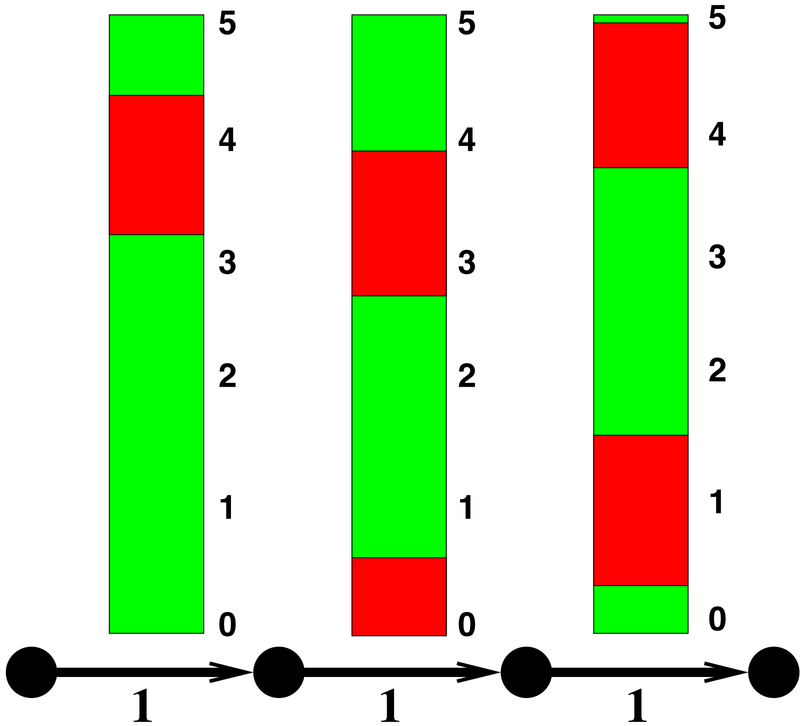
explicit time expansion



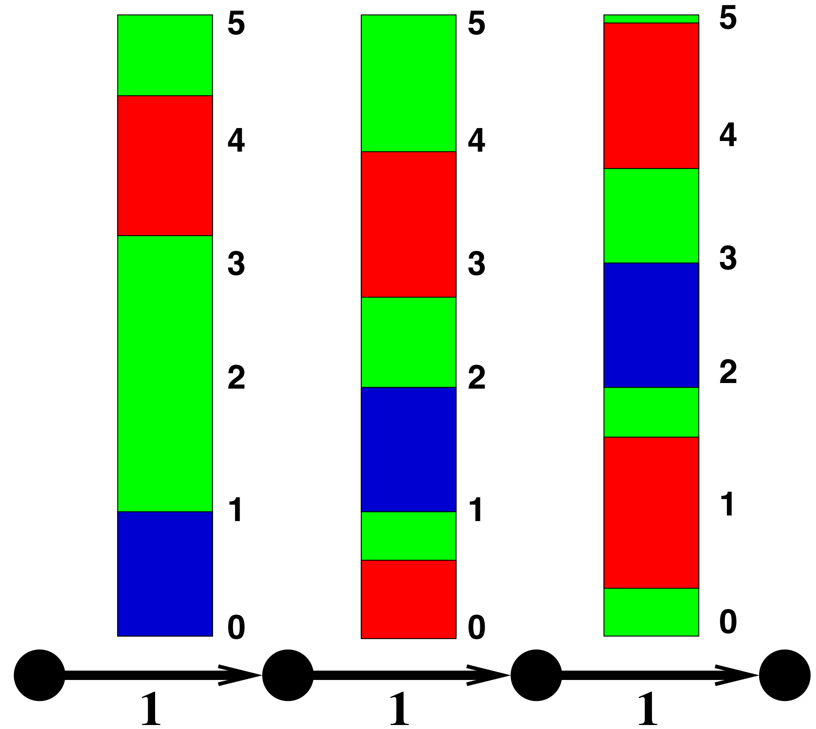
implicit time expansion

Shortest paths with time-dependent blockings

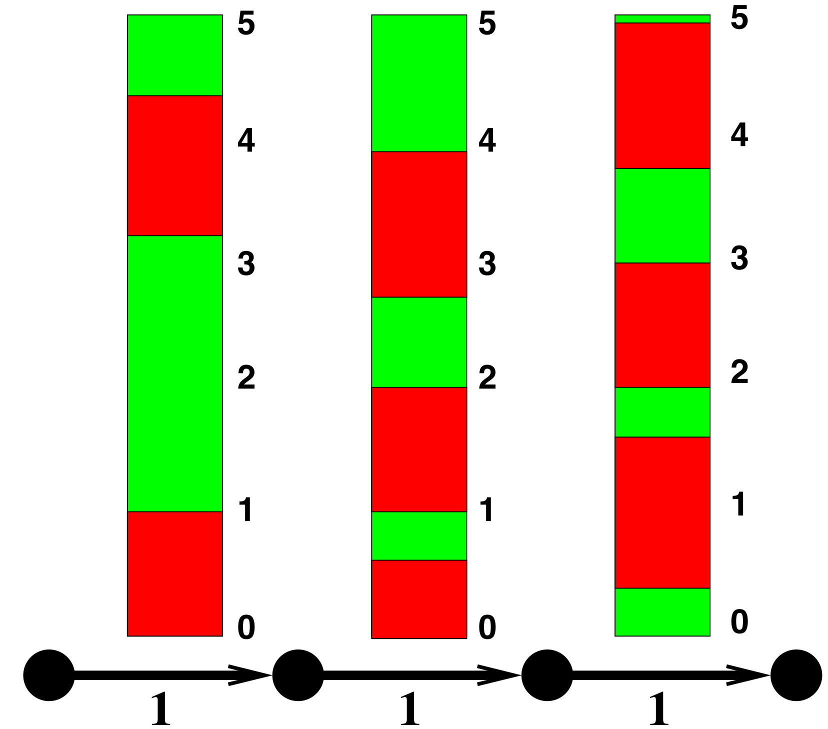
modeled by implicit time expansion



Graph with blockings



New path compatible with blockings



Updated blockings

Known: shortest paths with time windows

J. Desrosier, Y. Dumas, M. Solomon, F. Soumis:
Time Constrained Routing and Scheduling
in: Handbook in Operations Research and
Management Science Vol. 8
Chapter 2: Network Routing, pp. 35 - 139
Elsevier 1995

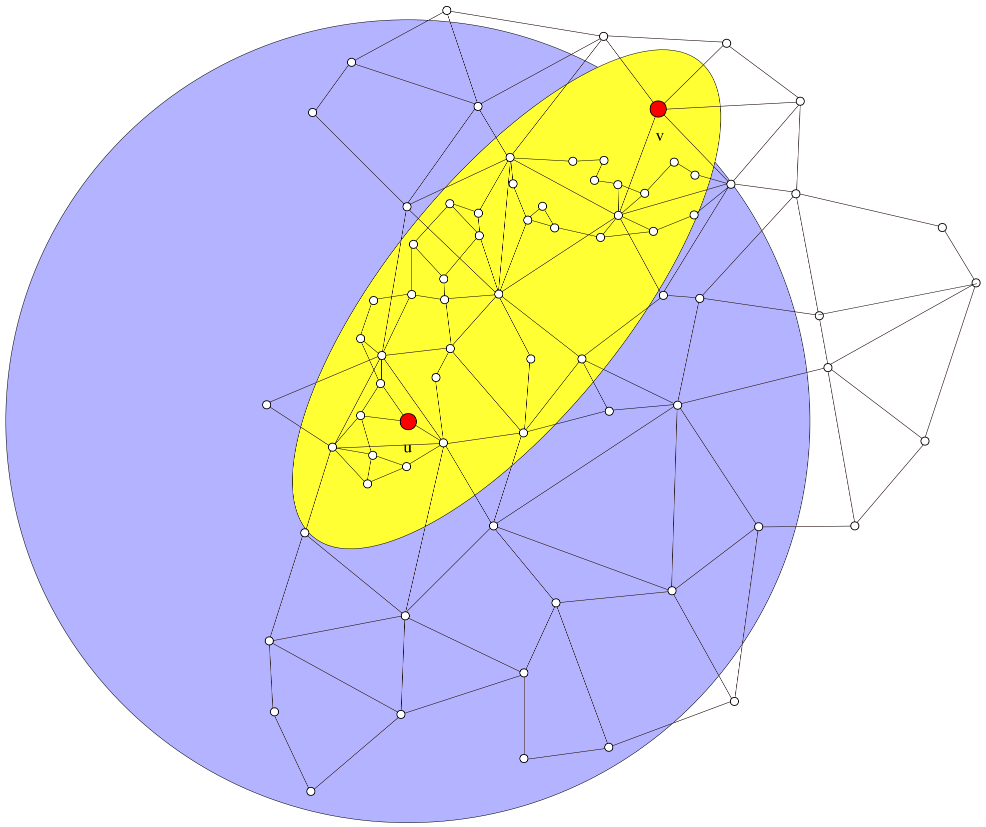
- Given: Graph $G = (V,E)$ with cost c_a, travel time τ_a and time windows F_a^i on every arc a
- Wanted: Shortest path w.r.t. cost c_a that respects the time windows w.r.t. the τ_a
- algorithmically difficult (NP-hard)
- easy here, as $c_a = \tau_a + \textit{time spent waiting}$

Overview

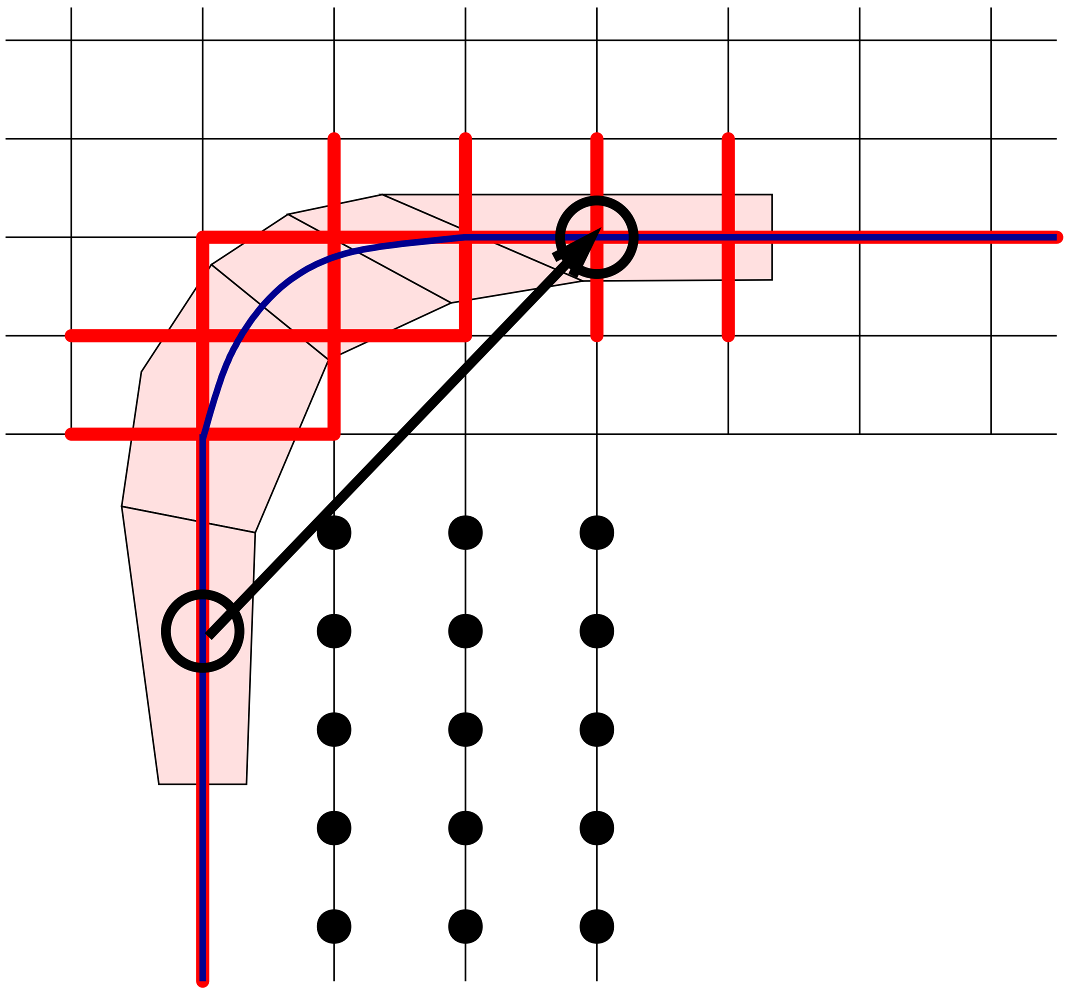
- ✓ Problem definition
- ✓ The current solution: static routing & deadlock avoidance
- ✓ Our approach: flows over time / dynamic routing
- Performance of our approach
- Theoretical foundation

Efficient algorithm

- Generalizes Dijkstra's algorithm
- Polynomial run time and very fast in practice
- Works also w.r.t. orientation and turning behavior of the AGVs
- Additional speed up by goal-directed search

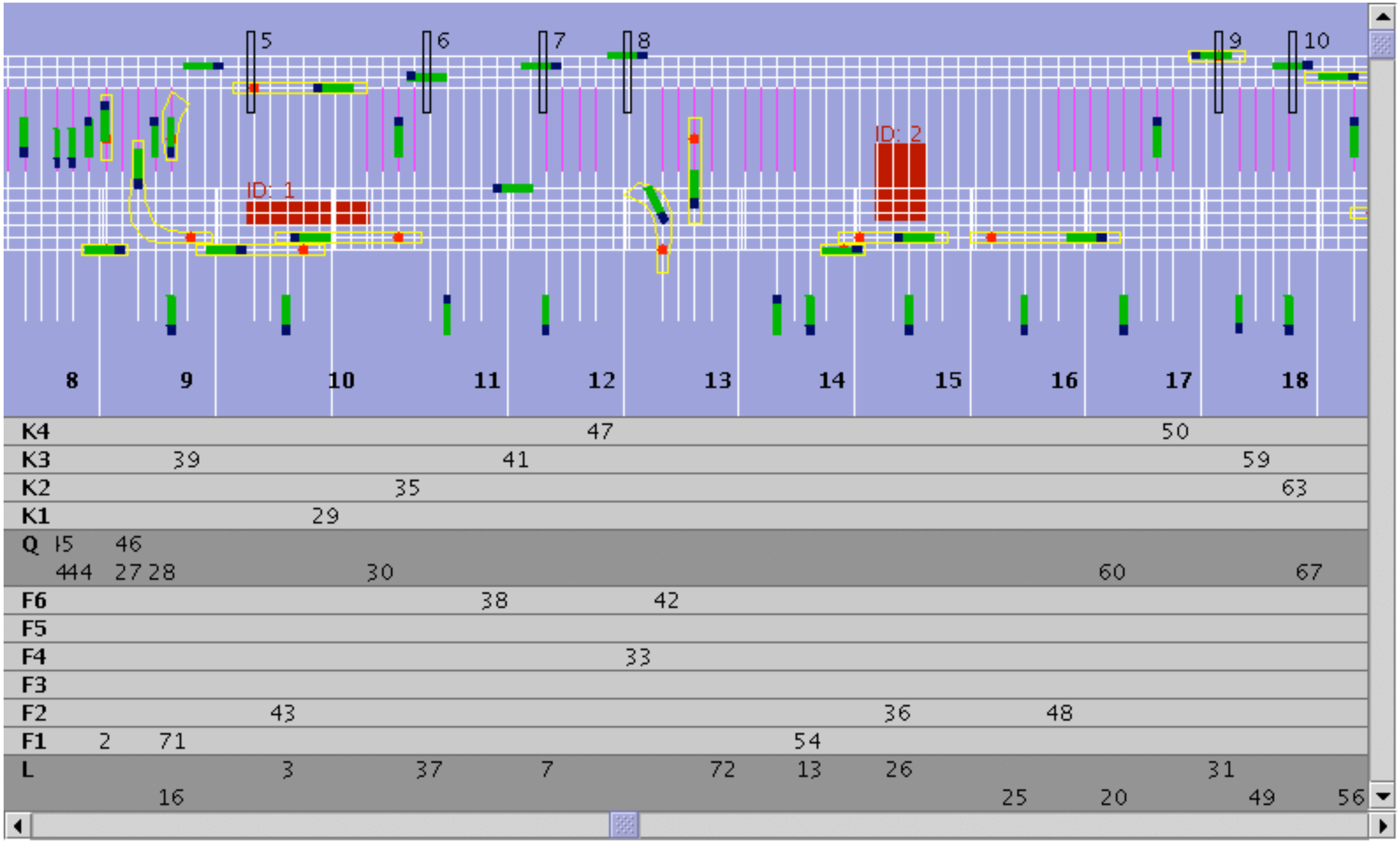


Details of the AGV layer

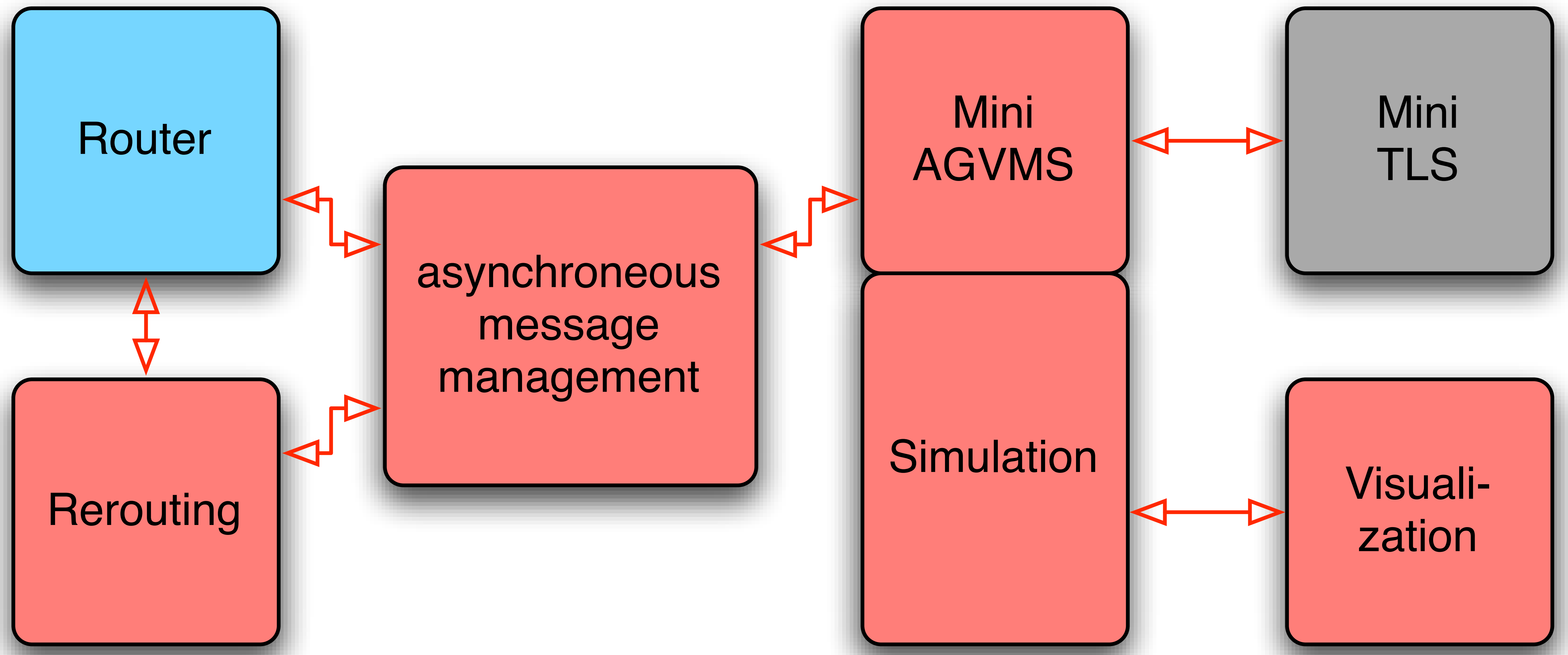


- Model turning behavior through preprocessing
- Increases graph size to 5,445 vertices and 43,324 arcs

Dynamic routing in action

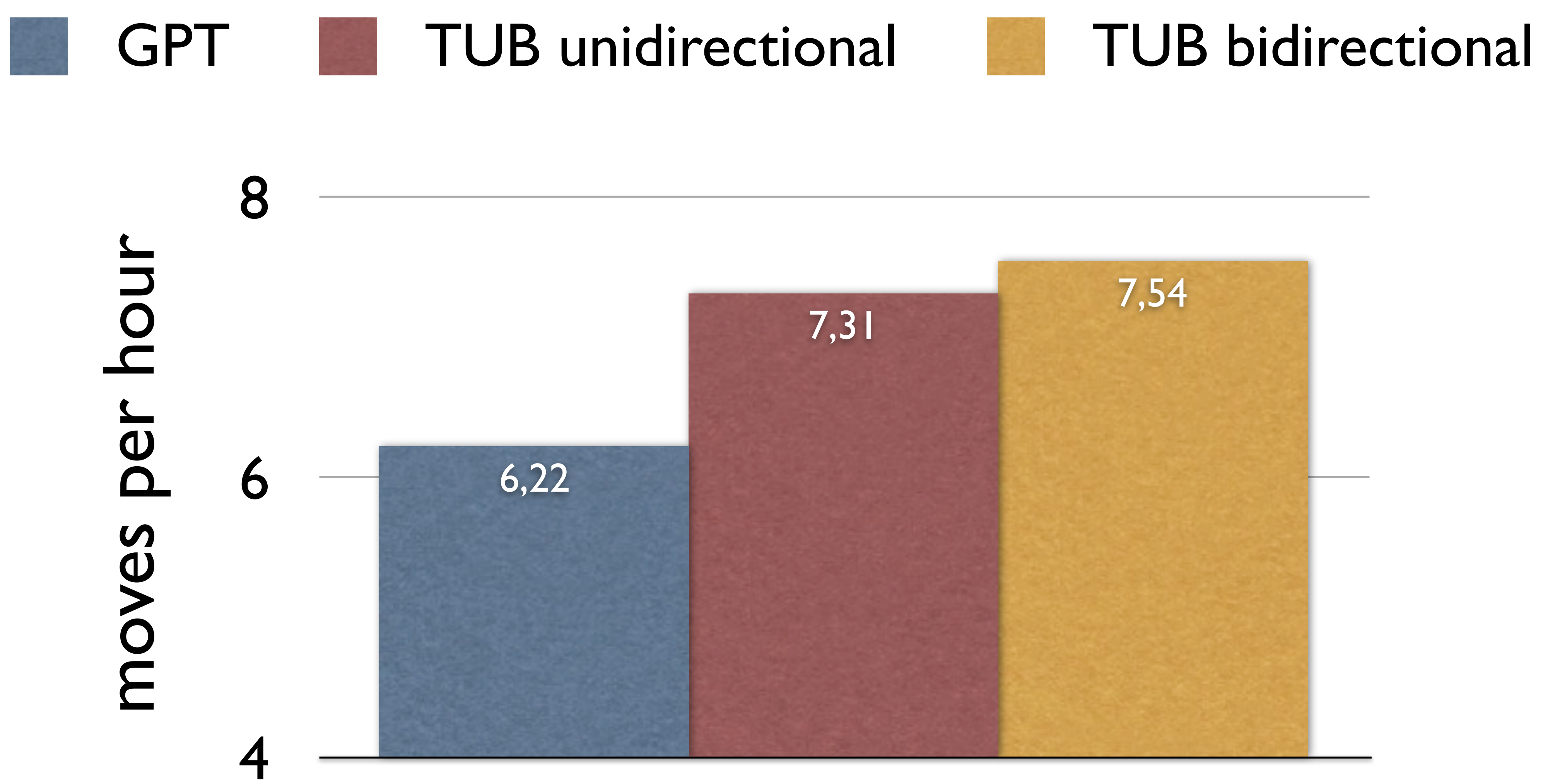


Architecture of the whole system



- AGVMS = AGV Management System
- TLS = Terminal Leit System

Results for 79 AGV scenario - overview



- 20% improvement in high traffic scenarios

Summary

- Main features
  - Dynamic routing is fast
  - Avoids deadlocks and livelocks, leads to shorter total travel time in dense scenarios
  - Guarantees reliable arrival times of the AGVs, a clear pro for subsequent planning steps in the logistic chain
- Can also
  - Handle disturbances (on the fly rerouting)
  - Incorporate priorities for specified AGVs
- HHLA bought our software in 2009

Overview

- ✓ Problem definition
- ✓ The current solution: static routing & deadlock avoidance
- ✓ Our approach: flows over time / dynamic routing
- ✓ Performance of our approach
- ▶ Theoretical foundation

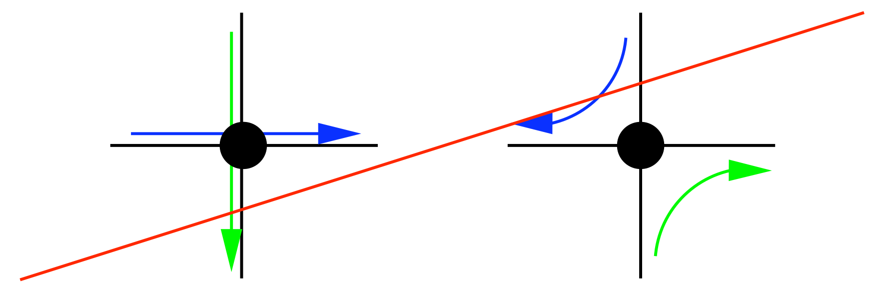
Detailed analysis

- Why route AGVs sequentially?
- What is the sequential gap?
- Can the static approach be improved?
- Is it competitive?

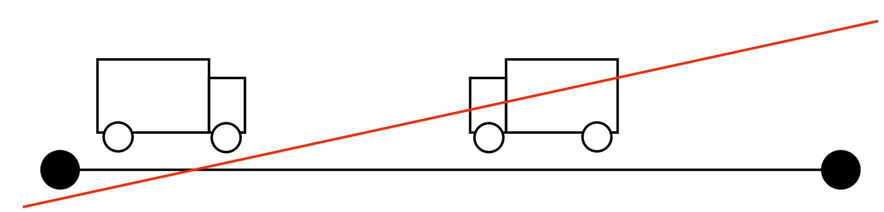
[Ewgenij Gawrilow, Max Klimm, R. M., Björn Stenzel]
EURO J Transp Logist 2012

Dynamic grid flows

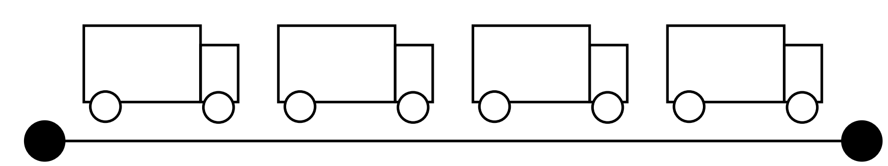
- integer multi-commodity flows
- discrete time



- undirected grid graphs
  - constant transit times (usually 1 per grid arc)

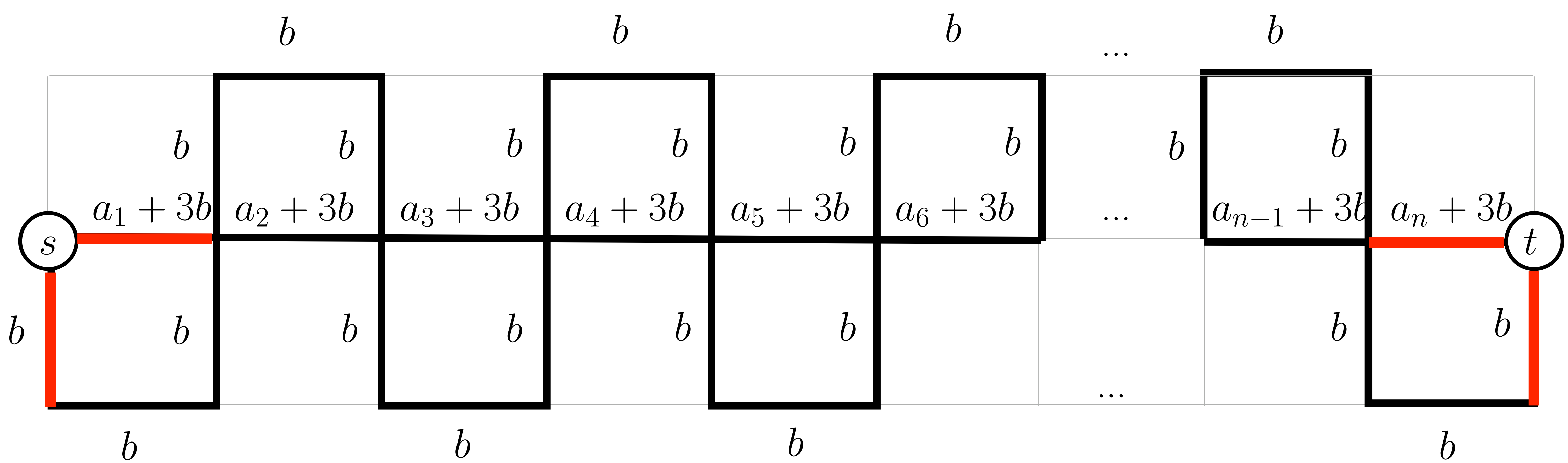


  - arc capacities 1
- no simultaneous bends, crossings, bidirectional use
- one vehicle per time unit



Problem without waiting is NP-hard

Reduction from PARTITION



red intervals = blockings

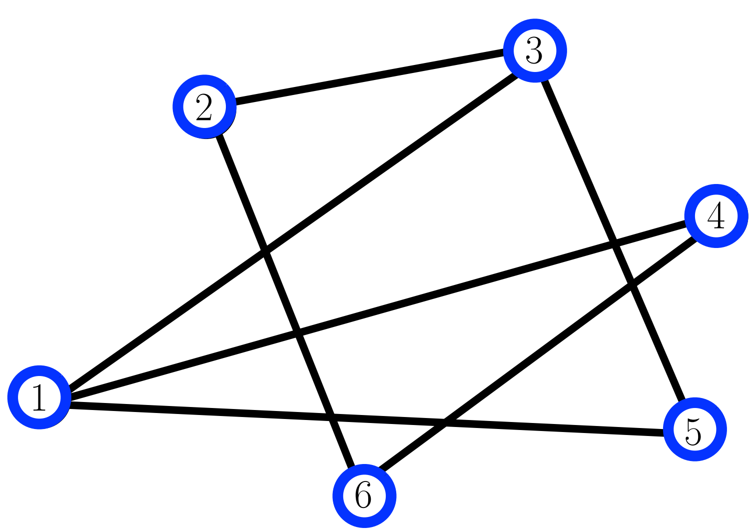
Can one route a demand of 1 in time $T = \sum_{i=1}^{n} \frac{a_i}{2}$?

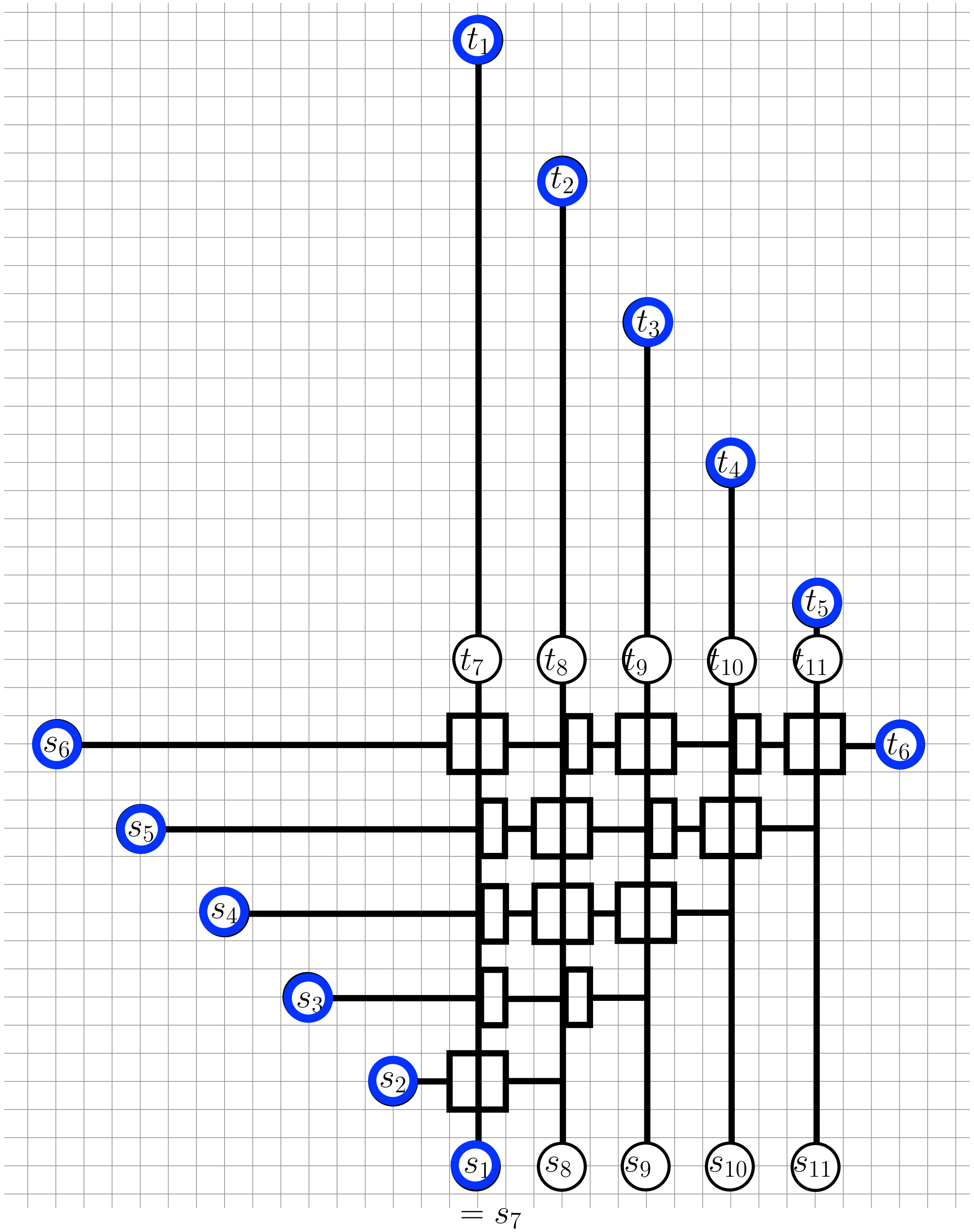
Related result by Busch, Magdon-Ismail, Mavronicolas, Spirakis in the context of direct packet routing, 2004

Multicommodity problem is strongly NP-hard

with and without time-windows

reduction from 3-VERTEX-COLORING

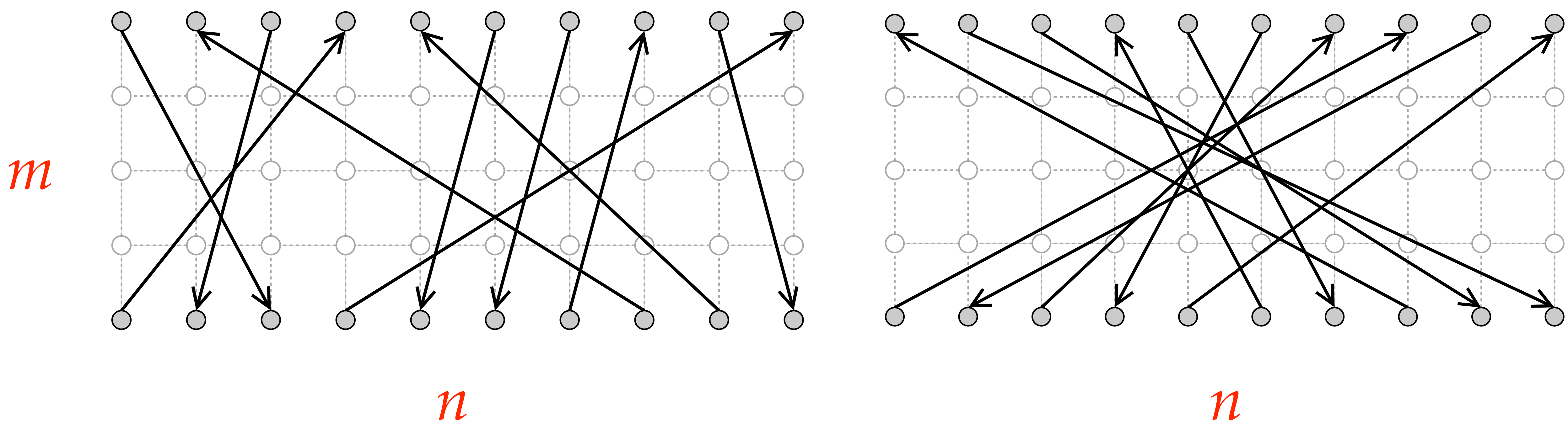




ideas from Busch, Magdon-Ismail, Mavronicolas, Spirakis 2004

The sequential gap

- How much do we loose by restricting to sequential routing?
  - Empirical tests via multcommodity flows in time expanded graphs
  - use $m \times n$ grid graphs as routing graph



Sequential gap decreases with m

Grid	mean	max	stand. dev
2 x 10	10.62%	28.57%	6.62
3 x 10	4.64%	13.19%	2.83
4 x 10	2.54%	9.00%	2.02
6 x 10	0.90%	4.17%	0.97
2 x 20	15.77%	31.67%	8.22
3 x 20	7.36%	17.13%	3.43
4 x 20	3.68%	6.33	1.33

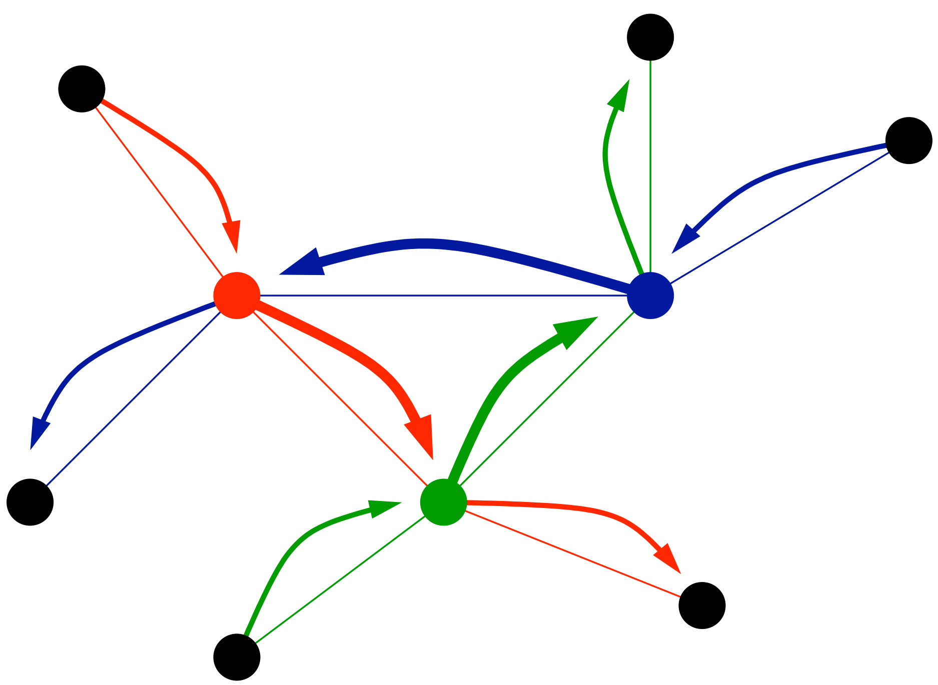
for *total travel time* (similar results for *makespan*)

Better static routing algorithms

- First step
  - compute short static routes that create few collisions at run time in an online fashion
  - load balancing with length bounds on paths
- Second step
  - compute a deadlock-free schedule for these routes
  - is NP hard, can be done by the Colorful Path Problem [Alon, Yuster & Zwick ,96]

Deadlock characterization & detection

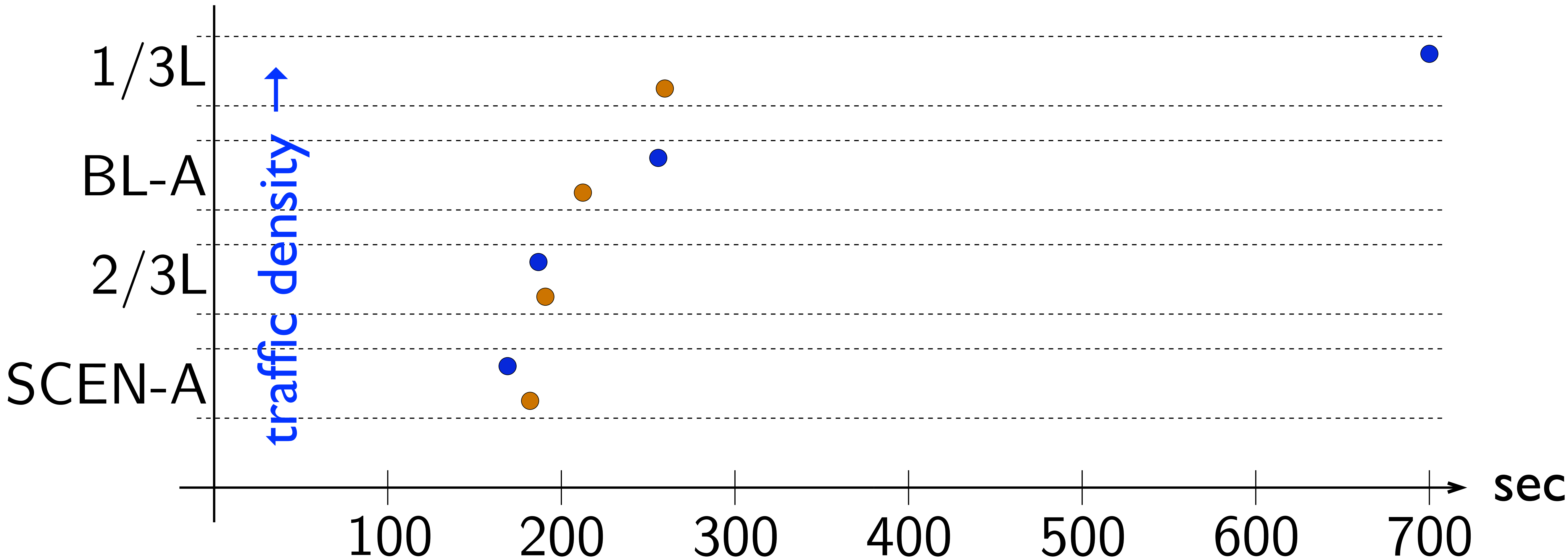
Deadlock = colorful directed cycle in deadlock detection graph



= directed cycle such that every color appears at most once
[Alon, Yuster & Zwick ,96]

- Deadlock detection is NP-complete
- Have $O(c \cdot 2^{c} \cdot |E|)$ algorithm (c = number of colors) from [Alon, Yuster & Zwick]

The static algorithm in practice

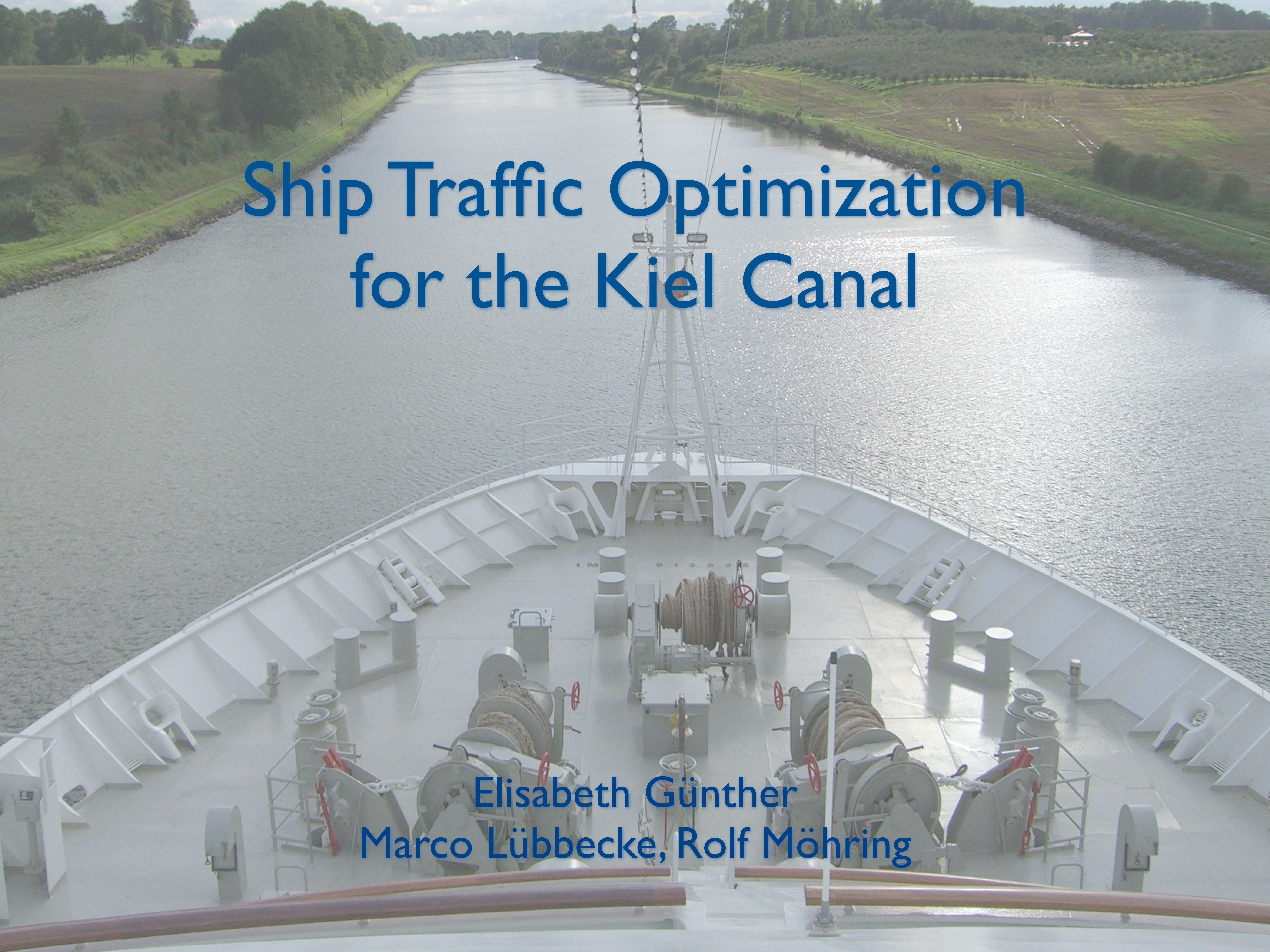


- average travel time with dynamic router (sec)
- average travel time with static router (sec)

Summary additional analysis

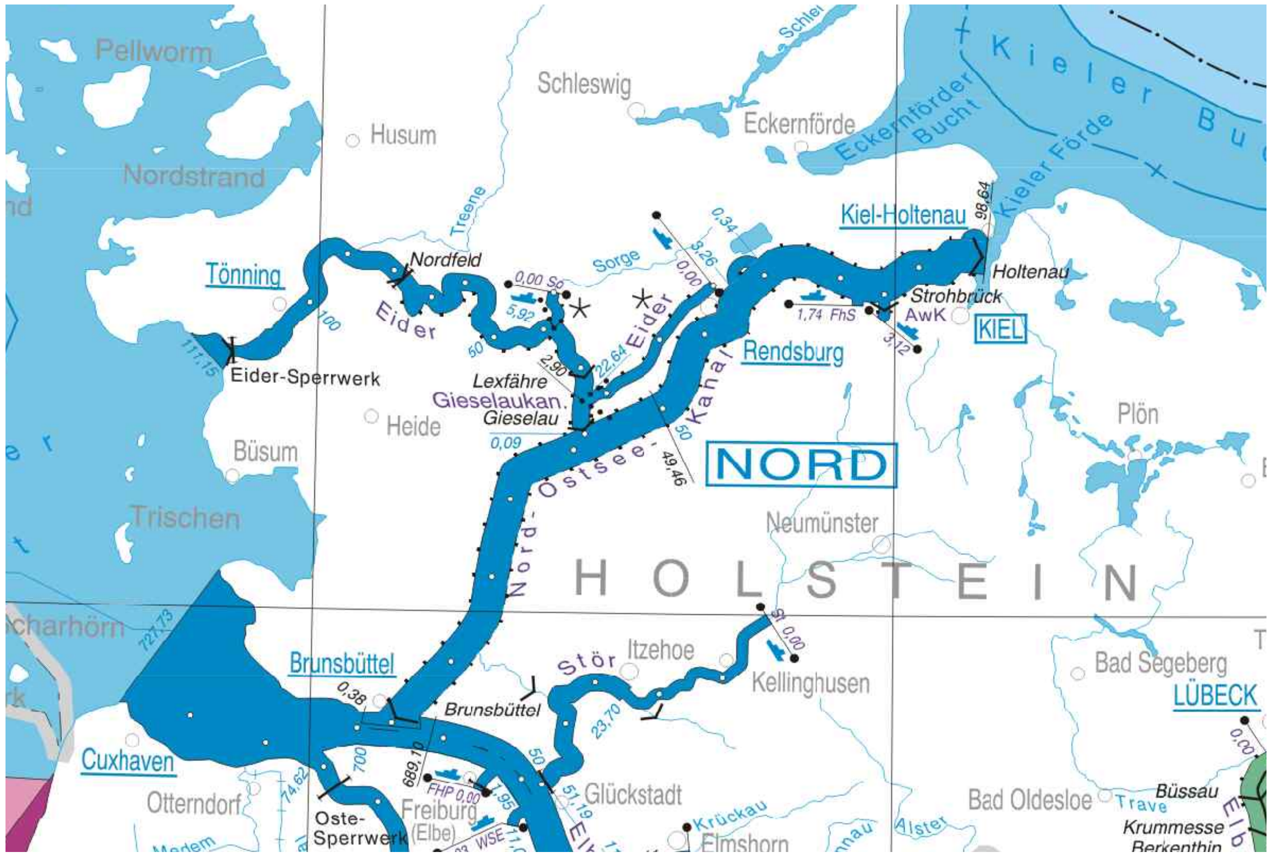
- Complexity results justify sequential routing algorithm
  - polynomial in theory and fast in practice
- Sequential gap is small for harbor grid layout
  - less than 4% for 4-6 horizontal tracks
- Static approach can be improved by load balancing and proper deadlock avoidance
  - load balancing improves runtime
  - runtime for deadlock avoidance increases rapidly with traffic density
- Dynamic router is is the clear winner for dense traffic
  - but (slightly) inferior in low traffic scenarios

Ship Traffic Optimization for the Kiel Canal



Elisabeth Günther
Marco Lübbecke, Rolf Möhring

The Kiel Canal (Nord-Ostsee-Kanal)



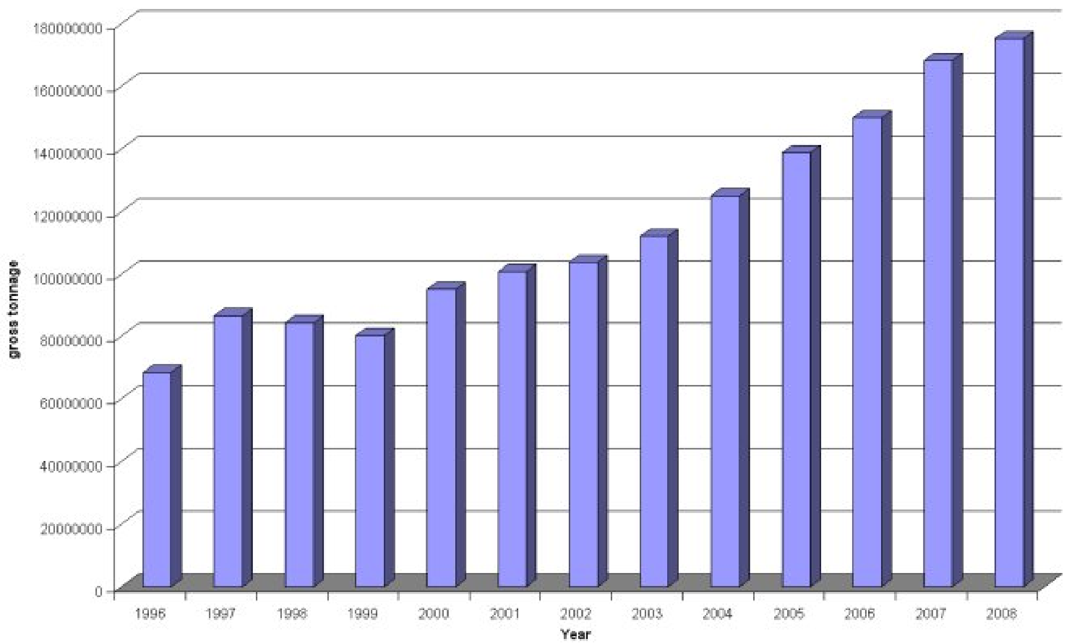
- Connects North Sea and Baltic Sea
- 280 nautical miles saved compared to the way around Skaw
- Canal with highest traffic in the World

Some traffic details

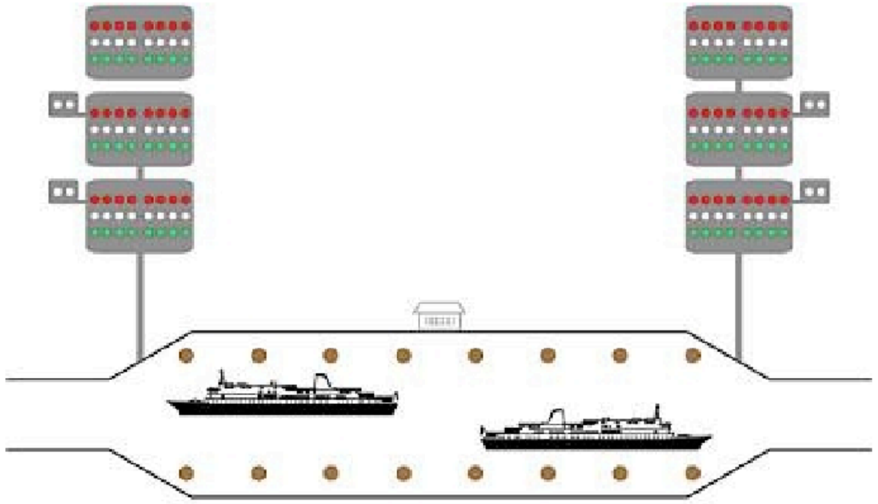
- Passage lasts 8–10 hours
- 40 vessels at the same time

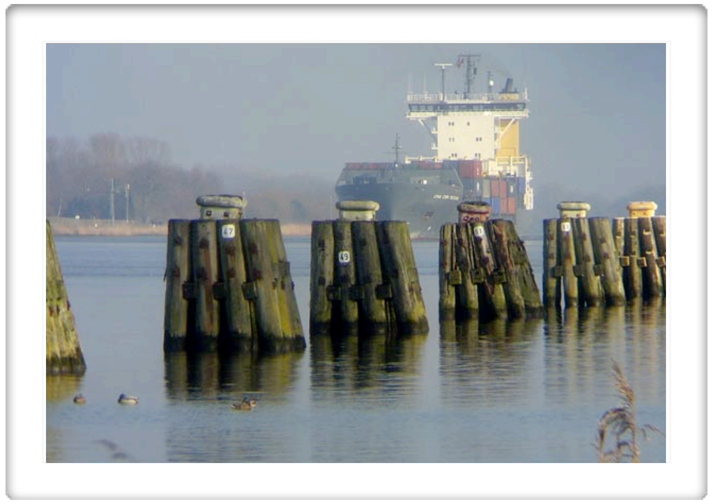
- It’s too tight in the canal
- regulations are needed

- Increasing gross tonnage
- 1996 – 2008



Opposing traffic creates problems



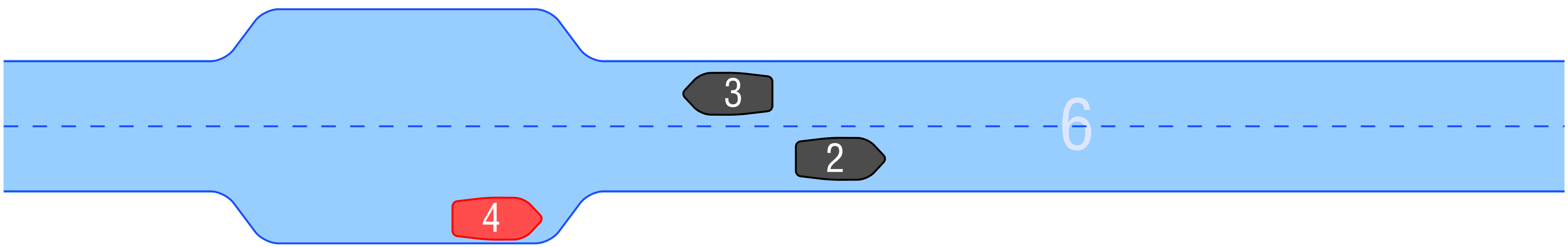


- Ships must be scheduled to wait in sidings
- Waiting can‘t be too long
- New ships arrive online

Combines routing and scheduling

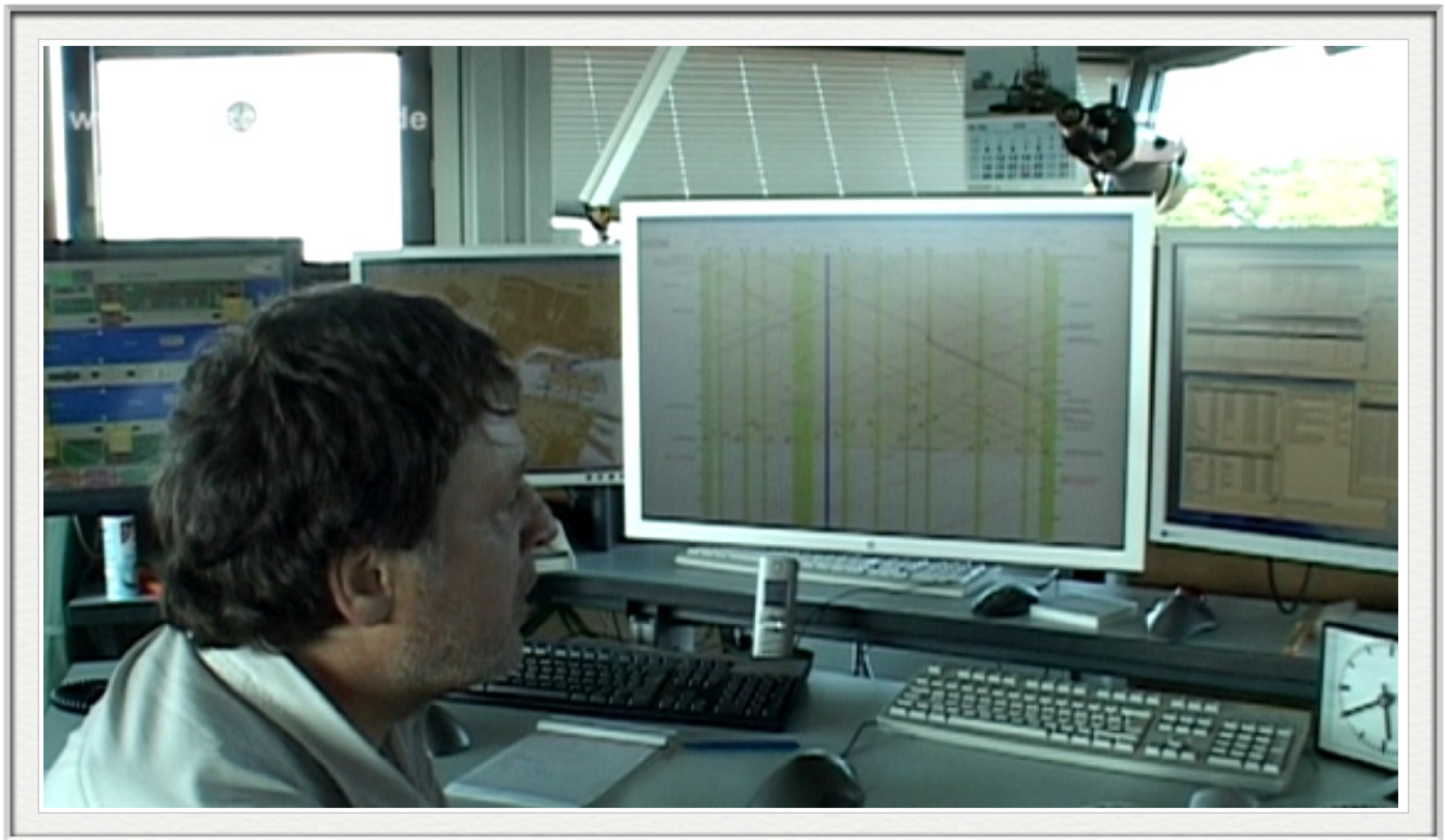
Optimization model

- The canal
  - segments and sidings
  - passing limits per segment
  - capacities per siding
- The ships
  - dimensions
  - origin, destination
  - release date, velocity
  - traffic group



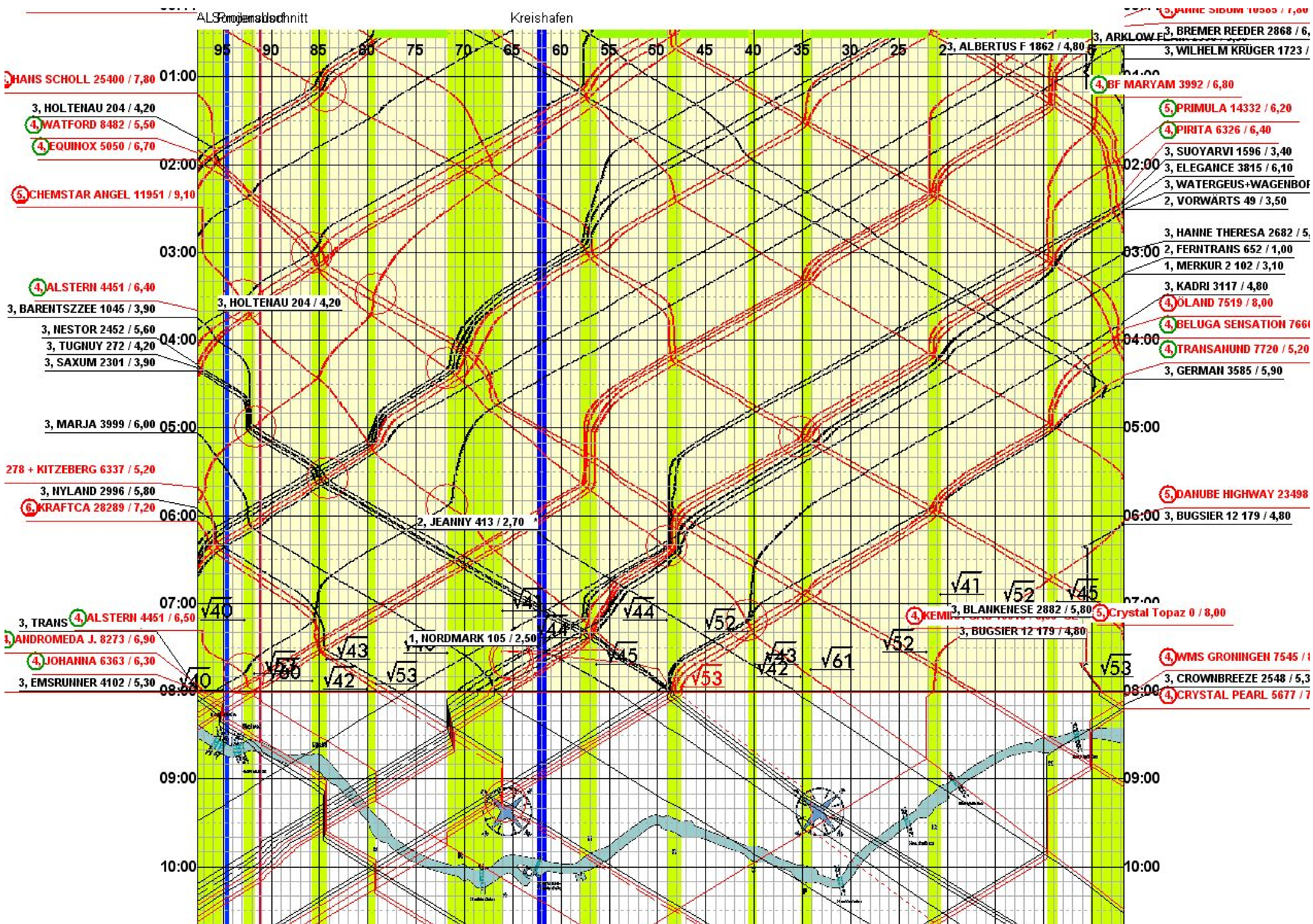
Goal: Find conflict-free dynamic routes minimizing total waiting time $\sum_{\text{ship } s} w_s$

Current practice



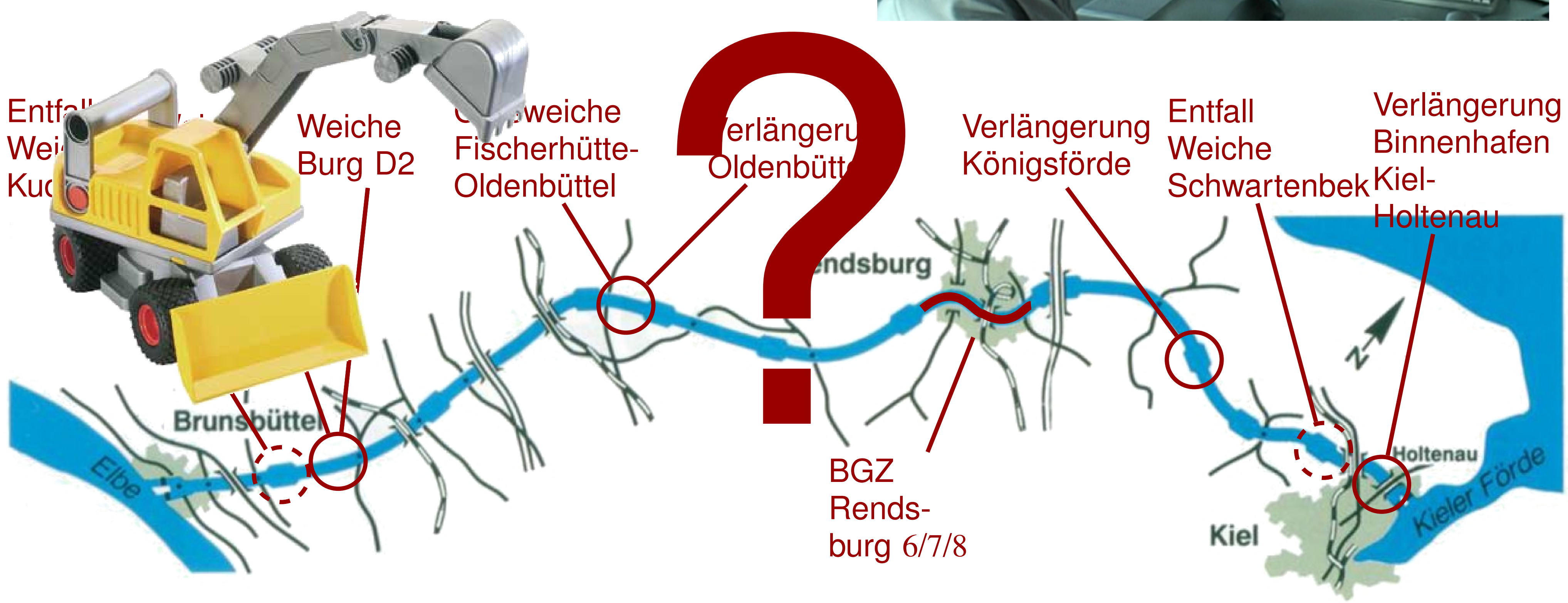
Manual traffic guidance by experienced planners

Space-time diagram



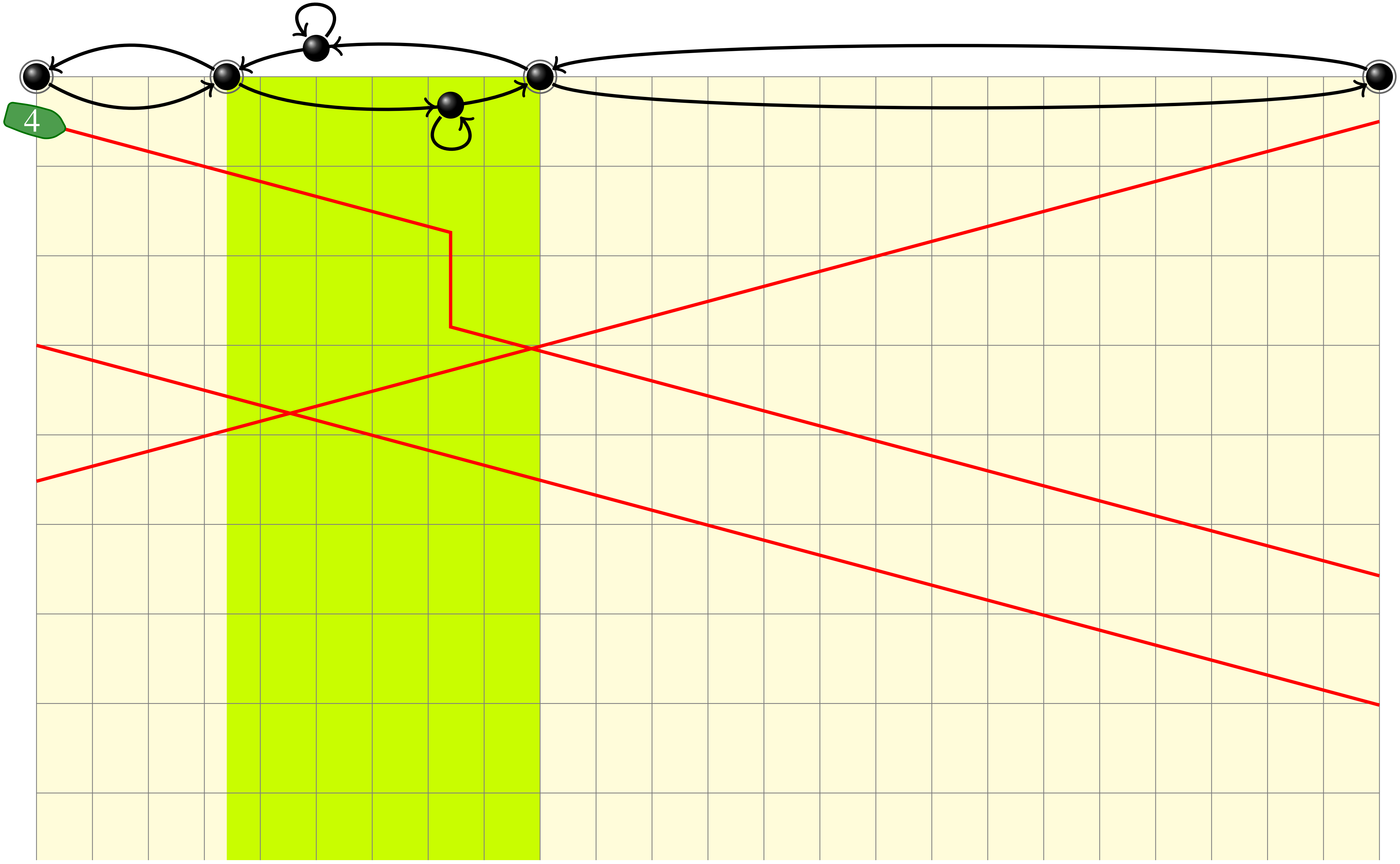
Goals of the project

- Improve manual planning



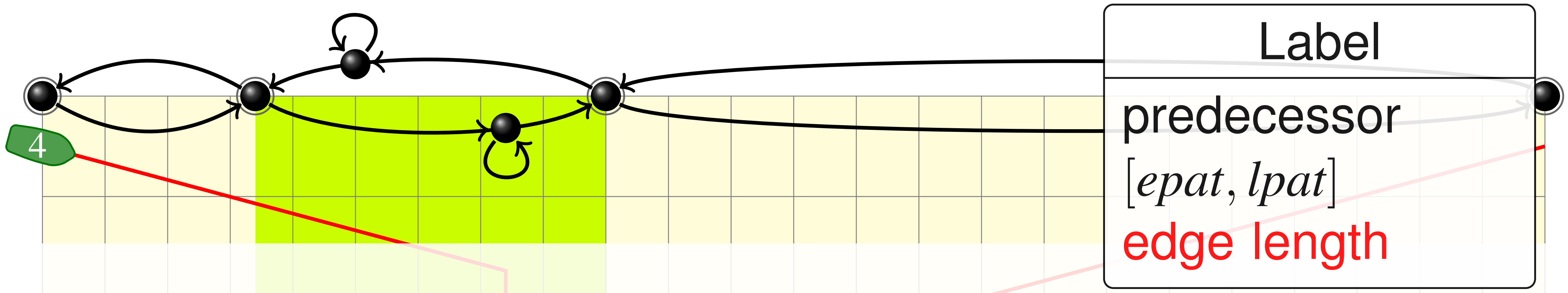
- Recommendations for canal enlargement (widening, new sidings ...)
- Automated guidance during construction phase

The routing part is polynomial...



Use the AGV routing algorithm

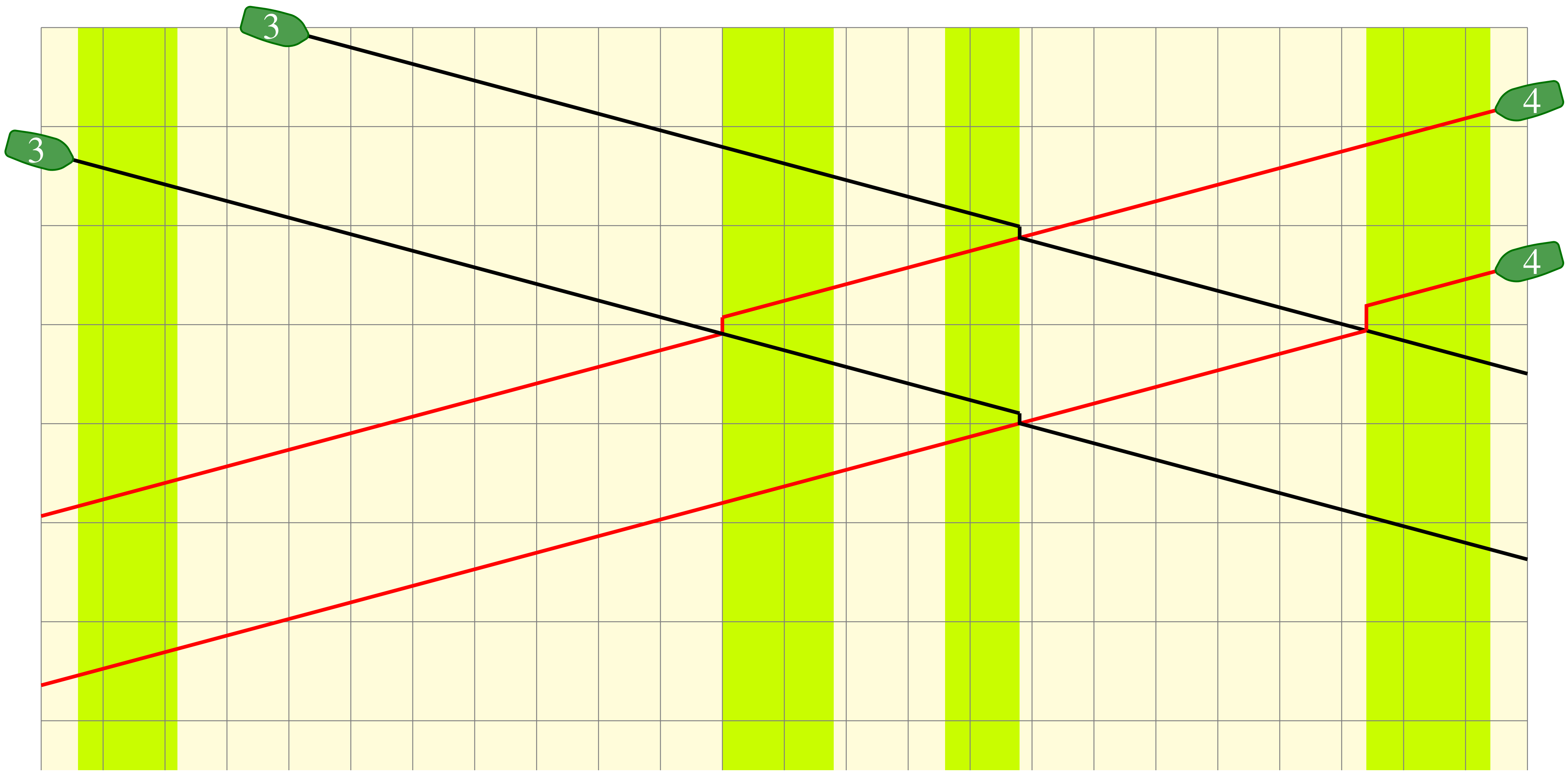
...but requires many more details



- advantage: algorithm takes all important sidings at once into account
- problem: different ship properties, location in sidings
- solution: special graph with "implicit discretization" and advanced blocking calculation
- covers all real world details

Sequential routing

- Can be arbitrarily bad
- There is no ordering that leads to an optimal solution



Mixed integer optimization models are too weak

even for simplifications of the model

$$\min \sum_{s\in S, t\in \mathcal{T}} w_{s,t}$$

s.t.

routing constraints

$$d_{s,e_{-s}} + \tau_{s,e} = d_{s,e} \qquad s \in S, e \in \mathcal{E} \setminus \mathcal{T}$$

$$d_{s,t_{-s}} + \tau_{s,t} + w_{s,t} = d_{s,t} \qquad s \in S, t \in \mathcal{T}$$

scheduling constraints

$$z_{s_1,s_2,e} = 1 \;\Rightarrow\; d_{s_1,e} + \Delta(s_1,s_2,e) \le d_{s_2,e} \qquad e \in \mathcal{E} \setminus \mathcal{T}, (s_1,s_2) \in C_e$$

$$z_{s_1,s_2,e} = 0 \;\Rightarrow\; d_{s_2,e} + \Delta(s_2,s_1,e) \le d_{s_1,e} \qquad e \in \mathcal{E} \setminus \mathcal{T}, (s_1,s_2) \in C_e$$

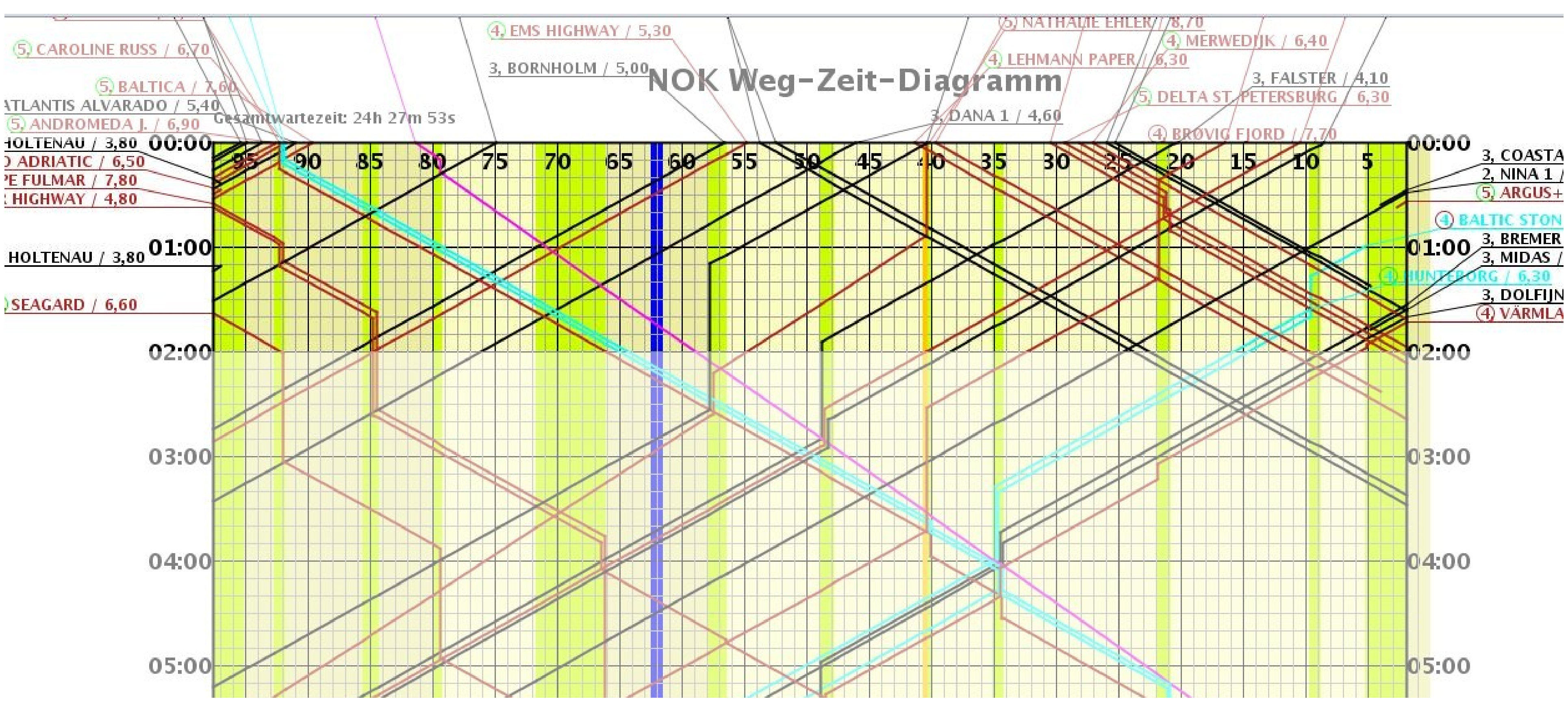
$$\underline{d}_{s,e} \le d_{s,e} \le \overline{d}_{s,e} \qquad s \in S, e \in \mathcal{E}$$

$$w_{s,t} \ge 0 \qquad s \in S, t \in \mathcal{T}$$

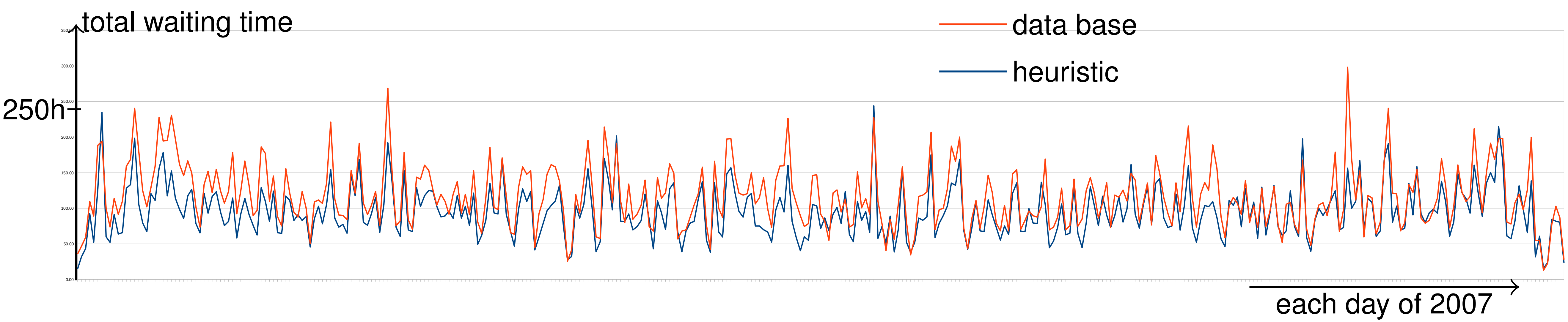
$$z_{s_1,s_2,e} \in \{0,1\} \qquad e \in \mathcal{E} \setminus \mathcal{T}, (s_1,s_2) \in C_e$$

Our algorithm is heuristic

- Uses a rolling time horizon and compute only partial routes
  - but ensures feasible extensions to a complete route can be done by the fast routing algorithm
- improves scheduling decisions by local search
- includes lock scheduling at both ends of the canal

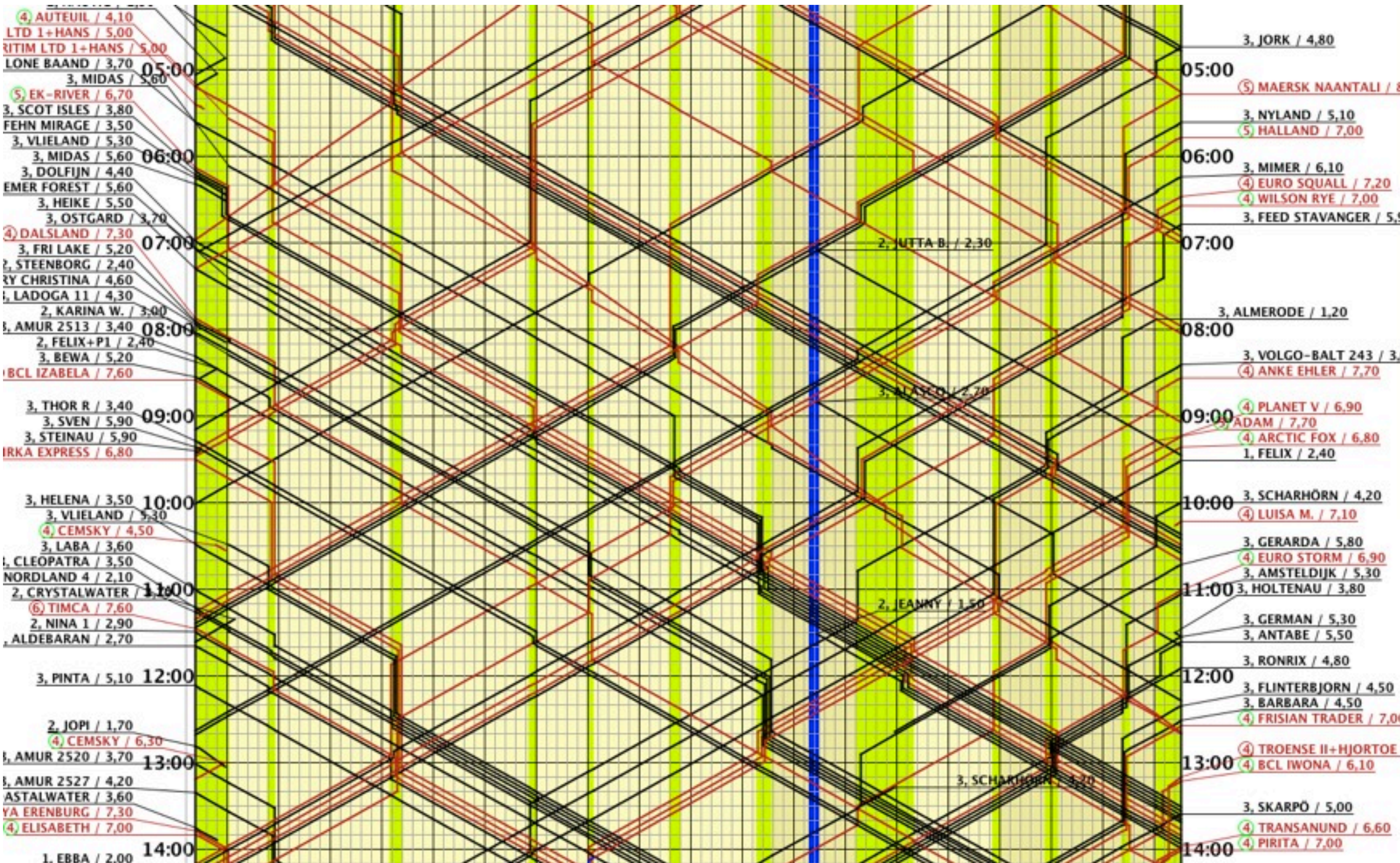


Results

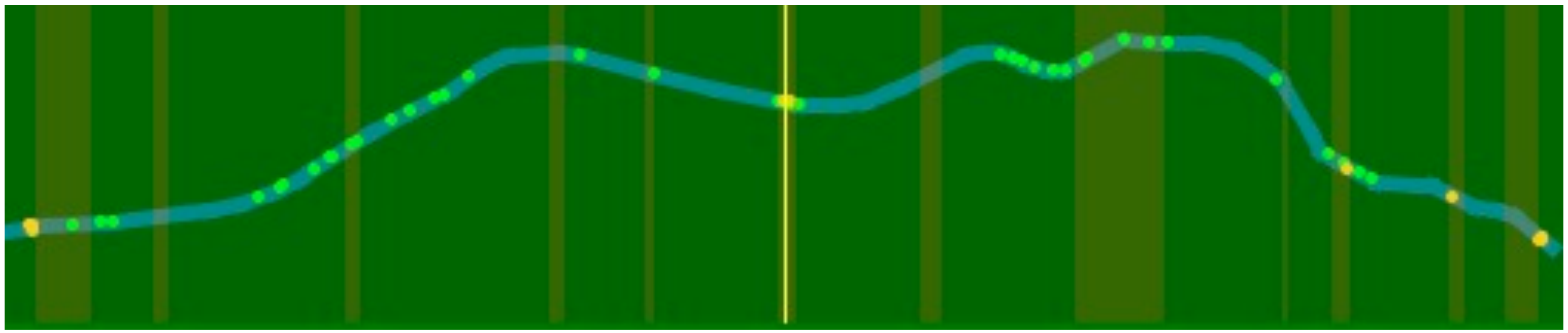


- Similar behavior as manual planning
- 15% improvement on average
- thus suited for studying different options for the canal enlargement
- recommendations enlargement made in 2011
- continuation of the project for the construction phase is planned

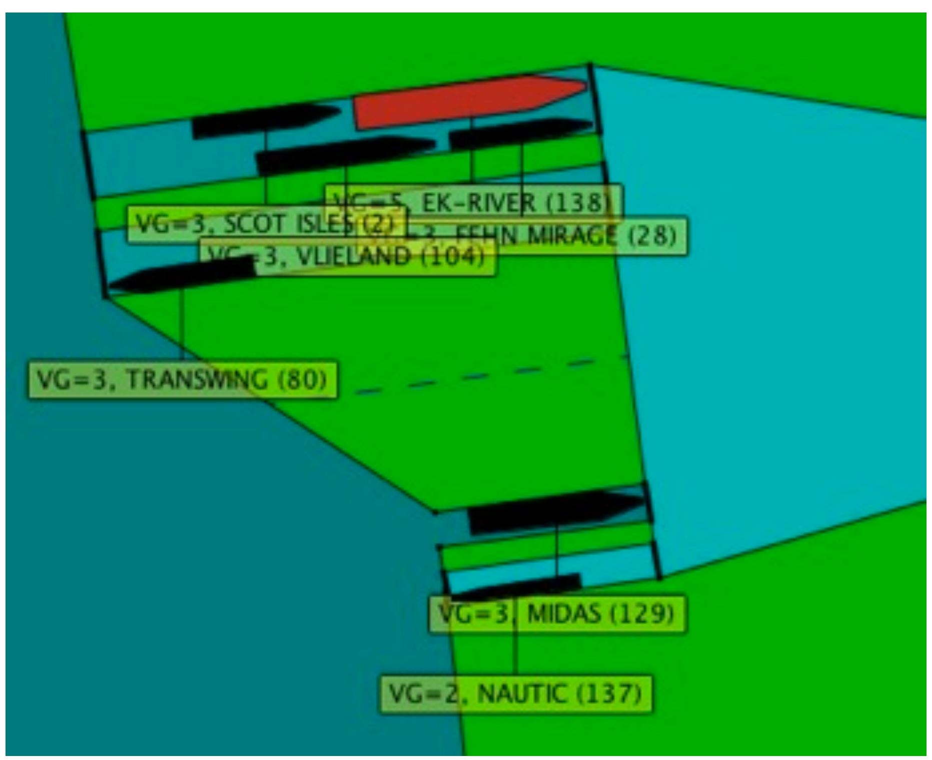
Glimpses of the algorithm: Space-time diagram



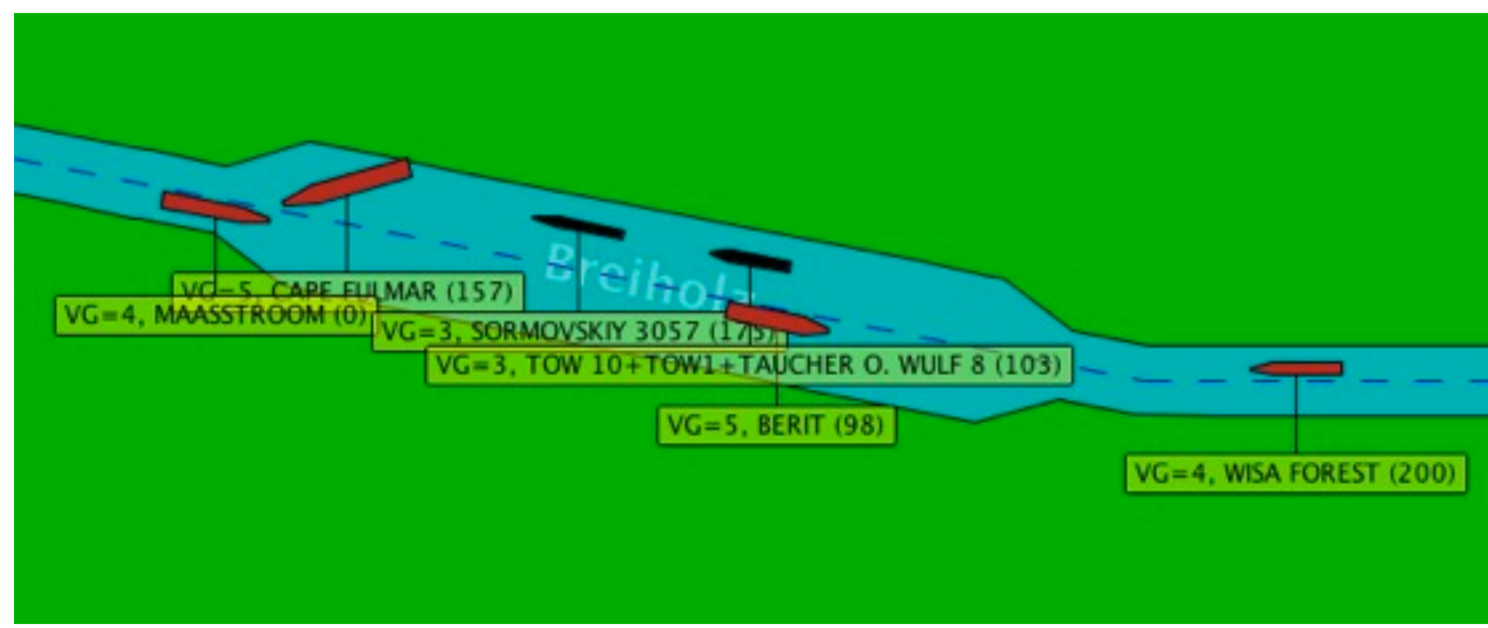
Glimpses of the algorithm: Traffic visualization



Global view



Locks in Brunsbüttel



Siding in Breiholz

COGA

Combinatorial Optimization & Graph Algorithms

Home Sitemap Glossary Search

Menu
- People
- Teaching
- Research
- Publications
- Projects
- Cooperations
- Jobs

News

05-09. Sep 2005
EuroComb Berlin

04-15. Oct 2005
Block course
CO@Work

Intranet

Login

Welcome at the Combinatorial Optimization & Graph Algorithms Group

Thanks!

Info:

www.coga.tu-berlin.de

Fakultät II - Mathematik und Naturwissenschaften
Institut für Mathematik, Sekr. MA 6-1
Technische Universität Berlin
Straße des 17. Juni 136
10623 Berlin
Germany

phone: +49 +30 314-25728
fax: +49 +30 314-25191

Campus Map (building MA)
How to find us