Green Lights for Everybody
Optimizing Traffic Signal Coordination in Networks

Ekkehard Köhler
Brandenburgische Technische Universität Cottbus

July 2009
Partners

- Rolf Möhring
  Technische Universität Berlin
- Gregor Wünsch
  Technische Universität Berlin
- Klaus Nökel
  ptv AG Karlsruhe
- Martin Strehler
  BTU Cottbus
- ADVEST Cluster
  bmbf
Problem: Too many red lights!

What to do?
Problem: Too many red lights!

What to do?

- change length of signal phases
- change red/green split
- change coordination, i.e. offsets between signals ("Grüne Welle"—progressive signal system)
Problem: Too many red lights!

What to do?

- change length of signal phases
- change red/green split
- change coordination, i.e. offsets between signals ("Grüne Welle"—progressive signal system)

Problem: Not all vehicles can get green corridor
(already single non-trivial)
Problem: Too many red lights!

What to do?
- change length of signal phases
- change red/green split
- change coordination, i.e. offsets between signals ("Grüne Welle"—progressive signal system)

Problem: Not all vehicles can get green corridor (already single non-trivial)

Question: Does exist generalization of green corridors in networks?
Problem: Too many red lights!

What to do?

- change length of signal phases
- change red/green split
- change coordination, i.e. offsets between signals ("Grüne Welle"—progressive signal system)

Problem: Not all vehicles can get green corridor (already single non-trivial)

Question: Does exist generalization of green corridors in networks?

Idea: Minimize waiting times (overall traffic time)
Problem: Too many red lights!

What to do?
- change length of signal phases
- change red/green split
- change coordination, i.e. offsets between signals ("Grüne Welle"—progressive signal system)

Problem: Not all vehicles can get green corridor (already single non-trivial)

Question: Does exist generalization of green corridors in networks?

Idea: Minimize waiting times (overall traffic time)

Method: Discrete optimization; mathematical programming
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Not optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Easiest Case: Optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized
Two-way-traffic: Not optimized

What to do?
Two-way-traffic: Not optimized

- What to do?
- Optimization by hand...
Optimization „by hand“
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized
Two-way-traffic: Optimized

► What to do in traffic networks?
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks
Signal coordination in networks

Green corridor for all — impossible! What to do?
Signal coordination in networks

Green corridor for all — impossible! What to do?
Minimize overall traffic time
Model: Traffic Signal

**fixed time control:** not traffic sensitive

- Reason 1: many cities have such signal systems
- Reason 2: rush-hour → traffic sensitive = fixed time control
Model: Traffic Signal

fixed time control: not traffic sensitive

- Reason 1: many cities have such signal systems
- Reason 2: rush-hour → traffic sensitive = fixed time control

Concepts:

- signal plan
- cycle time $T$
Model: Traffic Signal

**fixed time control:** not traffic sensitive

- Reason 1: many cities have such signal systems
- Reason 2: rush-hour → traffic sensitive = fixed time control

**Concepts:**

- signal plan
- cycle time $T$
- signal (LSA 1)
- signal group (A, B, C)
Model: Traffic Signal

**fixed time control:** *not traffic sensitive*

- Reason 1: many cities have such signal systems
- Reason 2: rush-hour → traffic sensitive = fixed time control

**Concepts:**

- signal plan
- cycle time $T$
- signal (LSA 1)
- signal group (A, B, C)
- intranode-offset $\Psi$
Model: Traffic Signal

**fixed time control:** not traffic sensitive

- Reason 1: many cities have such signal systems
- Reason 2: rush-hour $\rightarrow$ traffic sensitive $=$ fixed time control

**Concepts:**

- signal plan
- cycle time $T$
- signal (LSA 1)
- signal group (A, B, C)
- intranode-offset $\Psi$
- (arc-)offset $\phi$
Model: Traffic Signal

**fixed time control**: not traffic sensitive

- Reason 1: many cities have such signal systems
- Reason 2: rush-hour → traffic sensitive = fixed time control

**Concepts:**

- signal plan
- cycle time $T$
- signal (LSA 1)
- signal group (A, B, C)
- intranode-offset $\Psi$
- (arc-)offset $\phi$

**Aim:** Find set of offsets, such that overall delay is minimal
Model: Network

- street network $G = (V, A)$ (multi-graph)
Model: Network

- street network $G = (V, A)$ (multi-graph)
- vertices: signalized intersections
Model: Network

- street network $G = (V, A)$ (multi-graph)
- vertices: signalized intersections
- arcs: connecting streets between intersections (multi-arcs for different signal groups)
Model: Network

- street network $G = (V, A)$ (multi-graph)
- vertices: signalized intersections
- arcs: connecting streets between intersections (multi-arcs for different signal groups)
- cycle length $T$: length of signal phase (e.g. 60, 80, 120 sec.)
Model: Network

- street network $G = (V, A)$ (multi-graph)
- vertices: signalized intersections
- arcs: connecting streets between intersections (multi-arcs for different signal groups)
- cycle length $T$: length of signal phase (e.g. 60, 80, 120 sec.)
- transit time $\tau$ on arcs (constant)
Model: Network

- street network $G = (V, A)$ (multi-graph)
- **vertices**: signalized intersections
- **arcs**: connecting streets between intersections (multi-arcs for different signal groups)
- cycle length $T$: length of signal phase (e.g. 60, 80, 120 sec.)
- transit time $\tau$ on arcs (constant)
- given traffic volumes $f$ on arcs (from (static) traffic assignment for given OD-matrix)
Model: Network

- street network $G = (V, A)$ (multi-graph)
- vertices: signalized intersections
- arcs: connecting streets between intersections (multi-arcs for different signal groups)
- cycle length $T$: length of signal phase (e.g. 60, 80, 120 sec.)
- transit time $\tau$ on arcs (constant)
- given traffic volumes $f$ on arcs (from (static) traffic assignment for given OD-matrix)
- macroscopic model – vehicles move in platoons
Model: Network

- street network \( G = (V, A) \) (multi-graph)
- **vertices**: signalized intersections
- **arcs**: connecting streets between intersections
  (multi-arcs for different signal groups)
- cycle length \( T \): length of signal phase (e.g. 60, 80, 120 sec.)
- transit time \( \tau \) on arcs (constant)
- given traffic volumes \( f \) on arcs
  (from (static) traffic assignment for given OD-matrix)
- **macroscopic** model – vehicles move in platoons
- platoon length \( p \)
Model: Network

- street network \( G = (V, A) \) (multi-graph)
- **vertices**: signalized intersections
- **arcs**: connecting streets between intersections (multi-arcs for different signal groups)
- cycle length \( T \): length of signal phase (e.g. 60, 80, 120 sec.)
- transit time \( \tau \) on arcs (constant)
- given traffic volumes \( f \) on arcs (from (static) traffic assignment for given OD-matrix)
- **macroscopic** model – vehicles move in platoons
- platoon length \( p \)
- arrival time of platoon relative to beginning of green phase \( \gamma \)
Model: Platoons
Model: Platoons

Platoon of vehicles
Model: Platoons

\[ T \]

\[ R \quad G \]

platoon of vehicles

\[ p \]

\[ \tau \]
Model: Platoons

![Diagram of a platoon of vehicles]

- Offset
- T
- R
- G
- γ
- p
- τ

**platoon of vehicles**
Model: Platoons

 Offset

 T

 R  G  γ

 p

 τ

 platoon of vehicles
Model: Waiting Times

- $\gamma$
- $G - R$
- number of vehicles
- time
- cycle time $T$

Ekkehard Kühler
BTU Cottbus
Linearized sum of waiting times
MIP-Optimization: Constraints

arc \((v_i, v_j)\):

- offset \(\phi_{ij}\): fractional variable
- arrival time of platoon at vertex \(v_j\): \(\gamma_{ij}\) \((\gamma_{ij} \in [0, T])\)
- determine \(\gamma_{ij}\) arrival time of platoon, \(\tau_{ij} - \gamma_{ij} + r_{ij} = \phi_{ij}\) 
  \(\tau_{ij}\) transit time on arc, \(r_{ij}\) length of red phase; \(\phi_{ij}\) offset per arc
- admissibility condition:
  only valid if offsets on cycles sum up to multiple of cycle time
  \(\rightarrow\) cycle base of graphs
  (conversion from arc offsets to vertex offsets)

K., Liebchen, Rizzi, Wünsch. *Networks* 2008
MIP Formulation

\[
\begin{align*}
\min & \quad \sum_{(i,j) \in A} f_{ij} z_{ij} \\
\sum_{e \in F(\ell)} \phi_e - \sum_{e \in R(\ell)} \phi_e + \sum_{r=1}^{k_\ell} \psi_{v_{j},p} &= n_\ell \cdot T \quad \forall \ell \in C \\
\sum_{e \in F(\ell)} \phi_e - \sum_{e \in R(\ell)} \phi_e + \sum_{r=1}^{k_\ell} \psi_{v_{j},p} &= n_\ell \cdot T \quad \forall \ell \in C \\
z_{ij} &\geq g_e^e(\gamma_{ij}) \quad \forall e = (i,j) \in A \\
\tau_{ij} - \gamma_{ij} + r_{ij} &= \phi_{ij} \quad \forall (i,j) \in A \\
n_\ell \leq n_\ell \leq \bar{n}_\ell \\
n_\ell &\in \mathbb{Z} \\
\gamma_{ij} &\in [0, T] \quad \forall (i,j) \in A
\end{align*}
\]
MIP-Optimisation: Remarks

- basic model: Gartner, Little, Gabbay. 1975.
MIP-Optimisation: Remarks

- basic model: Gartner, Little, Gabbay. 1975.
MIP-Optimisation: Remarks

- basic model: Gartner, Little, Gabbay. 1975.
- related: PESP (Periodic Event Scheduling Problem) periodic time labeling
MIP-Optimisation: Remarks

- basic model: Gartner, Little, Gabbay. 1975.
- related: PESP (Periodic Event Scheduling Problem)
  periodic time labeling
- different red/green splits possible (switch between modes)
MIP-Optimisation: Remarks

- basic model: Gartner, Little, Gabbay. 1975.
- related: PESP (Periodic Event Scheduling Problem) periodic time labeling
- different red/green splits possible (switch between modes)
- non-uniform cycle length in network possible
MIP-Optimisation: Results

CPLEX; cancel after 10 minutes

- 140 vertices, 500 arcs, 400 cycles
  optimality gap 19%

- 70 vertices, 180 arcs, 110 cycles
  optimality gap 0.04% (1% after 10 sec.)
MIP-Optimisation: Results

CPLEX; cancel after 10 minutes

- 140 vertices, 500 arcs, 400 cycles
  optimality gap 19%

- 70 vertices, 180 arcs, 110 cycles
  optimality gap 0.04% (1% after 10 sec.)

- comparison with TRANSYT via simulation
  (microscopic simulation with VISSIM)
MIP-Optimisation: Results

CPLEX; cancel after 10 minutes

- 140 vertices, 500 arcs, 400 cycles
  optimality gap 19%

- 70 vertices, 180 arcs, 110 cycles
  optimality gap 0.04% (1% after 10 sec.)

- comparison with TRANSYT via simulation
  (microscopic simulation with VISSIM)

- comparable results in reasonable time
MIP-Optimisation: Results

CPLEX; cancel after 10 minutes

- 140 vertices, 500 arcs, 400 cycles
  optimality gap 19%

- 70 vertices, 180 arcs, 110 cycles
  optimality gap 0.04% (1% after 10 sec.)

- comparison with TRANSYT via simulation
  (microscopic simulation with VISSIM)

- comparable results in reasonable time

- proof of quality by lower bound
  (optimality gap)
But:

**Problem:** coordination changes transit times \(\Rightarrow\) traffic changes!
But:

**Problem:** coordination changes transit times $\Rightarrow$ traffic changes!

**New questions:** can coordination and traffic assignment be determined simultaneously?
But:

**Problem:** coordination changes transit times $\Rightarrow$ traffic changes!

**New questions:** can coordination and traffic assignment be determined simultaneously?

$\rightarrow$ Integrated model for traffic assignment and signal coordination

bmb+f Project ADVEST in cooperation with ptv AG
Integrated Model: Coordination — Traffic Assignment

**Question:** How does coordination change traffic assignment?
Integrated Model: Coordination — Traffic Assignment

**Question:** How does coordination change traffic assignment?

**Problem:**
underlying traffic assignment model is static (without temporal components)
⇒ offsets at signals cannot directly be mapped
Integrated Model: Coordination — Traffic Assignment

**Question:** How does coordination change traffic assignment?

**Problem:**
underlying traffic assignment model is static (without temporal components)
⇒ offsets at signals cannot directly be mapped

**Link Performance Function:** average transit time; depends on traffic
(constant) transit time + (traffic dependent) waiting time at intersection
Integrated Model: Coordination — Traffic Assignment

**Question:** How does coordination change traffic assignment?

**Problem:**
underlying traffic assignment model is static (without temporal components)
⇒ offsets at signals cannot directly be mapped

**Link Performance Function:** average transit time; depends on traffic
(constant) transit time + (traffic dependent) waiting time at intersection

**static traffic assignment:**
- convex
- monotonically increasing

**Signal coordination:**

\[
\begin{align*}
\tau_e &= 0 \\
\tau_e' &= x_e
\end{align*}
\]
Integrated Model: Coordination — Traffic Assignment

**Question:** How does coordination change traffic assignment?

**Problem:**
underlying traffic assignment model is static (without temporal components)
⇒ offsets at signals cannot directly be mapped

**Link Performance Function:** average transit time; depends on traffic
(constant) transit time + (traffic dependent) waiting time at intersection

**static traffic assignment:**
- convex
- monotonically increasing

**Signal coordination:**
- transit time constant
- offset changes waiting time
Integrated Model: Coordination — Traffic Assignment

**Question:** How does coordination change traffic assignment?

**Problem:**
underlying traffic assignment model is static (without temporal components)
⇒ offsets at signals cannot directly be mapped

**Link Performance Function:** average transit time; depends on traffic
(constant) transit time + (traffic dependent) waiting time at intersection

*static traffic assignment:*

- convex
- monotonically increasing

*Signal coordination:*

- transit time constant
- offset changes waiting time
1. Approach: Static Model

1. **Stage:** Assume offsets to be given
1. Approach: Static Model

1. **Stage:** Assume offsets to be given

**Question:** Can waiting times be mapped in link perf. function?
1. Approach: Static Model

1. **Stage**: Assume offsets to be given

**Question**: Can waiting times be mapped in link perf. function?
1. Approach: Static Model

1. **Stage:** Assume offsets to be given

**Question:** Can waiting times be mapped in link perf. function?
1. **Approach: Static Model**

1. **Stage:** Assume offsets to be given

**Question:** Can waiting times be mapped in link perf. function?

![Link Performance Function Diagram]

- non-monotonic
- non convex
- non concave
1. Approach: Static Model

1. Stage: Assume offsets to be given

Question: Can waiting times be mapped in link perf. function?

link performance function:
- non-monotonic
- non convex
- non concave

Open: Can traffic assignment be computed?
2. Approach: Time-Expanded Model

Idea: Model temporal character of offsets

Model: time-expanded network (dynamic flows, Ford/Fulkerson)
2. Approach: Time-Expanded Model

**Idea:** Model temporal character of offsets

**Model:** time-expanded network (dynamic flows, Ford/Fulkerson)

- flow travels through the network over time
- map time-dependent properties of arcs
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.
Time-Expanded Graph

**Idea (Ford & Fulkerson):**
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

**Observation:**
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

Observation:
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

**Idea (Ford & Fulkerson):**
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

**Observation:**
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

Observation:
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

Observation:
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

**Idea (Ford & Fulkerson):**
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

**Observation:**
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

**Idea (Ford & Fulkerson):**
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

**Observation:**
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
**Time-Expanded Graph**

**Idea (Ford & Fulkerson):**
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

**Observation:**
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

Observation:
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

Observation:
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

Observation:
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$. 
Time-Expanded Graph

Idea (Ford & Fulkerson):
Copy of each vertex per time step; arcs connect vertex copies corresponding to transit time.

Observation:
Dynamic flow in $G$ corresponds to static flow in time-expanded graph $G^T$.

$\Rightarrow$ standard algorithms for static flows applicable
2. Approach: Expanded Model

**Idea:** Model time-dependence of offsets

**Model:** time-expanded network (dynamic flows, Ford/Fulkerson)
2. Approach: Expanded Model

Idea: Model time-dependence of offsets

Model: time-expanded network (dynamic flows, Ford/Fulkerson)

New: cyclic expansion (combination of static and dynamic)
2. Approach: Expanded Model

Idea: Model time-dependence of offsets

Model: time-expanded network (dynamic flows, Ford/Fulkerson)

New: cyclic expansion (combination of static and dynamic)
2. Approach: Expanded Model

**Idea:** Model time-dependence of offsets

**Model:** time-expanded network (dynamic flows, Ford/Fulkerson)

**New:** cyclic expansion (combination of static and dynamic)
2. Approach: Expanded Model

**Idea:** Model time-dependence of offsets

**Model:** time-expanded network (dynamic flows, Ford/Fulkerson)

**New:** cyclic expansion (combination of static and dynamic)
2. Approach: Expanded Model

**objective function:** Minimize overall travel time

→ **Min Cost Circulation** (costs: const. transit-/waiting-times)
2. Approach: Expanded Model

**objective function:** Minimize overall travel time
→ **MIN COST CIRCULATION** (costs: const. transit-/waiting-times)
2. Approach: Expanded Model

**objective function:** Minimize overall travel time

→ **Min Cost Circulation** (costs: const. transit-/waiting-times)
2. Approach: Expanded Model

**objective function:** Minimize overall travel time

→ **Min Cost Circulation** *(costs: const. transit-/waiting-times)*
2. Approach: Expanded Model

**objective function**: Minimize overall travel time

→ **Min Cost Circulation** (costs: const. transit-/waiting-times)

And:

- time-expansion
- green-arcs
- waiting-arcs
- suitable capacities
- cost 0 on auxiliary arcs
New Model: Summary

- cyclic time expansion
- constant transit time on streets plus waiting times at intersection
- can change level of discretization
- traffic assignment reduces to min cost circulation
New Model: Summary

- cyclic time expansion
- constant transit time on streets plus waiting times at intersection
- can change level of discretization
- traffic assignment reduces to min cost circulation

Challenges:

- size of the model
- non-uniform cycle times
- calibration
- test with simulation tool
Computational Results: Inner city of Braunschweig

- 7 crossings
- 3 commodities
- cycle time of 84 seconds

data: group of Sándor Fekete
### Experiments:

Expanded networks with different number of expansion steps

<table>
<thead>
<tr>
<th>Steps</th>
<th>Nodes</th>
<th>Edges</th>
<th>Variables</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>266</td>
<td>238</td>
<td>1484</td>
<td>1168</td>
</tr>
<tr>
<td>28</td>
<td>532</td>
<td>476</td>
<td>2968</td>
<td>2332</td>
</tr>
<tr>
<td>84</td>
<td>1596</td>
<td>1428</td>
<td>8904</td>
<td>6980</td>
</tr>
</tbody>
</table>

Running times for different numbers of expansion steps

<table>
<thead>
<tr>
<th>Steps</th>
<th>CPLEX Server</th>
<th>MT CPLEX S.</th>
<th>CPLEX PC</th>
<th>SCIP PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>37 s</td>
<td>37 s</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>28</td>
<td>160 s</td>
<td>172 s</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>84</td>
<td>2700 s</td>
<td>3000 s</td>
<td>8000 s</td>
<td>21 hours</td>
</tr>
</tbody>
</table>
Refining model: Hierarchic approach

- solve rough model with few expansion steps (only allow a few offsets, i.e. multiples of $x$)
- refine network by more expansion steps
- solve refined model with additional constraints

<table>
<thead>
<tr>
<th>Steps</th>
<th>original</th>
<th>refined</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>37 s</td>
<td>37 s</td>
</tr>
<tr>
<td>28</td>
<td>160 s</td>
<td>2 s</td>
</tr>
<tr>
<td>84</td>
<td>3000 s</td>
<td>8 s</td>
</tr>
</tbody>
</table>

- about 75% of binary variables fixed
- Refined model yields same optimal solution for this example (speed-up factor 50)
Simulation with Vissim:

Comparison of

- (estimated) actual coordination
- various solutions obtained by CPLEX
- random coordinations
- heuristic approach
1st Commodity
2nd Commodity
3rd Commodity
Simulation with low load

Load: 1: 300 cars per hour, 2: 150 cph, 3: 300 cph

Mean mean travel time (10 runs) in seconds:

<table>
<thead>
<tr>
<th>Commodity</th>
<th>actual</th>
<th>CPLEX</th>
<th>random</th>
<th>heuristic</th>
<th>no TLs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>145</td>
<td>98</td>
<td>151</td>
<td>145</td>
<td>64</td>
</tr>
<tr>
<td>2</td>
<td>86</td>
<td>85</td>
<td>152</td>
<td>137</td>
<td>47</td>
</tr>
<tr>
<td>3</td>
<td>63</td>
<td>63</td>
<td>77</td>
<td>66</td>
<td>44</td>
</tr>
<tr>
<td>weighted</td>
<td>$\sum$</td>
<td>100</td>
<td>81</td>
<td>121</td>
<td></td>
</tr>
</tbody>
</table>

Standard deviation of mean travel time: 2 s
Simulation with dense traffic

Load: 1: 600 cars per hour, 2: 300 cph, 3: 600 cph

Mean travel time in seconds:

<table>
<thead>
<tr>
<th>Commodity</th>
<th>actual</th>
<th>CPLEX</th>
<th>random</th>
<th>heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>163</td>
<td>104</td>
<td>164</td>
<td>167</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>96</td>
<td>153</td>
<td>141</td>
</tr>
<tr>
<td>3</td>
<td>65</td>
<td>65</td>
<td>90</td>
<td>68</td>
</tr>
<tr>
<td>weighted</td>
<td>∑</td>
<td>109</td>
<td>87</td>
<td>132</td>
</tr>
</tbody>
</table>
Simulation with extreme load

Load: 1: max, 2: max, 3: max

Mean travel time in seconds / cars per hour:

<table>
<thead>
<tr>
<th>Commodity</th>
<th>actual</th>
<th>CPLEX</th>
<th>random</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>528 / 830</td>
<td>528 / 830</td>
<td>552 / 760</td>
</tr>
<tr>
<td>2</td>
<td>182 / 600</td>
<td>164 / 600</td>
<td>239 / 500</td>
</tr>
<tr>
<td>3</td>
<td>121 / 1400</td>
<td>118 / 1400</td>
<td>157 / 1400</td>
</tr>
</tbody>
</table>

weighted $\sum$ - - -
Summary

- showed approaches how to model signal coordination using discrete optimization methods
Summary

- showed approaches how to model signal coordination using discrete optimization methods
- used macroscopic models; evaluated with microscopic simulation
Summary

- showed approaches how to model signal coordination using discrete optimization methods
- used macroscopic models; evaluated with microscopic simulation
- purely static MIP optimization; applicable also for larger instances
Summary

- showed approaches how to model signal coordination using discrete optimization methods
- used macroscopic models; evaluated with microscopic simulation
- purely static MIP optimization; applicable also for larger instances
- semi-dynamic MIP optimization; work in progress
Summary

- showed approaches how to model signal coordination using discrete optimization methods
- used macroscopic models; evaluated with microscopic simulation
- purely static MIP optimization; applicable also for larger instances
- semi-dynamic MIP optimization; work in progress
- important property: know lower bounds (distance from optimal solution)