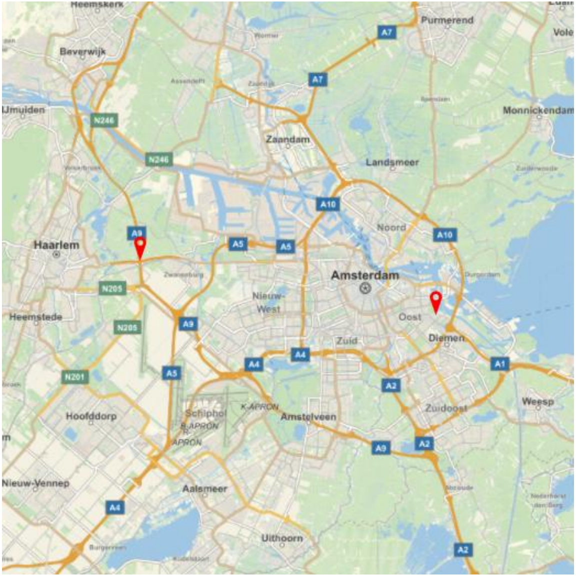
A Framework for Efficient Dynamic Routing under Stochastically Varying Conditions

Nikki Levering, University of Amsterdam

R. Núñez-Queija (UvA, CWI), M. Mandjes, M. Boon (TU/e), and R. Kamphuis (UvA)

May, 2023

Random Environment



Random Environment

Trip travel times in real traffic networks suffer from:

- Time-dependence: daily patterns, seasonality
- Stochasticity: random effects that lead to delays

Outline

1. Markovian velocity model
2. Find the optimal route
3. Determine the optimal departure time
4. Related and other work

Outline

1. **Markovian velocity model**
2. Find the optimal route
3. Determine the optimal departure time
4. Related and other work

Markovian Velocity Model

Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections)
- A the set of arcs (roads between ramps/intersections)

Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections)
- A the set of arcs (roads between ramps/intersections)

Markovian Velocity Model

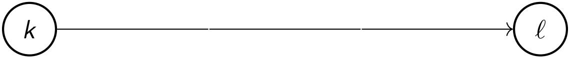
Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections)
- A the set of arcs (roads between ramps/intersections)

Markovian Velocity Model

Consider the graph $G = (N, A)$ with

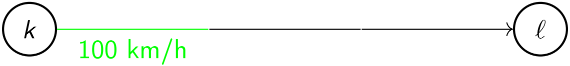
- N the set of nodes (ramps/intersections)
- A the set of arcs (roads between ramps/intersections)



Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections);
- A the set of arcs (roads between ramps/intersections)



Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections);
- A the set of arcs (roads between ramps/intersections)

Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections);
- A the set of arcs (roads between ramps/intersections)

Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections);
- A the set of arcs (roads between ramps/intersections)

Speeds between ramps vary between finitely many values.

Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections);
- A the set of arcs (roads between ramps/intersections)

Speeds between ramps vary between finitely many values. We deal with these changing speeds by working with a Markovian background process.

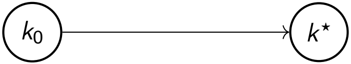
Markovian Velocity Model

Consider the graph $G = (N, A)$ with

- N the set of nodes (ramps/intersections);
- A the set of arcs (roads between ramps/intersections)

Speeds between ramps vary between finitely many values. We deal with these changing speeds by working with a Markovian background process. This way we can incorporate both time-dependence and random effects.

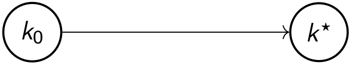
Markov modulation: random effects



Markov process with incidents:

$$X_{k_0 k^\star}(t) = \begin{cases} 0 & \text{if no incident on road } k_0 k^\star \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

Markov modulation: random effects

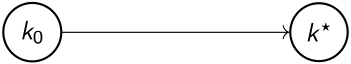


Markov process with incidents:

$$X_{k_0 k^\star}(t) = \begin{cases} 0 & \text{if no incident on road } k_0 k^\star \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

The speed on link $k_0 k^\star$ can be defined dependent on $X_{k_0 k^\star}(t)$

Markov modulation: random effects

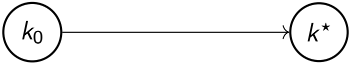


Markov process with incidents:

$$X_{k_0 k^\star}(t) = \begin{cases} 0 & \text{if no incident on road } k_0 k^\star \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

The speed on link $k_0 k^\star$ can be defined dependent on $X_{k_0 k^\star}(t)$,
e.g.
$v_{k_0 k^\star}(0) = 100$

Markov modulation: random effects

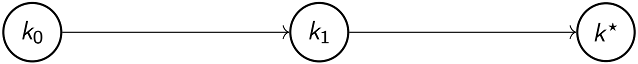


Markov process with incidents:

$$X_{k_0 k^\star}(t) = \begin{cases} 0 & \text{if no incident on road } k_0 k^\star \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

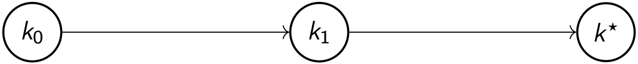
The speed on link $k_0 k^\star$ can be defined dependent on $X_{k_0 k^\star}(t)$,
e.g.
$v_{k_0 k^\star}(0) = 100$
$v_{k_0 k^\star}(1) = 30.$

Markov modulation: random effects



,

Markov modulation: random effects

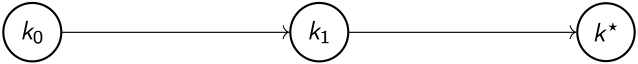


MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

$$X_i(t) = \begin{cases} 0 & \text{if no incident on road i at time } t \\ 1 & \text{oth.} \end{cases}$$

,

Markov modulation: random effects

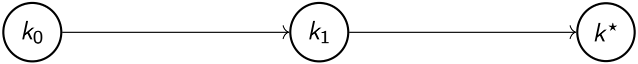


MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

$$X_i(t) = \begin{cases} 0 & \text{if no incident on road i at time } t \\ 1 & \text{oth.} \end{cases}$$

The speed on $k_0 k_1$ and $k_1 k^\star$ can be defined dependent on $B(t)$,

Markov modulation: random effects

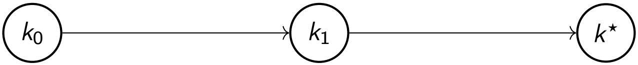


MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

$$X_i(t) = \begin{cases} 0 & \text{if no incident on road i at time } t \\ 1 & \text{oth.} \end{cases}$$

The speed on $k_0 k_1$ and $k_1 k^\star$ can be defined dependent on $B(t)$, e.g.
$v_{k_0 k_1}((0,0)) = 100$

Markov modulation: random effects



MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

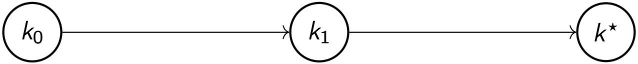
$$X_i(t) = \begin{cases} 0 & \text{if no incident on road i at time } t \\ 1 & \text{oth.} \end{cases}$$

The speed on $k_0 k_1$ and $k_1 k^\star$ can be defined dependent on $B(t)$, e.g.

$v_{k_0 k_1}((0,0)) = 100$

$v_{k_0 k_1}((1,0)) = 30$

Markov modulation: random effects



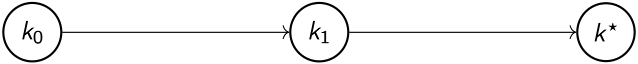
MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

$$X_i(t) = \begin{cases} 0 & \text{if no incident on road i at time } t \\ 1 & \text{oth.} \end{cases}$$

The speed on $k_0 k_1$ and $k_1 k^\star$ can be defined dependent on $B(t)$, e.g.

$v_{k_0 k_1}((0,0)) = 100$
$v_{k_0 k_1}((1,0)) = 30$
$v_{k_0 k_1}((0,1)) = 70$

Markov modulation: random effects



MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

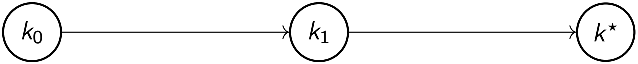
$$X_i(t) = \begin{cases} 0 & \text{if no incident on road i at time } t \\ 1 & \text{oth.} \end{cases}$$

The speed on $k_0 k_1$ **and** $k_1 k^\star$ **can be defined dependent on** $B(t)$, e.g.

$v_{k_0 k_1}((0,0)) = 100$
$v_{k_0 k_1}((1,0)) = 30$
$v_{k_0 k_1}((0,1)) = 70$
$v_{k_0 k_1}((1,1)) = 20$

NET WORKS THENETWORKCENTER.NL

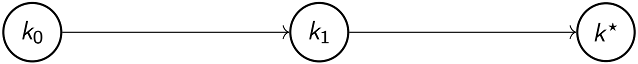
Markov modulation: random effects



MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

$$X_i(t) = \begin{cases} 0 & \text{if no incident on road i at time } t \\ 1 & \text{oth.} \end{cases}$$

Markov modulation: random effects



MP with incidents: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t))$,

$$X_i(t) = \begin{cases} 0 & \text{if free-flow speed on road i at time } t \\ 1 & \text{if incident on road } i \text{ at time } t \\ 2 & \text{if recovery state on road } i \text{ at time } t \end{cases}$$

Markov modulation

We can extend our framework by adding more edges.
Notation: $k_1\ell_1, k_2\ell_2, \ldots, k_n\ell_n$.

Markov modulation

We can extend our framework by adding more edges.
Notation: $k_1\ell_1, k_2\ell_2, \ldots, k_n\ell_n$.

Markovian background process:
$B(t) = \big(X_{k_1\ell_1}(t), \ldots, X_{k_n\ell_n}(t)\big)$

NET WORKS
THENETWORKCENTER.NL

Markov modulation

We can extend our framework by adding more edges.
Notation: $k_1\ell_1, k_2\ell_2, \ldots, k_n\ell_n$.

Markovian background process:
$B(t) = \big(X_{k_1\ell_1}(t), \ldots, X_{k_n\ell_n}(t)\big)$

Note:

Markov modulation

We can extend our framework by adding more edges.
Notation: $k_1\ell_1, k_2\ell_2, \ldots, k_n\ell_n$.

Markovian background process:
$B(t) = \left(X_{k_1\ell_1}(t), \ldots, X_{k_n\ell_n}(t)\right)$

Note:

- $X_{k_i\ell_i}(t)$ may be any Markov Process

NET WORKS
THENETWORKCENTER.NL

Markov modulation

We can extend our framework by adding more edges.
Notation: $k_1\ell_1, k_2\ell_2, \ldots, k_n\ell_n$.

Markovian background process:
$B(t) = \left(X_{k_1\ell_1}(t), \ldots, X_{k_n\ell_n}(t)\right)$

Note:

- $X_{k_i\ell_i}(t)$ may be any Markov Process
- $X_{k_i\ell_i}(t), X_{k_j\ell_j}(t)$ evolve independently;

Markov modulation

We can extend our framework by adding more edges.
Notation: $k_1\ell_1, k_2\ell_2, \dots, k_n\ell_n$.

Markovian background process:
$B(t) = \left(X_{k_1\ell_1}(t), \dots, X_{k_n\ell_n}(t)\right)$

Note:

- $X_{k_i\ell_i}(t)$ may be any Markov Process
- $X_{k_i\ell_i}(t), X_{k_j\ell_j}(t)$ evolve independently;
- Dependence introduced by $v_{k_i\ell_i}(B(t))$;

Markov modulation: deterministic patterns

Markov modulation: deterministic patterns

Transition times in our Markov model are exponential:

$$\mathbb{E}[\mathsf{Exp}(\lambda)] = \frac{1}{\lambda} \qquad \mathsf{Var}[\mathsf{Exp}(\lambda)] = \frac{1}{\lambda^2}$$

Markov modulation: deterministic patterns

Transition times in our Markov model are exponential:

$$\mathbb{E}[\mathsf{Exp}(\lambda)] = \frac{1}{\lambda} \qquad \mathsf{Var}[\mathsf{Exp}(\lambda)] = \frac{1}{\lambda^2}$$

What if we have events which are much more deterministic?

Markov modulation: deterministic patterns

Transition times in our Markov model are exponential:

$$\mathbb{E}[\mathsf{Exp}(\lambda)] = \frac{1}{\lambda} \qquad \mathsf{Var}[\mathsf{Exp}(\lambda)] = \frac{1}{\lambda^2}$$

What if we have events which are much more deterministic?

To include this in our framework, model these events by sums of exponentials ('Erlang phases').

NET WORKS
THENETWORKCENTER.NL

Markov modulation: deterministic patterns

Transition times in our Markov model are exponential:

$$\mathbb{E}[\mathrm{Exp}(\lambda)] = \frac{1}{\lambda} \qquad \mathrm{Var}[\mathrm{Exp}(\lambda)] = \frac{1}{\lambda^2}$$

What if we have events which are much more deterministic?

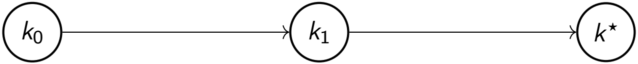
To include this in our framework, model these events by sums of exponentials ('Erlang phases'). Realize:

$$\mathbb{E}[\sum_{m=1}^{n} \mathrm{Exp}_m(\lambda n)] = \frac{1}{\lambda}$$

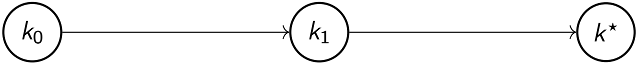
$$\mathrm{Var}[\sum_{m=1}^{n} \mathrm{Exp}_m(\lambda n)] = \frac{1}{\lambda^2 n}.$$

NET WORKS
THENETWORKCENTER.NL

Markov modulation: deterministic patterns

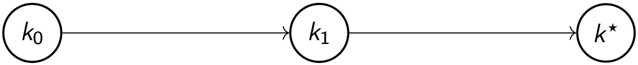


Markov modulation: deterministic patterns

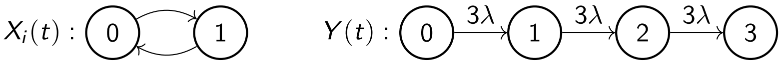


MP with incidents and weather: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t), Y(t))$

Markov modulation: deterministic patterns



MP with incidents and weather: $B(t) = (X_{k_0 k_1}(t), X_{k_1 k^\star}(t), Y(t))$



Markov modulation

We can extend our framework by adding more edges.
Notation: $k_1\ell_1, k_2\ell_2, \ldots, k_n\ell_n$.

Markovian background process:
$B(t) = \big(X_{k_1\ell_1}(t), \ldots, X_{k_n\ell_n}(t),\ Y(t)\big)$

Note:

- $X_{k_i\ell_i}(t)$ may be any Markov Process
- Given $Y(t)$, $X_{k_i\ell_i}(t), X_{k_j\ell_j}(t)$ evolve independently;
- Dependence introduced by $v_{k_i\ell_i}(B(t))$;

Markovian Velocity Model

- Can handle stochasticity
- Can handle time-dependence
- Can handle correlation
- Extreme flexibility (phase-type counterparts)
- Tractability (LST of travel times)

THENETWORKCENTER.NL

Outline

1. **Markovian velocity model**
2. Find the optimal route
3. Determine the optimal departure time
4. Related and other work

Outline

1. Markovian velocity model
2. **Find the optimal route**
3. Determine the optimal departure time
4. Related and other work

Outline

1. Markovian velocity model
2. **Find the optimal route**: minimize expected travel time
3. Determine the optimal departure time
4. Related and other work

Dynamic routing

Dynamic routing: vehicle is allowed to adapt route

Dynamic routing

Question: does a minimizing policy exist?

Dynamic routing

Question: does a minimizing policy exist?
Answer: Yes

Dynamic routing

Question: does a minimizing policy exist?
Answer: Yes

$$V(k, s_0) = \mathbb{E}_{s_0}[\tau_{k\ell}] + \sum_s \mathbb{P}(B(\tau_{k\ell}) = s) V(\ell, s)$$

Dynamic routing

Question: does a minimizing policy exist?
Answer: Yes

$$V(k, s_0) = \mathbb{E}_{s_0}[\tau_{k\ell}] + \sum_s \mathbb{P}(B(\tau_{k\ell}) = s) V(\ell, s)$$

$\implies$ Bellman equation

Dynamic routing

Question: does a minimizing policy exist?
Answer: Yes

$$V(k, s_0) = \mathbb{E}_{s_0}[\tau_{k\ell}] + \sum_s \mathbb{P}(B(\tau_{k\ell}) = s) V(\ell, s)$$

$\Longrightarrow$ Bellman equation
$\Longrightarrow$ Minimizing policy satisfies Bellman optimality equations

Dynamic routing

Question: does a minimizing policy exist?
Answer: Yes

$$V(k, s_0) = \mathbb{E}_{s_0}[\tau_{k\ell}] + \sum_{s} \mathbb{P}(B(\tau_{k\ell}) = s) V(\ell, s)$$

$\Longrightarrow$ Bellman equation
$\Longrightarrow$ Minimizing policy satisfies Bellman optimality equations
$\Longrightarrow$ DP or linear programming

NET WORKS
THENETWORKCENTER.NL

Dynamic routing

Question: does a minimizing policy exist?
Answer: Yes

$$V(k, s_0) = \mathbb{E}_{s_0}[\tau_{k\ell}] + \sum_s \mathbb{P}(B(\tau_{k\ell}) = s)V(\ell, s)$$

$\implies$ Bellman equation
$\implies$ Minimizing policy satisfies Bellman optimality equations
$\implies$ DP or linear programming

Note: not efficient!

Dynamic Routing

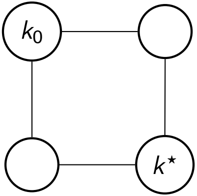


Figure: 2×2-network

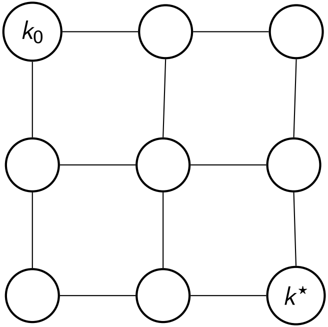
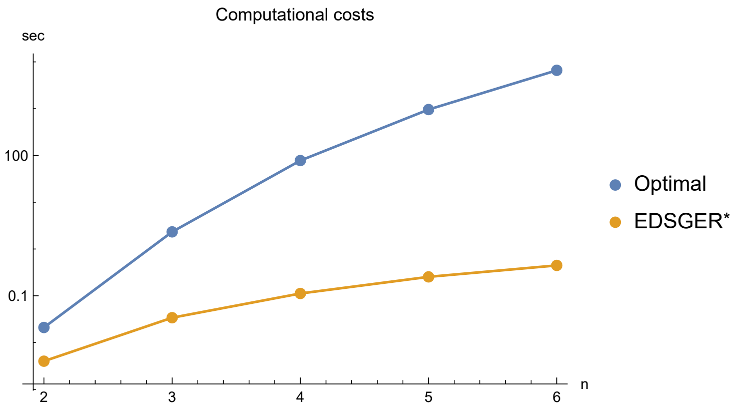


Figure: 3×3-network

Dynamic Routing



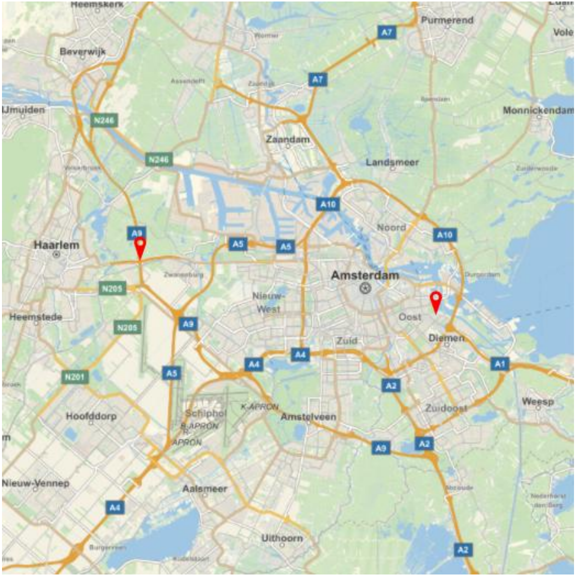
EDSGER*

Algorithm for Expected Delay on a Stochastic Graph with Efficient Routing, also known as

EDSGER*.

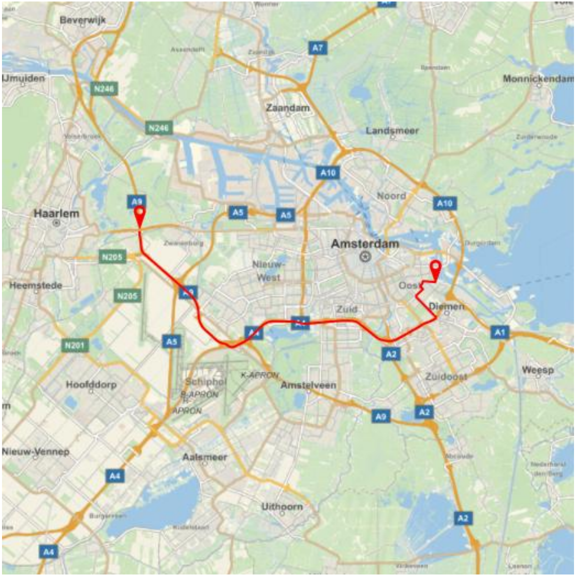
Figure: Edsger W. Dijkstra

EDSGER*

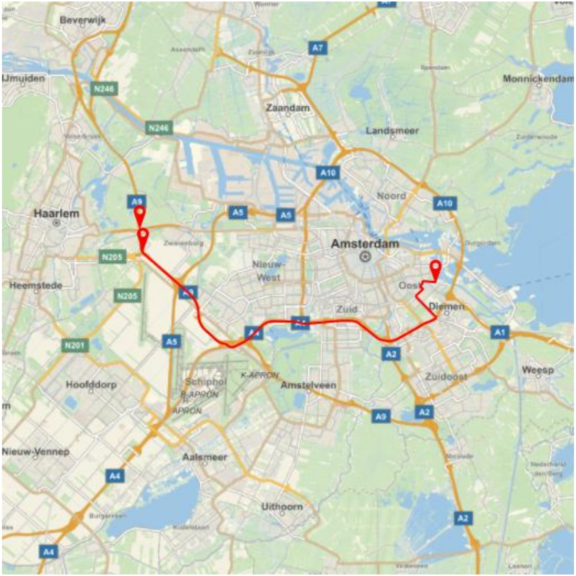


NETWORKS
THENETWORKCENTER.NL

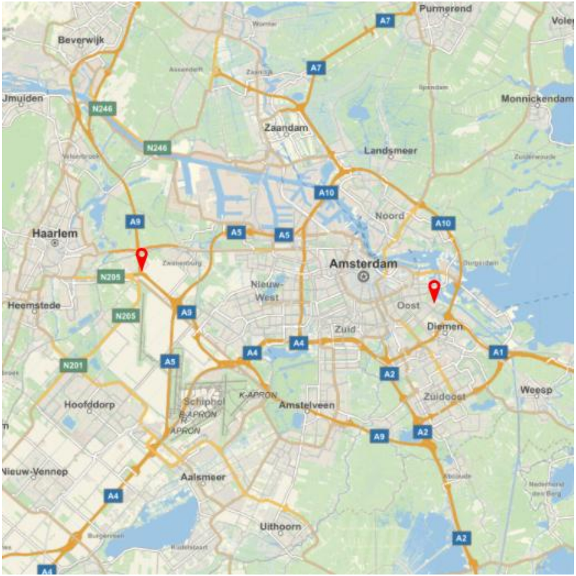
EDSGER*


Beverwijk
IJmuiden
N246
N246
Zaandam
A7
Landsmeer
Monnickendam
A10
Noord
A10
A9
Haarlem
A5
A5
Amsterdam
N205
Nieuw-
West
Oost
Heemstede
N205
A9
Zuid
Diemen
A4
A4
A1
A5
N201
A2
Weesp
Zuidoost
Hoofddorp
Schiphol
K-APRON
Amstelveen
A9
A2
Nieuw-Vennep
Aalsmeer
A4
Uithoorn
A2
A4

EDSGER*

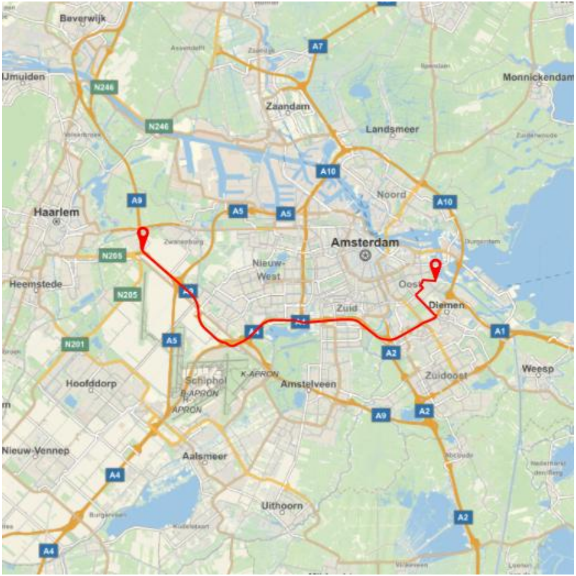

Beverwijk
IJmuiden
N246
Zaandam
Landsmeer
Monnickendam
A7
A10
Noord
Haarlem
A9
A5
Amsterdam
N205
Nieuw-
West
Heemstede
Oost
Diemen
Zuid
A1
A4
A2
N201
Weesp
Zuidoost
Hoofddorp
Schiphol
Amstelveen
Nieuw-Vennep
Aalsmeer
Uithoorn

EDSGER*



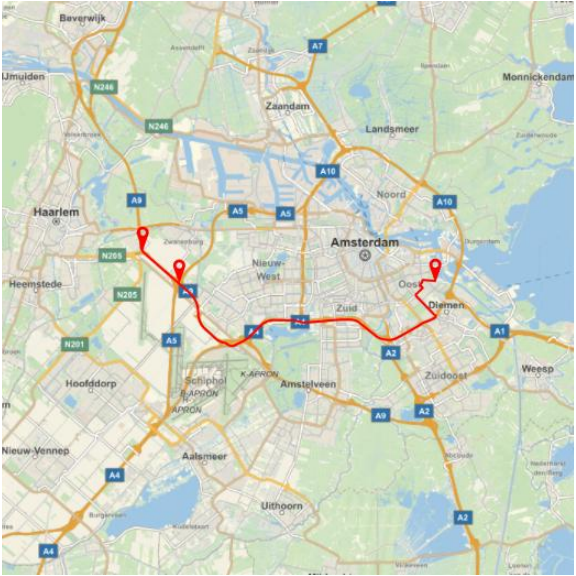
THENETWORKCENTER.NL

EDSGER*



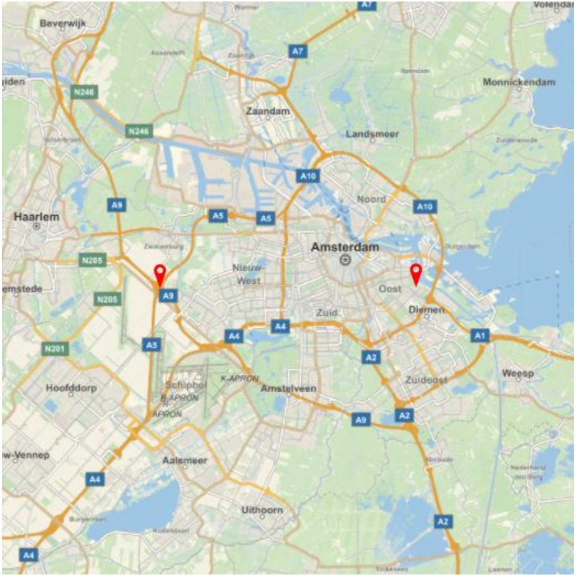
Beverwijk
IJmuiden
N246
N246
Zaandam
A7
Landsmeer
Monnickendam
A10
Noord
A10
Haarlem
A9
A5
A5
Amsterdam
N205
Heemstede
N205
Nieuw-
West
Oost
A9
Zuid
Diemen
A5
A4
A1
A2
A1
N201
Hoofddorp
Schiphol
K-APRON
Amstelveen
Zuidoost
Weesp
A9
A2
Nieuw-Vennep
Aalsmeer
A4
Uithoorn
A2
A4

EDSGER*



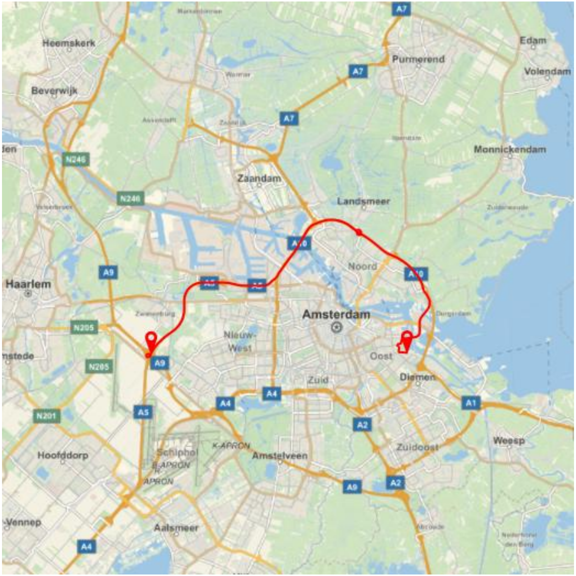
Beverwijk
IJmuiden
N246
Zaandam
A7
Landsmeer
Monnickendam
A10
Noord
Haarlem
A9
A5
Amsterdam
N205
Nieuw-
West
Oost
Heemstede
Zuid
Diemen
A1
A2
N201
A5
Schiphol
Amstelveen
Zuidoost
Weesp
Hoofddorp
Nieuw-Vennep
Aalsmeer
A4
Uithoorn

EDSGER*



Beverwijk
Zaandam
Landsmeer
Monnickendam
Haarlem
Noord
Amsterdam
Nieuw-West
Oost
Zuid
Diemen
Hoofddorp
Schiphol
Amstelveen
Zuidoost
Weesp
Aalsmeer
Uithoorn
A1
A2
A4
A5
A7
A9
A10
N201
N205
N246

EDSGER*


Heemskerk
Beverwijk
Purmerend
Edam
Volendam
Monnickendam
Zaandam
Landsmeer
Noord
Haarlem
Amsterdam
Nieuw-West
Oost
Zuid
Diemen
Zuidoost
Weesp
Amstelveen
Schiphol
Hoofddorp
Aalsmeer
Vennep
A7
A10
A5
A9
A4
A2
A1
N246
N205
N201

EDSGER*

EDSGER*

Shortest path algorithm within EDSGER*: Dijkstra-like

EDSGER⋆

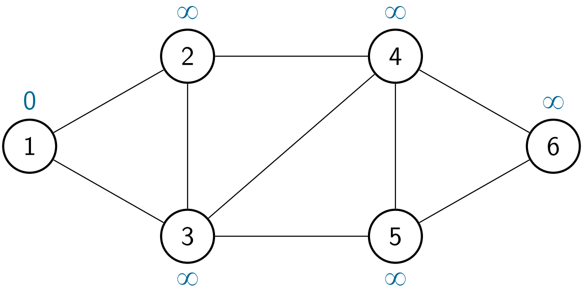
Shortest path algorithm within EDSGER⋆: Dijkstra-like

Assigns labels and iteratively updates the labels of these nodes:

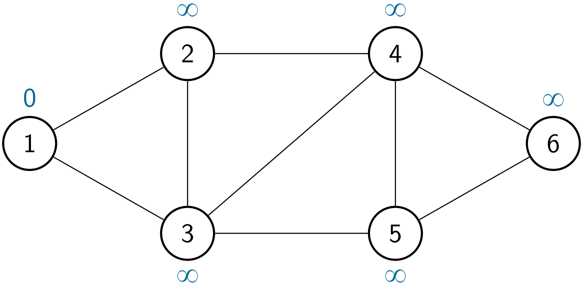
EDSGER*

Shortest path algorithm within EDSGER*: Dijkstra-like

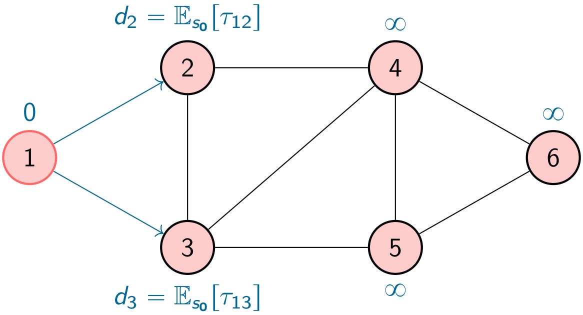
Assigns labels and iteratively updates the labels of these nodes:



EDSGER*

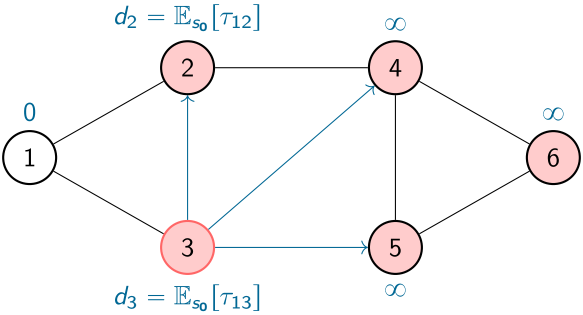


EDSGER*

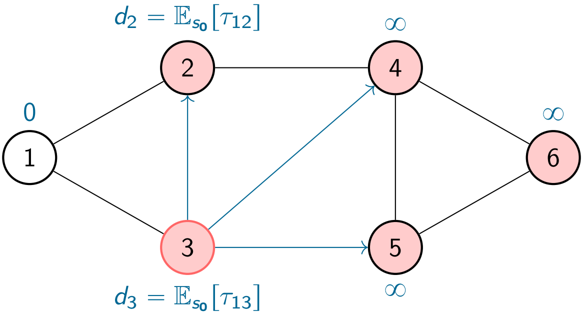


$d_2 = \mathbb{E}_{s_0}[\tau_{12}]$
∞
0
1
2
3
4
5
6
∞
$d_3 = \mathbb{E}_{s_0}[\tau_{13}]$
∞

EDSGER*


$d_2 = \mathbb{E}_{s_0}[\tau_{12}]$
∞
0
∞
1
2
3
4
5
6
∞
$d_3 = \mathbb{E}_{s_0}[\tau_{13}]$

EDSGER*



$$d_5 = d_3 + \sum_{s'} p_{s_0 s'} \mathbb{E}_{s'}[\tau_{35}]$$

EDSGER*

$$d_5 = d_3 + \sum_{s'} p_{s_0 s'} \mathbb{E}_{s'}[\tau_{35}]$$

Note:

EDSGER*

$$d_5 = d_3 + \sum_{s'} p_{s_0 s'} \mathbb{E}_{s'}[\tau_{35}]$$

Note:

- Computation of expectation and transition probabilities involves a term e^Q

EDSGER*

$$d_5 = d_3 + \sum_{s'} p_{s_0 s'} \mathbb{E}_{s'}[\tau_{35}]$$

Note:

- Computation of expectation and transition probabilities involves a term e^Q
- If $|A| = n$ and every arc has two states, Q has dimension 2^n, which grows exponentially

NET WORKS
THENETWORKCENTER.NL

$$d_5 = d_3 + \sum_{s'} p_{s_0 s'} \mathbb{E}_{s'}[\tau_{35}]$$

Note:

- Computation of expectation and transition probabilities involves a term e^Q
- If $|A| = n$ and every arc has two states, Q has dimension 2^n, which grows exponentially

Workaround:

EDSGER*

$$d_5 = d_3 + \sum_{s'} p_{s_0 s'} \mathbb{E}_{s'}[\tau_{35}]$$

Note:

- Computation of expectation and transition probabilities involves a term e^Q
- If $|A| = n$ and every arc has two states, Q has dimension 2^n, which grows exponentially

Workaround:

- local-correlation

Local-correlation

EDSGER*

$$d_5 = d_3 + \sum_{s'} p_{s_0 s'} \mathbb{E}_{s'}[\tau_{35}]$$

Note:

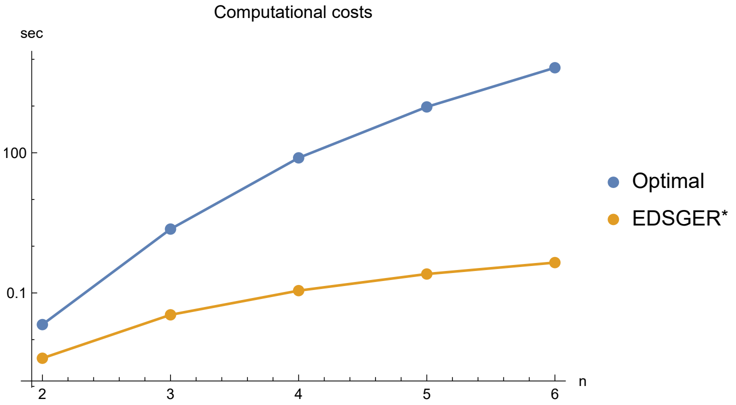
- Computation of expectation and transition probabilities involves a term e^Q
- If $|A| = n$ and every arc has two states, Q has dimension 2^n, which grows exponentially

Workaround:

- local-correlation: $\dim(Q^{\text{local}}) \ll \dim(Q)$

NET WORKS
THENETWORKCENTER.NL

Dynamic Routing



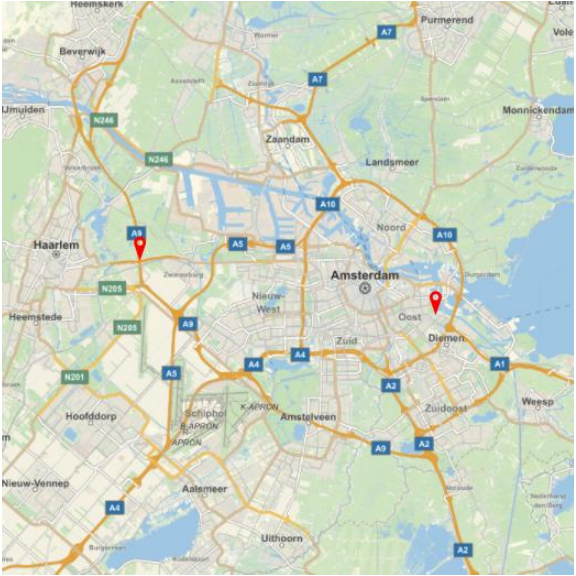
Outline

1. Markovian velocity model
2. **Find the optimal route**
3. Determine the optimal departure time
4. Related and other work

Outline

1. Markovian velocity model
2. Find the optimal route
3. **Determine the optimal departure time**
4. Related and other work

Random Environment



Random Environment

Travelers do not want to:

Random Environment

Travelers do not want to:

- Arrive too late at the destination

Random Environment

Travelers do not want to:

- Arrive too late at the destination
- Depart too early from their origin

Random Environment

Travelers do not want to:

- Arrive too late at the destination
- Depart too early from their origin

Optimal departure time:
Latest time of departure for which a chosen on-time arrival probability can be guaranteed.

NET WORKS
THENETWORKCENTER.NL

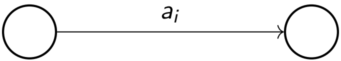
Challenges

Challenges

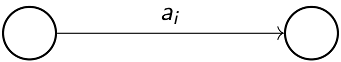
Trip travel times in real traffic networks suffer from:

- Time-dependence: daily patterns, seasonality
- Stochasticity: random effects that lead to delays

Markovian Velocity Model



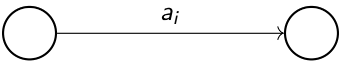
Markovian Velocity Model



Markov process with incidents:

$$X_{a_i}(t) = \begin{cases} 0 & \text{if no incident on link } a_i \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

Markovian Velocity Model



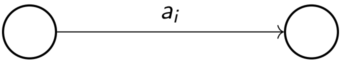
Markov process with incidents:

$$X_{a_i}(t) = \begin{cases} 0 & \text{if no incident on link } a_i \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

Assumption: for $i \neq j$, processes $X_{a_i}(t), X_{a_j}(t)$ evolve independently.

NET WORKS
THENETWORKCENTER.NL

Markovian Velocity Model



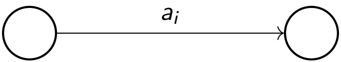
Markov process with incidents:

$$X_{a_i}(t) = \begin{cases} 0 & \text{if no incident on link } a_i \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

Assumption: for $i \neq j$, processes $X_{a_i}(t), X_{a_j}(t)$ evolve independently.
Then, process $X_{a_i}(t)$ is described by its initial state, and Q-matrix:

$$Q_{a_i} = \begin{bmatrix} -\alpha_i & \alpha_i \\ \beta_i & -\beta_i \end{bmatrix} \qquad \alpha_i, \beta_i \in \mathbb{R}_{>0}.$$

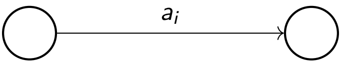
Markovian Velocity Model



Markov process with incidents:

$$X_{a_i}(t) = \begin{cases} 0 & \text{if no incident on link } a_i \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

Markovian Velocity Model



Markov process with incidents:

$$X_{a_i}(t) = \begin{cases} 0 & \text{if no incident on link } a_i \text{ at time } t \\ 1 & \text{oth.} \end{cases}$$

Impact on travel time modeled via vehicle speeds:
Speed at time t equals

- $v_{a_i}(0)$ if $X_{a_i}(t) = 0$
- $v_{a_i}(1)$ if $X_{a_i}(t) = 1$.

Markovian Velocity Model

Markovian Velocity Model

With

$$B(t) := (X_{a_1}(t), \ldots, X_{a_n}(t)),$$

$B(t)$ is a Markovian background process that tracks the occurrence of incidents, and their affect on arc speeds.

Optimal Departure Time Advice

Optimal Departure Time Advice

To determine the optimal departure time, we need:

Optimal Departure Time Advice

To determine the optimal departure time, we need:

- The desired arrival time M

Optimal Departure Time Advice

To determine the optimal departure time, we need:

- The desired arrival time M
- The desired on-time arrival probability η

Optimal Departure Time Advice

To determine the optimal departure time, we need:

- The desired arrival time M
- The desired on-time arrival probability η
- The current state of the network $B(0)$

Optimal Departure Time Advice

To determine the optimal departure time, we need:

- The desired arrival time M
- The desired on-time arrival probability η
- The current state of the network $B(0)$
- *A path P to travel*

Optimal Departure Time Advice

To determine the optimal departure time, we need:

- The desired arrival time M
- The desired on-time arrival probability η
- The current state of the network $B(0)$
- *A path P to travel*

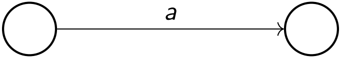
We define the optimal departure time as

$$t_0^* := \sup\{t \geqslant 0 \mid \mathbb{P}\big(t + T_t \leqslant M \mid B(0)\big) \geqslant \eta\},$$

where T_t is the travel time when departing at time t.

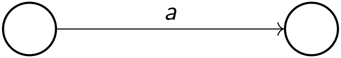
Travel Time Distribution

Travel Time Distribution



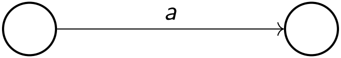
1. Immediate departure, one link ($P = \{a\}$, T_0):

Travel Time Distribution



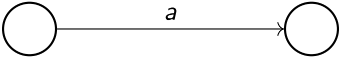
1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Travel time distribution described by Q_{a_i} and $X_a(0)$

Travel Time Distribution



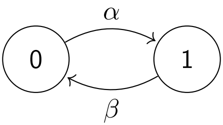
1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Travel time distribution described by Q_{a_i} and $X_a(0)$
   - LST known, inversion unsuccessful

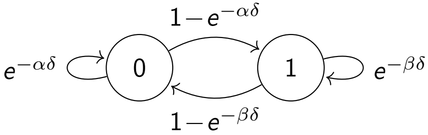
Travel Time Distribution



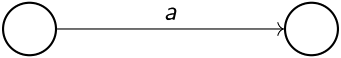
1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Travel time distribution described by Q_{a_i} and $X_a(0)$
   - LST known, inversion unsuccessful
   - PMF via discretization approach:

Discretization





Travel Time Distribution



1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Travel time distribution described by Q_{a_i} and $X_a(0)$
   - LST known, inversion unsuccessful
   - PMF via discretization approach:

$$\begin{cases} t_1 & \text{w.p. } p_1 \\ t_2 & \text{w.p. } p_2 \\ \vdots & \end{cases}$$

Travel Time Distribution

1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Given $X_a(0)$, approximating PMF can be computed.

Travel Time Distribution

1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Given $X_a(0)$, approximating PMF can be computed.
2. Immediate departure, path ($P = \{a_1, \ldots, a_m\}$, T_0):

Travel Time Distribution

1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Given $X_a(0)$, approximating PMF can be computed.
2. Immediate departure, path ($P = \{a_1, \ldots, a_m\}$, T_0):
   - Given $B(0)$, path distribution via convolution.

NET WORKS
THENETWORKCENTER.NL

Travel Time Distribution

1. Immediate departure, one link $(P = \{a\},\ T_0)$:
   - Given $X_a(0)$, approximating PMF can be computed.
2. Immediate departure, path $(P = \{a_1, \ldots, a_m\},\ T_0)$:
   - Given $B(0)$, path distribution via convolution.
3. Departure at time t, path $(P = \{a_1, \ldots, a_m\},\ T_t)$:

NET WORKS
THENETWORKCENTER.NL

Travel Time Distribution

1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Given $X_a(0)$, approximating PMF can be computed.
2. Immediate departure, path ($P = \{a_1, \ldots, a_m\}$, T_0):
   - Given $B(0)$, path distribution via convolution.
3. Departure at time t, path ($P = \{a_1, \ldots, a_m\}$, T_t):
   - State $B(t)$ at departure time t unknown

Travel Time Distribution

1. Immediate departure, one link ($P = \{a\},\ T_0$):
   - Given $X_a(0)$, approximating PMF can be computed.
2. Immediate departure, path ($P = \{a_1, \dots, a_m\},\ T_0$):
   - Given $B(0)$, path distribution via convolution.
3. Departure at time t, path ($P = \{a_1, \dots, a_m\},\ T_t$):
   - State $B(t)$ at departure time t unknown
   - Let $P' = \{a_0, P\}$, with a_0 a fictional link with PMF set as t w.p. 1.

Travel Time Distribution

1. Immediate departure, one link ($P = \{a\}$, T_0):
   - Given $X_a(0)$, approximating PMF can be computed.
2. Immediate departure, path ($P = \{a_1, \ldots, a_m\}$, T_0):
   - Given $B(0)$, path distribution via convolution.
3. Departure at time t, path ($P = \{a_1, \ldots, a_m\}$, T_t):
   - State $B(t)$ at departure time t unknown
   - Let $P' = \{a_0, P\}$, with a_0 a fictional link with PMF set as t w.p. 1.
   - Traveling P at time t equals traveling P' at time 0.

Travel Time Distribution

1. Immediate departure, one link ($P = \{a\},\ T_0$):
   - Given $X_a(0)$, approximating PMF can be computed.
2. Immediate departure, path ($P = \{a_1, \ldots, a_m\},\ T_0$):
   - Given $B(0)$, path distribution via convolution.
3. Departure at time t, path ($P = \{a_1, \ldots, a_m\},\ T_t$):
   - State $B(t)$ at departure time t unknown
   - Let $P' = \{a_0, P\}$, with a_0 a fictional link with PMF set as t w.p. 1.
   - Traveling P at time t equals traveling P' at time 0.
   - Given $B(0)$, path distribution via convolution.

Determining the Optimal Departure Time

For each $t \geqslant 0$, we are now able to compute

$$\mathbb{P}\big(t + T_t \leqslant M \,|\, B(0)\big), \quad t \geqslant 0.$$

Determining the Optimal Departure Time

For each $t \geqslant 0$, we are now able to compute

$$\mathbb{P}\big(t + T_t \leqslant M \,|\, B(0)\big), \quad t \geqslant 0.$$

The MVM has the FIFO property: for $t' \leqslant \tilde{t}$ we have

$$t' + T'_{t'} \leqslant \tilde{t} + T_{\tilde{t}}.$$

NET WORKS
THENETWORKCENTER.NL

Determining the Optimal Departure Time

For each $t \geqslant 0$, we are now able to compute

$$\mathbb{P}\big(t + T_t \leqslant M \,|\, B(0)\big), \quad t \geqslant 0.$$

The MVM has the FIFO property: for $t' \leqslant \tilde{t}$ we have

$$t' + T'_{t'} \leqslant \tilde{t} + T_{\tilde{t}}.$$

- Probability monotone decreasing in the departure time
- Bisection algorithm outputs optimal departure time

NET WORKS
THENETWORKCENTER.NL

Optimal Departure Time Advice

So far, we have assumed that the path to the destination is known beforehand.

Optimal Departure Time Advice

So far, we have assumed that the path to the destination is known beforehand.

For an origin-destination pair:

- Optimal: bisection on output label-correcting algorithm

Optimal Departure Time Advice

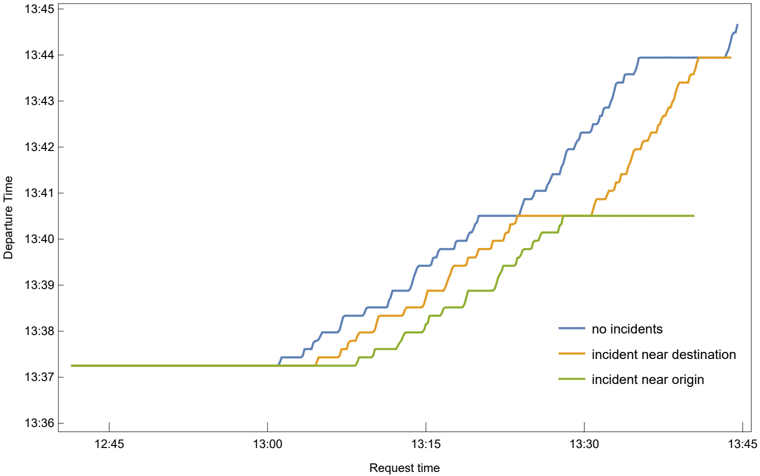
So far, we have assumed that the path to the destination is known beforehand.

For an origin-destination pair:

- Optimal: bisection on output label-correcting algorithm
- Efficient alternative: k-shortest paths (parallel)

Numerical Experiments

Numerical Experiments



Online Optimal Departure time

Online Optimal Departure time

- When waiting for departure, new information on the state of the $B(\cdot)$ becomes available

Online Optimal Departure time

- When waiting for departure, new information on the state of the $B(\cdot)$ becomes available
- Ideally, the optimal departure time is updated such that it incorporates the latest state of the network

NET WORKS
THENETWORKCENTER.NL

Online Optimal Departure time

- When waiting for departure, new information on the state of the $B(\cdot)$ becomes available
- Ideally, the optimal departure time is updated such that it incorporates the latest state of the network

Online Optimal Departure time

- When waiting for departure, new information on the state of the $B(\cdot)$ becomes available
- Ideally, the optimal departure time is updated such that it incorporates the latest state of the network

Therefore, we also consider an online version of the problem:

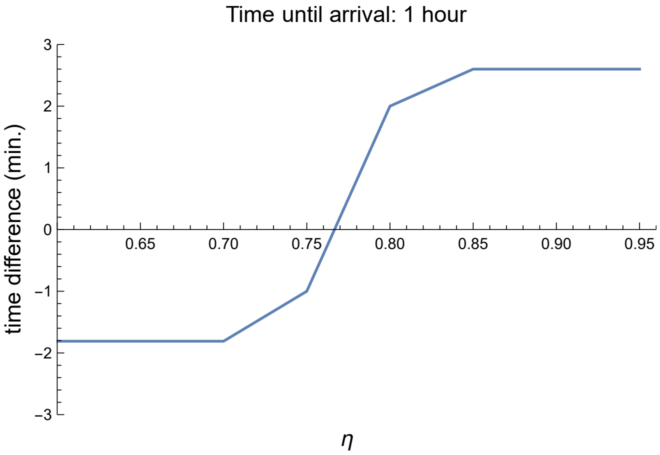
$$t_u^* := \sup\{t \geqslant u \mid \mathbb{P}\big(t + T_t \leqslant M \mid B(u)\big) \geqslant \eta\}$$

Online vs Offline Optimal Departure time

For a range of on-time arrival probabilities η, we look at the difference in minutes for the online and offline departure time:

Online vs Offline Optimal Departure time

For a range of on-time arrival probabilities η, we look at the difference in minutes for the online and offline departure time:



Outline

1. Markovian velocity model
2. Find the optimal route
3. **Determine the optimal departure time**
4. Related and other work

N. Levering, M. Boon, M. Mandjes, and R. Núñez-Queija
A framework for efficient dynamic routing under stochastically varying conditions
Transportation Research Part B: Methodological (2022), 160, 97-124.

R. Kamphuis, N. Levering, and M. Mandjes
Optimal departure-time advice in road networks with stochastic disruptions
Under review (Pre-print, arXiv:2208.14516 [math.OC]).

NET WORKS
THENETWORKCENTER.NL

Outline

1. Markovian velocity model
2. Find the optimal route
3. Determine the optimal departure time
4. **Related and other work**

MVM: Operationalization

N. Levering, M. Boon, and M. Mandjes
Estimating probability distributions of travel times by fitting a Markovian velocity model
IEEE Transactions on Intelligent Transportation Systems (in press).

MVM: Operationalization

N. Levering, M. Boon, and M. Mandjes
Estimating probability distributions of travel times by fitting a Markovian velocity model
IEEE Transactions on Intelligent Transportation Systems (in press).

Idea:

- When using the MVM, it is assumed that Q and $v_a(B(t))$ are known

MVM: Operationalization

N. Levering, M. Boon, and M. Mandjes
Estimating probability distributions of travel times by fitting a Markovian velocity model
IEEE Transactions on Intelligent Transportation Systems (in press).

Idea:

- When using the MVM, it is assumed that Q and $v_a(B(t))$ are known
- How can these be found for a given highway network?

MVM: Operationalization

N. Levering, M. Boon, and M. Mandjes
Estimating probability distributions of travel times by fitting a Markovian velocity model
IEEE Transactions on Intelligent Transportation Systems (in press).

Idea:

- When using the MVM, it is assumed that Q and $v_a(B(t))$ are known
- How can these be found for a given highway network?
- Present methodology, with lists of incidents and a data base of recorded speeds as input
- Proof of concept in Dutch highway network

NET WORKS
THENETWORKCENTER.NL

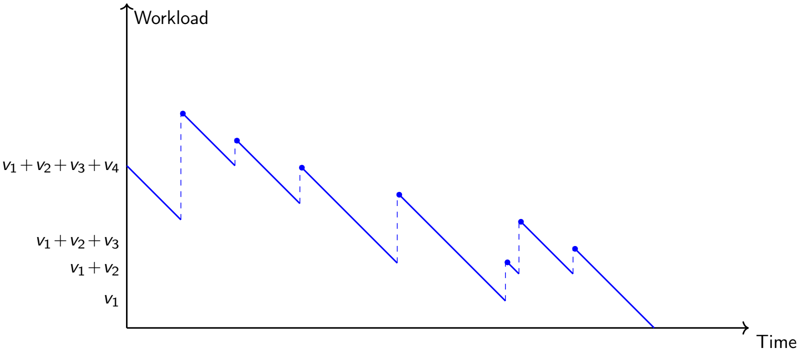
Road traffic streams

N. Levering and R. Núñez-Queija
Input rate control in stochastic road traffic networks: Effective bandwidths
Under review.

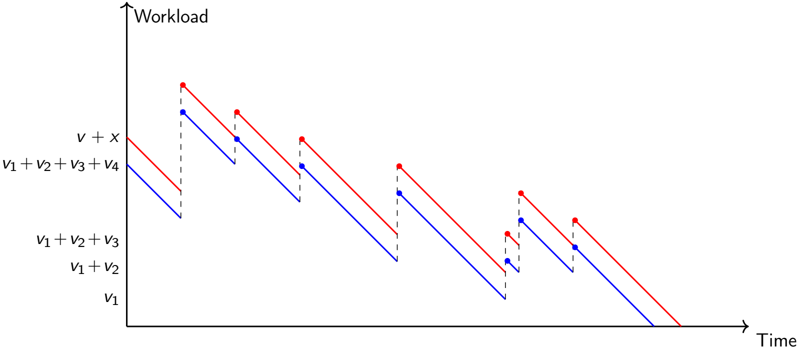
R. Kamphuis, N. Levering, and M. Mandjes
Optimal routing advice in highway networks with stochastic fundamental diagram dynamics
Working paper.

Queueing Theory

Queueing Theory

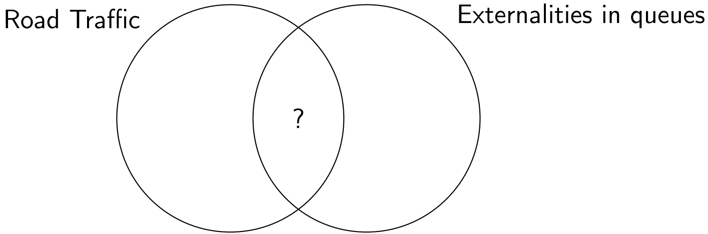


Queueing Theory



Queueing Theory

R. Jacobovic, N. Levering, and O. Boxma
Externalities in the M/G/1 queue: LCFS-PR versus FCFS
Queueing Systems (in press).


Road Traffic
Externalities in queues
?

Thank you for the attention!