Resilience of Traffic Networks with Partially Controlled Routing

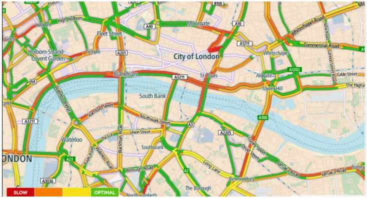
Gianluca Bianchin
in collaboration with: Fabio Pasqualetti and Soumya Kundu

Department of Mechanical Engineering
University of California, Riverside

Optimization and Control Group
Pacific Northwest National Laboratory

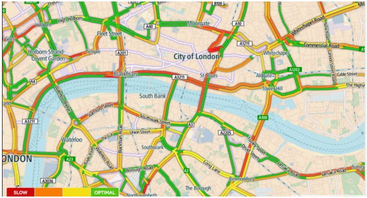
2019 American Control Conference
July 11, 2019 | Philadelphia, USA

Routing in Traffic Networks

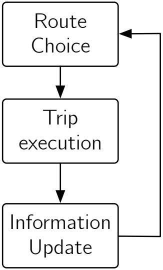


Network routing: captures how travelers respond to congestion

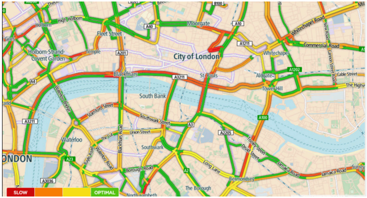
Routing in Traffic Networks



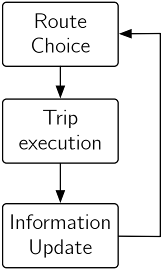
Network routing: captures how travelers respond to congestion

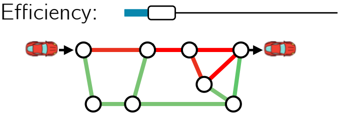


Routing in Traffic Networks

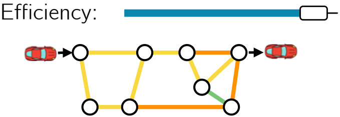


Network routing: captures how travelers respond to congestion





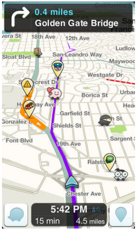
- Minimize individual delay
- Braess Paradox

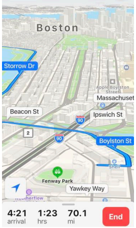


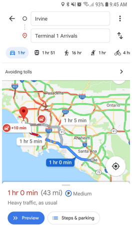
- Do what's best for network
- Can do longer commute

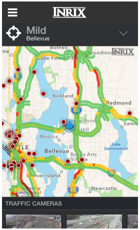
The Introduction of Real-Time Traffic Information

- Commute as fast as we can



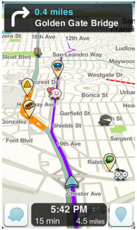


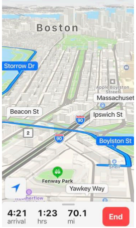


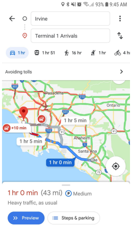


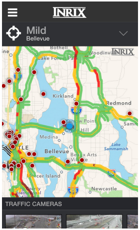
The Introduction of Real-Time Traffic Information

- Commute as fast as we can









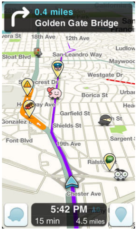
- Information does not necessarily make things better:

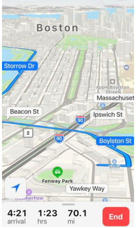

PCWorld
FROM IDG
SUBSCRIBE
NEWS REVIEWS HOW-TO VIDEO DEALS BUSINESS LAPTOPS SMARTPHONES HARDWARE SECURITY SOFTWARE GADGETS
Privacy Encryption Antivirus
AdChoices driving route highway map directions maps
Home / Security
NEWS

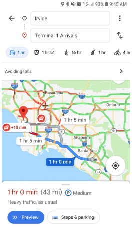
Researcher: Hackers can cause traffic jams by manipulating real-time traffic data

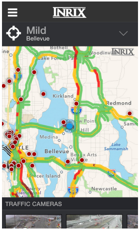
Hackers can influence real-time traffic-flow-analysis systems to make people drive into traffic jams or to keep roads clear in areas where a lot of people use Google or Waze navigation systems, a German researcher demonstrated at BlackHat Europe.

- Commute as fast as we can









- Information does not necessarily make things better:


PCWorld

Home / Security

NEWS

Researcher: Hackers can c
manipulating real-time traf

Hackers can influence real-time traffic
people drive into traffic jams or to kee
people use Google or Waze navigatio
demonstrated at BlackHat Europe.


STORY · Published January 29, 2009 · Last Update May 15, 2015

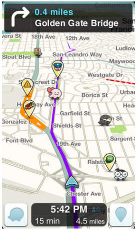
Hackers Crack Into Texas Road Sign, Warn of Zombies Ahead

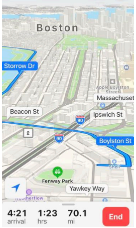
Transportation officials in Texas are scrambling to prevent hackers from changing messages on digital road signs after one sign in Austin was altered to read, "Zombies Ahead."

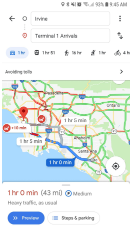
Chris Lippincott, director of media relations for the Texas Department of Transportation, confirmed that a portable traffic sign at Lamar Boulevard and West 15th Street, near the University of Texas at Austin, was hacked into during the early hours of Jan. 19.

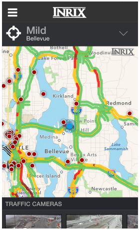
The Introduction of Real-Time Traffic Information

- Commute as fast as we can









- Information does not necessarily make things better:


PCWorld
NEWS REVIEWS HOW-TO VIDEO DEALS BUSINESS LAPTOP
Privacy Encryption Antivirus
Home / Security
NEWS
Researcher: Hackers can c
manipulating real-time traf
Hackers can influence real-time traffic
people drive into traffic jams or to kee
people use Google or Waze navigatio
demonstrated at BlackHat Europe.


FOX NEWS
U.S. World Opinion Politics Entertainment Business Lifestyle TV Fox Nation
STORY · Published January 29, 2009 · Last Update
Hackers Crack Into Tex
Warn of Zombies
Transportation officials in Texas are scrambling to prev
messages on digital road signs after one sign in Austir
Ahead."
Chris Lippincott, director of media relations for the Tex
confirmed that a portable traffic sign at Lamar Bouleva
University of Texas at Austin, was hacked into during tl


gt government technology
MAGAZINE NEWSLETTERS EVENTS PAPERS NAVIGATOR SECTIONS
Cybersecurity GovTech Biz Emerging Tech Cloud Gov Experience More
APPLICATIONS
Navigation App Waze Vulnerable to Hackers
Researchers have found that hackers can break into users accounts, track the users in real time, issue instructions and provide an inaccurate picture of traffic at any given time.
(TNS) -- A serious security breach has been discovered in the Google-owned navigation and traffic monitoring app Waze used by millions of drivers worldwide. It seems that Waze, whose slogan is "Outsmarting Traffic, Together" can be outsmarted by hackers, researchers at the University of California Santa Barbara have found, with serious privacy implications for users and scope for criminal abuse.

Two Emerging Questions

1. Do real-time routing suggestions improve performance?

✓ Yes, under appropriate design

2. What is the impact on the network robustness?

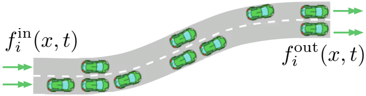
✗ Controlled routing can increase network fragility

Model of Network and Routing

Model of Traffic Network

- Roads dynamics

$$\dot{x}_i = f_i^{\text{in}}(x,t) - f_i^{\text{out}}(x,t)$$



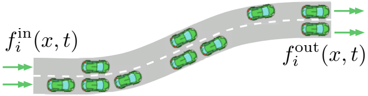
x_i = density [veh/mile] f_i = flow [veh/h]

Model of Traffic Network

- Roads dynamics

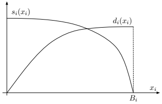
$$\dot{x}_i = f_i^{\text{in}}(x,t) - f_i^{\text{out}}(x,t)$$

x_i = density [veh/mile] $\quad$ f_i = flow [veh/h]



- Cell Transmission Model

$$f_i^{\text{in}}(x,t) \leq s_i(x_i) \quad f_i^{\text{out}}(x,t) \leq d_i(x_i)$$



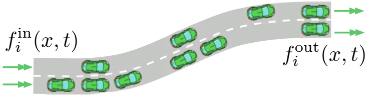
B_i = max occupancy

Model of Traffic Network

- Roads dynamics

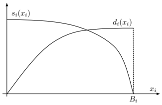
$$\dot{x}_i = f_i^{\text{in}}(x,t) - f_i^{\text{out}}(x,t)$$

x_i = density [veh/mile] $\quad$ f_i = flow [veh/h]



- Cell Transmission Model

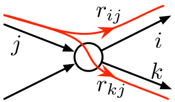
$$f_i^{\text{in}}(x,t) \le s_i(x_i) \qquad f_i^{\text{out}}(x,t) \le d_i(x_i)$$



B_i = max occupancy

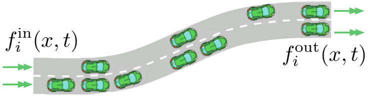
- Flow conservation at intersections

$$f_i^{\text{in}}(x,t) = \sum_j r_{ij} f_j^{\text{out}}(x,t)$$



Model of Traffic Network

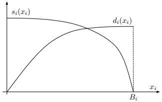
- Roads dynamics



$$\dot{x}_i = f_i^{\text{in}}(x,t) - f_i^{\text{out}}(x,t)$$

x_i = density [veh/mile] $\qquad$ f_i = flow [veh/h]

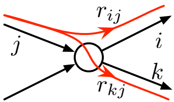
- Cell Transmission Model



B_i = max occupancy

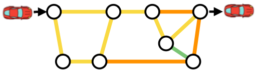
$$f_i^{\text{in}}(x,t) \leq s_i(x_i) \qquad f_i^{\text{out}}(x,t) \leq d_i(x_i)$$

- Flow conservation at intersections



$$f_i^{\text{in}}(x,t) = \sum_j r_{ij} f_j^{\text{out}}(x,t)$$

- Network dynamics:



$$\dot{x} = (R - I) f(x,t) + \lambda$$

Partially Controlled Routing

$$R = \Sigma R^{\mathsf{C}} + (I - \Sigma) R^{\mathsf{S}}, \quad R^{\mathsf{S}} \in \mathcal{R}_{\mathcal{G}}, \quad R^{\mathsf{C}} \in \mathcal{R}_{\mathcal{G}}$$

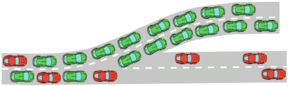
- R^{C} = design parameter
- R^{S} = fixed routing
- $\Sigma = \mathrm{Diag}\{\sigma_1, \dots, \sigma_n\}$

Partially Controlled Routing

$$R = \Sigma R^{\mathsf{C}} + (I - \Sigma) R^{\mathsf{S}}, \quad R^{\mathsf{S}} \in \mathcal{R}_{\mathcal{G}}, \quad R^{\mathsf{C}} \in \mathcal{R}_{\mathcal{G}}$$

- R^{C} = design parameter
- R^{S} = fixed routing
- $\Sigma = \mathrm{Diag}\{\sigma_1, \ldots, \sigma_n\}$

- $\sigma_i \in [0, 1]$ describes penetration level at road i
- σ_i will follow suggestions r_{ij}^{c}
- $(1 - \sigma_i)$ will follow fixed routing r_{ij}^{s}



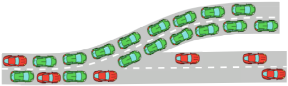
σ_i are link dependent and time-varying

Partially Controlled Routing

$$R = \Sigma R^{\mathsf{C}} + (I - \Sigma) R^{\mathsf{S}}, \quad R^{\mathsf{S}} \in \mathcal{R}_{\mathcal{G}}, \quad R^{\mathsf{C}} \in \mathcal{R}_{\mathcal{G}}$$

- R^{C} = design parameter
- R^{S} = fixed routing
- $\Sigma = \mathrm{Diag}\{\sigma_1, \ldots, \sigma_n\}$

- $\sigma_i \in [0, 1]$ describes penetration level at road i
- σ_i will follow suggestions r^{c}_{ij}
- $(1 - \sigma_i)$ will follow fixed routing r^{s}_{ij}



σ_i are link dependent and time-varying

Routing Design

1. Do real-time routing suggestions improve performance?

$$
\begin{aligned}
\min_{R^{\mathsf{C}}} \quad & \int_0^{\mathcal{H}} x_1 + \cdots + x_n \; dt \\
\text{s.t} \quad & \dot{x} = (R - I) f(x,t) + \lambda \\
& R = \Sigma R^{\mathsf{C}} + (I - \Sigma) R^{\mathsf{S}} \qquad (\sigma = \sigma_0) \\
& R^{\mathsf{C}} \in \mathcal{R}_{\mathcal{G}}, \quad x \leq B
\end{aligned}
$$

- Optimizing network total travel time
- Design parameter: R^{C} (routing suggestions)

Network Resilience

2. What is the impact real-time information on the network robustness?

$$\begin{aligned} \rho(\mathcal{G}, x_0) := \min_{\tilde{\sigma}} \quad & \|\tilde{\sigma} - \sigma_0\|_1 \\ \text{s.t.} \quad & x_i = B_i, \ \text{for some } i \text{ and some } t \end{aligned}$$

- Smallest change in penetration levels that results in spillbacks
- The variable parameter is: σ (penetration rate)

Routing Suggestions Design

Numerically Solving the Optimization

1. Discretization: $x_{k+1} = x_k + T_s((R_k - I)f(x_k) + \lambda_k) := \mathcal{F}(x_k, r_k, \lambda_k)$
2. Vectorization: $r_k = (\Sigma_k^\top \otimes I)r^{\mathsf{c}} + ((I - \Sigma_k)^\top \otimes I)r^{\mathsf{s}} := \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}})$
3. Sparsity: $\sum_i r_{ij}^{\mathsf{c}} = b_j,\ 0 \leq r_{ij}^{\mathsf{c}} \leq 1, \quad (i,j) \in \mathcal{E}$

Optimization on discretized dynamics:

$$
\begin{aligned}
\min_{r^{\mathsf{c}}} \quad & \sum_{k=1}^{h} \phi(x_k) \\
\text{s.t.} \quad & x_{k+1} = \mathcal{F}(x_k, r_k, \lambda_k), && k = 1, \dots, h, \\
& r_k = \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}}), && k = 1, \dots, h, \\
& \sum_i r_{ij} = b_j, && j = 1, \dots, n, \\
& 0 \leq r_{ij}^{\mathsf{c}} \leq 1, && (i,j) \in \mathcal{E}, \\
& x_k \leq B, && k = 1, \dots, h,
\end{aligned}
$$

Because of nonlinearity of $\mathcal{F}$: **nonlinear programming opt. problem**

Numerically Solving the Optimization

1. Discretization: $x_{k+1} = x_k + T_s((R_k - I)f(x_k) + \lambda_k) := \mathcal{F}(x_k, r_k, \lambda_k)$
2. Vectorization: $r_k = (\Sigma_k^\top \otimes I)r^{\mathsf{c}} + ((I - \Sigma_k)^\top \otimes I)r^{\mathsf{s}} := \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}})$
3. Sparsity: $\sum_i r_{ij}^{\mathsf{c}} = b_j,\ 0 \le r_{ij}^{\mathsf{c}} \le 1, \quad (i,j) \in \mathcal{E}$

Optimization on discretized dynamics:

$$
\begin{aligned}
\min_{r^{\mathsf{c}}} \quad & \sum_{k=1}^{h} \phi(x_k) \\
\text{s.t.} \quad & x_{k+1} = \mathcal{F}(x_k, r_k, \lambda_k), && k = 1, \dots, h, \\
& r_k = \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}}), && k = 1, \dots, h, \\
& \sum_i r_{ij} = b_j, && j = 1, \dots, n, \\
& 0 \le r_{ij}^{\mathsf{c}} \le 1, && (i,j) \in \mathcal{E}, \\
& x_k \le B, && k = 1, \dots, h,
\end{aligned}
$$

Because of nonlinearity of $\mathcal{F}$: **nonlinear programming opt. problem**

Numerically Solving the Optimization

1. Discretization: $x_{k+1} = x_k + T_s((R_k - I)f(x_k) + \lambda_k) := \mathcal{F}(x_k, r_k, \lambda_k)$
2. Vectorization: $r_k = (\Sigma_k^\mathsf{T} \otimes I)r^\mathsf{c} + ((I - \Sigma_k)^\mathsf{T} \otimes I)r^\mathsf{s} := \Psi(\sigma_k, r^\mathsf{s}, r^\mathsf{c})$
3. Sparsity: $\sum_i r_{ij}^\mathsf{c} = b_j,\ 0 \leq r_{ij}^\mathsf{c} \leq 1, \quad (i,j) \in \mathcal{E}$

Optimization on discretized dynamics:

$$
\begin{aligned}
\min_{r^\mathsf{c}} \quad & \sum_{k=1}^{h} \phi(x_k) & \\
\text{s.t.} \quad & x_{k+1} = \mathcal{F}(x_k, r_k, \lambda_k), & k = 1, \dots, h, \\
& r_k = \Psi(\sigma_k, r^\mathsf{s}, r^\mathsf{c}), & k = 1, \dots, h, \\
& \sum_i r_{ij} = b_j, & j = 1, \dots, n, \\
& 0 \leq r_{ij}^\mathsf{c} \leq 1, & (i,j) \in \mathcal{E}, \\
& x_k \leq B, & k = 1, \dots, h,
\end{aligned}
$$

Because of nonlinearity of $\mathcal{F}$: **nonlinear programming opt. problem**

Numerically Solving the Optimization

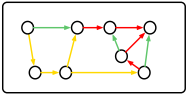
1. Discretization: $x_{k+1} = x_k + T_s((R_k - I)f(x_k) + \lambda_k) := \mathcal{F}(x_k, r_k, \lambda_k)$
2. Vectorization: $r_k = (\Sigma_k^\top \otimes I)r^{\mathsf{c}} + ((I - \Sigma_k)^\top \otimes I)r^{\mathsf{s}} := \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}})$
3. Sparsity: $\sum_i r_{ij}^{\mathsf{c}} = b_j,\ 0 \leq r_{ij}^{\mathsf{c}} \leq 1, \quad (i,j) \in \mathcal{E}$

Optimization on discretized dynamics:

$$
\begin{aligned}
\min_{r^{\mathsf{c}}} \quad & \sum_{k=1}^{h} \phi(x_k) & \\
\text{s.t.} \quad & x_{k+1} = \mathcal{F}(x_k, r_k, \lambda_k), & k = 1, \dots, h, \\
& r_k = \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}}), & k = 1, \dots, h, \\
& \sum_i r_{ij} = b_j, & j = 1, \dots, n, \\
& 0 \leq r_{ij}^{\mathsf{c}} \leq 1, & (i,j) \in \mathcal{E}, \\
& x_k \leq B, & k = 1, \dots, h,
\end{aligned}
$$

Because of nonlinearity of $\mathcal{F}$: **nonlinear programming opt. problem**

Numerically Solving the Optimization

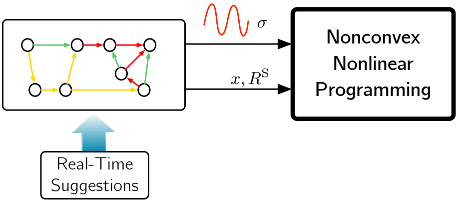


Optimization on discretized dynamics:

$$
\begin{aligned}
\min_{r^{\mathsf{c}}} \quad & \sum_{k=1}^{h} \phi(x_k) & \\
\text{s.t.} \quad & x_{k+1} = \mathcal{F}(x_k, r_k, \lambda_k), & k = 1, \ldots, h, \\
& r_k = \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}}), & k = 1, \ldots, h, \\
& \sum_i r_{ij} = b_j, & j = 1, \ldots, n, \\
& 0 \leq r^{\mathsf{c}}_{ij} \leq 1, & (i,j) \in \mathcal{E}, \\
& x_k \leq B, & k = 1, \ldots, h,
\end{aligned}
$$

Because of nonlinearity of $\mathcal{F}$: **nonlinear programming opt. problem**

Numerically Solving the Optimization

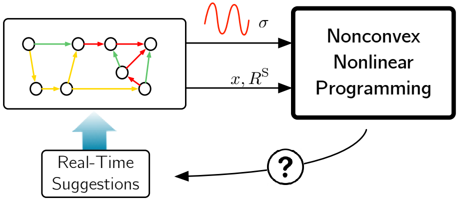


Optimization on discretized dynamics:

$$
\begin{aligned}
\min_{r^{\mathsf{c}}} \quad & \sum_{k=1}^{h} \phi(x_k) & \\
\text{s.t.} \quad & x_{k+1} = \mathcal{F}(x_k, r_k, \lambda_k), & k = 1, \dots, h, \\
& r_k = \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}}), & k = 1, \dots, h, \\
& \sum_i r_{ij} = b_j, & j = 1, \dots, n, \\
& 0 \leq r_{ij}^{\mathsf{c}} \leq 1, & (i,j) \in \mathcal{E}, \\
& x_k \leq B, & k = 1, \dots, h,
\end{aligned}
$$

Because of nonlinearity of $\mathcal{F}$: **nonlinear programming opt. problem**

Numerically Solving the Optimization



Optimization on discretized dynamics:

$$\min_{r^{\mathsf{c}}} \quad \sum_{k=1}^{h} \phi(x_k)$$

$$\begin{aligned}
\text{s.t.} \quad & x_{k+1} = \mathcal{F}(x_k, r_k, \lambda_k), && k = 1, \dots, h, \\
& r_k = \Psi(\sigma_k, r^{\mathsf{s}}, r^{\mathsf{c}}), && k = 1, \dots, h, \\
& \sum_i r_{ij} = b_j, && j = 1, \dots, n, \\
& 0 \leq r^{\mathsf{c}}_{ij} \leq 1, && (i,j) \in \mathcal{E}, \\
& x_k \leq B, && k = 1, \dots, h,
\end{aligned}$$

Because of nonlinearity of $\mathcal{F}$: **nonlinear programming opt. problem**

Solving Nonlinear Programming in Real Time

$$\boxed{\begin{aligned} \min_{r^{\mathsf{c}}} \quad & f_0(r^{\mathsf{c}}, \hat{x}, \sigma) \\ \text{s.t.} \quad & g_i(r^{\mathsf{c}}, \hat{x}, \sigma) \leq 0 \\ & h_j(r^{\mathsf{c}}, \hat{x}, \sigma) = 0 \end{aligned}}$$

1. Solve NLP problem offline with $\sigma = \sigma_0$
2. Compose the Lagrangian

$$\mathcal{L}(r^{\mathsf{c}}, \hat{x}, \sigma, w, u) = f_0(r^{\mathsf{c}}, \hat{x}, \sigma) + u^\top g(r^{\mathsf{c}}, \hat{x}, \sigma) + w^\top h(r^{\mathsf{c}}, \hat{x}, \sigma)$$

Solving Nonlinear Programming in Real Time

$$\begin{aligned} \min_{r^{\mathsf{c}}} \quad & f_0(r^{\mathsf{c}}, \hat{x}, \sigma) \\ \text{s.t.} \quad & g_i(r^{\mathsf{c}}, \hat{x}, \sigma) \leq 0 \\ & h_j(r^{\mathsf{c}}, \hat{x}, \sigma) = 0 \end{aligned}$$

1. Solve NLP problem offline with $\sigma = \sigma_0$
2. Compose the Lagrangian

$$\mathcal{L}(r^{\mathsf{c}}, \hat{x}, \sigma, w, u) = f_0(r^{\mathsf{c}}, \hat{x}, \sigma) + u^\top g(r^{\mathsf{c}}, \hat{x}, \sigma) + w^\top h(r^{\mathsf{c}}, \hat{x}, \sigma)$$

Solving Nonlinear Programming in Real Time

$$\min_{r^{\mathsf{c}}} \quad f_0(r^{\mathsf{c}}, \hat{x}, \sigma)$$
$$\text{s.t.} \quad g_i(r^{\mathsf{c}}, \hat{x}, \sigma) \leq 0$$
$$h_j(r^{\mathsf{c}}, \hat{x}, \sigma) = 0$$

1. Solve NLP problem offline with $\sigma = \sigma_0$
2. Compose the Lagrangian

$$\mathcal{L}(r^{\mathsf{c}}, \hat{x}, \sigma, w, u) = f_0(r^{\mathsf{c}}, \hat{x}, \sigma) + u^\top g(r^{\mathsf{c}}, \hat{x}, \sigma) + w^\top h(r^{\mathsf{c}}, \hat{x}, \sigma)$$

3. Set of KKT conditions

$$\nabla \mathcal{L}(r^{\mathsf{c}*}, \hat{x}^*, \sigma_0, w^*, u^*) = 0$$
$$u_i g_i(r^{\mathsf{c}*}, \hat{x}^*, \sigma_0) = 0$$
$$h_j(r^{\mathsf{c}*}, \hat{x}^*, \sigma_0) = 0$$

Implicit Equation

$$F(r^{\mathsf{c}*}, \hat{x}^*, \sigma_0, u^*, w^*) = 0,$$

Describes Relation Between Variables At Optimality

Solving Nonlinear Programming in Real Time (2)

$F(r^{c*}, \hat{x}^*, \sigma_0, u^*, w^*) = 0$

④ Let $y = [r^c(\sigma)\ u(\sigma)\ w(\sigma)]$ and derive w.r.t. σ

$$M(\sigma)\frac{dy}{d\sigma} + N(\sigma) = 0,$$

$M(\sigma) = [\partial F_i/\partial y_j]$, $dy/d\sigma = [dy_i/d\sigma_j]$, $N(\sigma) = [\partial F_i/\partial \sigma_j]$

④ If $r^{c*}(\sigma_0)$ is a local isolated minimizing point

Describes variation of optimal solution to changes in σ

$$\frac{dy}{d\sigma} = -M^{-1}(\sigma_0)N(\sigma_0)$$

⑤ Let $dy/d\sigma = [\eta_1,\ \eta_2,\ \eta_3]^\mathsf{T}$

$$r^c(\sigma) = r^{c*}(\sigma_0) + \eta_1(\sigma - \sigma_0)$$

Linear update: significantly faster than solving nonlinear programming

Solving Nonlinear Programming in Real Time (2)

$F(r^{\mathsf{c}*}, \hat{x}^*, \sigma_0, u^*, w^*) = 0$

4 Let $y = [r^{\mathsf{c}}(\sigma)\ u(\sigma)\ w(\sigma)]$ and derive w.r.t. σ

$$M(\sigma)\frac{dy}{d\sigma} + N(\sigma) = 0,$$

$M(\sigma) = [\partial F_i / \partial y_j]$, $\quad dy/d\sigma = [dy_i/d\sigma_j]$, $\quad N(\sigma) = [\partial F_i / \partial \sigma_j]$

4 If $r^{\mathsf{c}*}(\sigma_0)$ is a local isolated minimizing point

$$\frac{dy}{d\sigma} = -M^{-1}(\sigma_0)N(\sigma_0)$$

Describes variation of optimal solution to changes in σ

5 Let $dy/d\sigma = [\eta_1,\ \eta_2,\ \eta_3]^{\mathsf{T}}$

$$r^{\mathsf{c}}(\sigma) = r^{\mathsf{c}*}(\sigma_0) + \eta_1(\sigma - \sigma_0)$$

Linear update: significantly faster than solving nonlinear programming

Solving Nonlinear Programming in Real Time (2)

$F(r^{c*}, \hat{x}^*, \sigma_0, u^*, w^*) = 0$

4. Let $y = [r^{c}(\sigma)\ u(\sigma)\ w(\sigma)]$ and derive w.r.t. σ

$$M(\sigma)\frac{dy}{d\sigma} + N(\sigma) = 0,$$

$M(\sigma) = [\partial F_i/\partial y_j], \quad dy/d\sigma = [dy_i/d\sigma_j], \quad N(\sigma) = [\partial F_i/\partial \sigma_j]$

4. If $r^{c*}(\sigma_0)$ is a local isolated minimizing point

Describes variation of optimal solution to changes in σ

$$\frac{dy}{d\sigma} = -M^{-1}(\sigma_0)N(\sigma_0)$$

5. Let $dy/d\sigma = [\eta_1,\ \eta_2,\ \eta_3]^{\mathsf{T}}$

$$r^{c}(\sigma) = r^{c*}(\sigma_0) + \eta_1(\sigma - \sigma_0)$$

Linear update: significantly faster than solving nonlinear programming

Fast Update Error Bound

Theorem

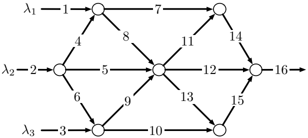
Assume

1. All constraints are linearly independent
2. Second order KKT conditions hold at optimizer
3. Strict complementary slackness

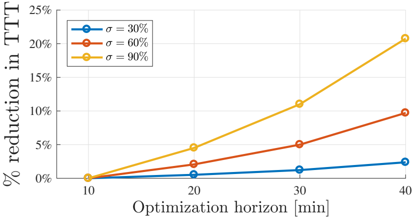
Then,

$$r^{\mathsf{c}*}(\sigma) = r^{\mathsf{c}*}(\sigma_0) + \eta_1(\sigma - \sigma_0) + o(\|\sigma - \sigma_0\|^2)$$

Numerical Results

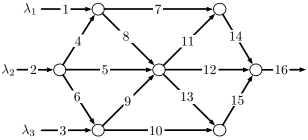


- Network: $n = 15$ roads, $m = 7$ junctions
- Roads: $L_i = 5\text{mi}$, capacity $B_i = 200\text{veh}$
- Routing: r_{ij}^{s} split uniformly

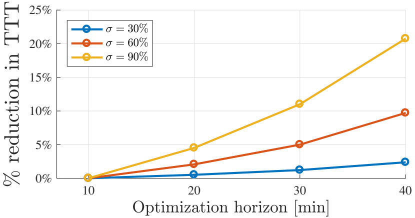


Consistent reduction in travel time not only requires significant penetration levels, but also appropriately large control horizons

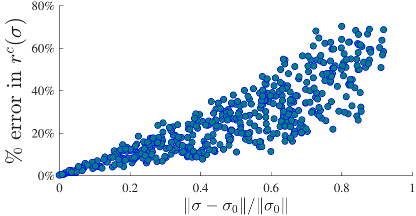
Numerical Results



- Network: $n = 15$ roads, $m = 7$ junctions
- Roads: $L_i = 5$mi, capacity $B_i = 200$veh
- Routing: r_{ij}^{s} split uniformly



Consistent reduction in travel time not only requires significant penetration levels, but also appropriately large control horizons



Error with real-time update = $o(\|\sigma - \sigma_0\|^2)$

Network Resilience

Resilience of a Link

The Margin of Resilience of link i is

Smallest change in σ that generates its failure due to a jam

$$\begin{aligned} \rho_i(x_0) := \min_{\sigma} \quad & \|\sigma - \sigma_0\|_1 \\ \text{s.t.} \quad & \dot{x} = (R - I) f(x, t) + \lambda \\ & R = \Sigma R^{\mathsf{C}} + (I - \Sigma) R^{\mathsf{S}} \\ & x_i \geq B_i, \ \text{for some } t \in [0, \mathcal{H}] \end{aligned}$$

Estimating the Margin of Resilience

1. Constraint violation is $\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) \geq B_i$
2. For small changes of σ, take Taylor expansion

$$\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) = \mathcal{F}_i(x_k, r_k(\sigma_0), \lambda_k) + \underbrace{\frac{d\mathcal{F}_i}{d\sigma}(x_k, r_k(\sigma), \lambda_k)\Big|_{\sigma=\sigma_0}}_{\Psi_i(r_k, x_k, \lambda_k, \sigma)}(\sigma - \sigma_0) + o(\|\delta_\sigma\|^2) \geq B_i$$

3. By rearranging the terms and by taking the norms

$$\rho_i(\mathcal{G}, x_0) \geq \min_k \frac{B_i - \mathcal{F}_i(x_k, r_k, \lambda_k)}{\|\Psi_i(k, \lambda, \sigma_0)\|_\infty}$$

Lower Bound on Margin of Resilience

(Can be quickly computed from real-time update)

Estimating the Margin of Resilience

1. Constraint violation is $\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) \geq B_i$
2. For small changes of σ, take Taylor expansion

$$\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) = \mathcal{F}_i(x_k, r_k(\sigma_0), \lambda_k) + \underbrace{\frac{d\mathcal{F}_i}{d\sigma}(x_k, r_k(\sigma), \lambda_k)\Big|_{\sigma=\sigma_0}}_{\Psi_i(r_k, x_k, \lambda_k, \sigma)} (\sigma - \sigma_0) + o(\|\delta_\sigma\|^2) \geq B_i$$

3. By rearranging the terms and by taking the norms

$$\rho_i(\mathcal{G}, x_0) \geq \min_k \frac{B_i - \mathcal{F}_i(x_k, r_k, \lambda_k)}{\|\Psi_i(k, \lambda, \sigma_0)\|_\infty}$$

Lower Bound on Margin of Resilience

(Can be quickly computed from real-time update)

Estimating the Margin of Resilience

1. Constraint violation is $\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) \geq B_i$
2. For small changes of σ, take Taylor expansion

$$\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) = \mathcal{F}_i(x_k, r_k(\sigma_0), \lambda_k) + \underbrace{\frac{d\mathcal{F}_i}{d\sigma}(x_k, r_k(\sigma), \lambda_k)\bigg|_{\sigma=\sigma_0}}_{\Psi_i(r_k, x_k, \lambda_k, \sigma)} (\sigma - \sigma_0) + o(\|\delta_\sigma\|^2) \geq B_i$$

3. By rearranging the terms and by taking the norms

$$\rho_i(\mathcal{G}, x_0) \geq \min_k \frac{B_i - \mathcal{F}_i(x_k, r_k, \lambda_k)}{\|\Psi_i(k, \lambda, \sigma_0)\|_\infty}$$

Lower Bound on Margin of Resilience

(Can be quickly computed from real-time update)

Estimating the Margin of Resilience

1. Constraint violation is $\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) \geq B_i$
2. For small changes of σ, take Taylor expansion

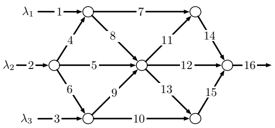
$$\mathcal{F}_i(x_k, r_k(\sigma), \lambda_k) = \mathcal{F}_i(x_k, r_k(\sigma_0), \lambda_k) + \underbrace{\frac{d\mathcal{F}_i}{d\sigma}(x_k, r_k(\sigma), \lambda_k)\bigg|_{\sigma=\sigma_0}}_{\Psi_i(r_k, x_k, \lambda_k, \sigma)} (\sigma - \sigma_0) + o(\|\delta_\sigma\|^2) \geq B_i$$

3. By rearranging the terms and by taking the norms

$$\rho_i(\mathcal{G}, x_0) \geq \min_k \frac{B_i - \mathcal{F}_i(x_k, r_k, \lambda_k)}{\|\Psi_i(k, \lambda, \sigma_0)\|_\infty}$$

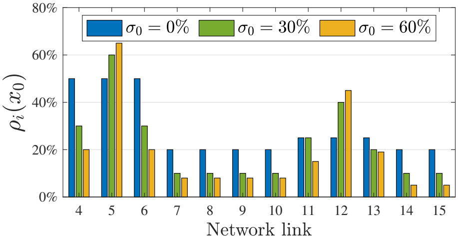
Lower Bound on Margin of Resilience

(Can be quickly computed from real-time update)



- Network: $n = 15$ roads, $m = 7$ junctions
- Roads: $L_i = 5$mi, capacity $B_i = 200$veh
- Routing: r_{ij}^{s} split uniformly

Lower bound on resilience:



Networks where the routing is partially controlled by a system planner are more prone to traffic jam phenomena originated by changes in fluctuations in routing choices

Conclusions

Contribution:

- Real-time mechanism based on first-order sensitivity analysis for NLP
- Technique to estimate margin of resilience

Outcomes:

- (Expected) Routing control leads to improved network performance
- (Counterintuitive) Performance comes at the cost of higher fragility

Directions:

✓ Need of a framework that allows us to formalize these observations

✎ Increasing need to capture human responses to control policies

Thanks: Control of Complex Systems Initiative (CCSI) at PNNL & NSF

Resilience of Traffic Networks with Partially Controlled Routing

Gianluca Bianchin
in collaboration with: Fabio Pasqualetti and Soumya Kundu

Department of Mechanical Engineering
University of California, Riverside

Optimization and Control Group
Pacific Northwest National Laboratory

2019 American Control Conference
July 11, 2019 | Philadelphia, USA