FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Cellpath: state estimation of traffic networks via convex optimization

Cathy Wu[1], Jérôme Thai[1], Steve Yadlowsky[1], Alexei Pozdnoukhov[2], Alexandre Bayen[1,2,3]

[1]Department of Electrical Engineering & Computer Sciences, UC Berkeley
[2]Department of Civil and Environmental Engineering, UC Berkeley
[3]Institute for Transportation Studies (ITS), UC Berkeley

Berkeley UNIVERSITY OF CALIFORNIA   Massachusetts Institute of Technology   NSF   MICHIGAN   VANDERBILT UNIVERSITY

Outline

Route flow estimation problem and formulation

Block simplex constrained quadratic program: analysis and algorithms

Experiments and conclusions

Outline

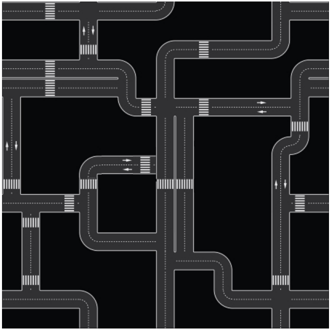
Route flow estimation problem and formulation

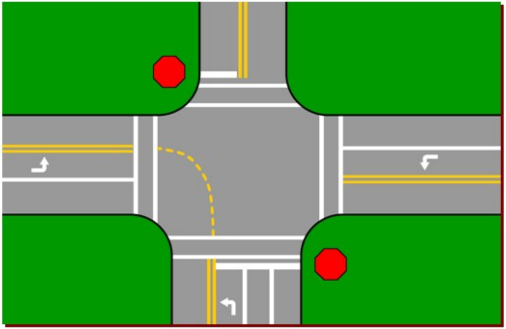
Block simplex constrained quadratic program: analysis and algorithms

Experiments and conclusions

The root of all traffic evils

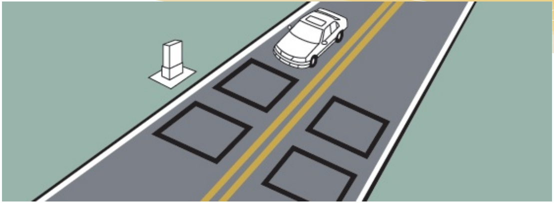
We have little information of what's going on in the road network.





Information

Traditional approaches to traffic *flow* estimation

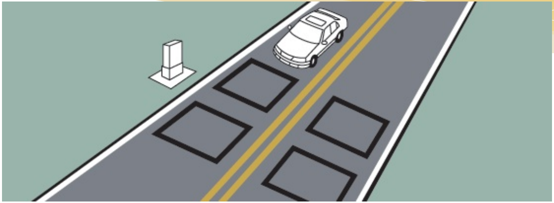


Sparse static sensors

+

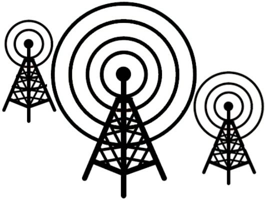
Equilibrium models

This talk: *data-driven* estimation of *route* flow



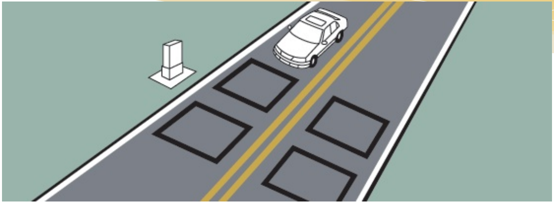
Sparse static sensors $\longrightarrow$ observed link flow

$+$



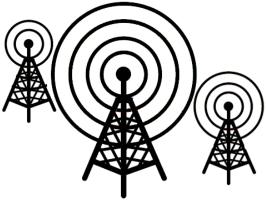
Cellular network sensors $\longrightarrow$ cellpath flow

This talk: *data-driven* estimation of *route* flow



Sparse static sensors $\rightarrow$ observed link flow

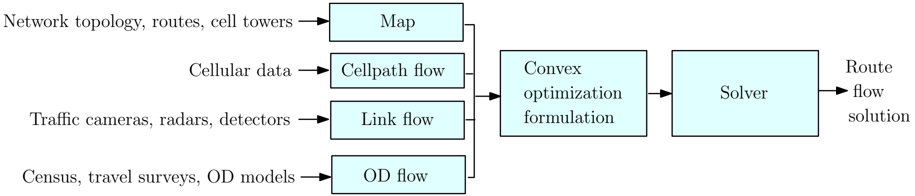
$+$



Cellular network sensors $\rightarrow$ cellpath flow

Our contributions

- Convex optimization formulation for route flow estimation problem[1,2]
- Formalize cellular network data as *cellpaths*[1]
- Formulation compatible with equilibrium concepts and other data[1]
- Projected gradient method with $O(n)$ projection step[1,3]
- 90% accuracy in numerical experiments on large-scale networks[1]
- Proposed framework: full pipeline for traffic estimation[1]



[1] C Wu, J Thai, S Yadlowsky, A Pozdnoukhov, A Bayen. "Cellpath: fusion of cellular and traffic sensor data for route flow estimation via convex optimization." *Transportation Research Part C: Emerging Technologies* (2015).

[2] C Wu, A Pozdnoukhov, A Bayen. "Block simplex signal recovery: a method comparison and an application to routing." *In review, ACM TIST*.

[3] J Thai, C Wu, A Pozdnoukhov, A Bayen. "Convex programming on the l1-ball and on the simplex via isotonic regression." *CDC* (2015).

Problem statement: route flow estimation

Route flow estimation problem

Given

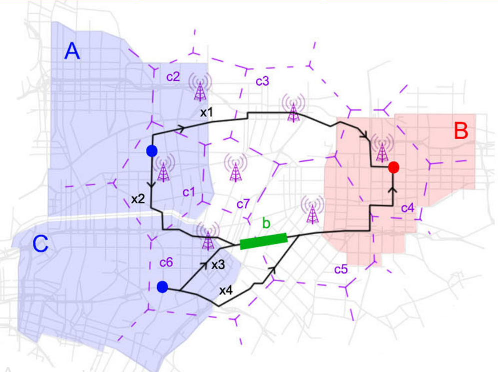
- Road network, origins, cells
- Top routes between OD pairs
- Cellpath flows, f
- OD flows, d
- Observed link flows, b

Recover

- Flow along routes, x

Cellpath flow

Flow along a sequence of cells



Assumptions

- Static, noiseless
- Cell partitioning = Voronoi
- Cellpaths contiguous
- Cellpaths well-posed

Problem statement: route flow estimation

Route flow estimation problem

Given

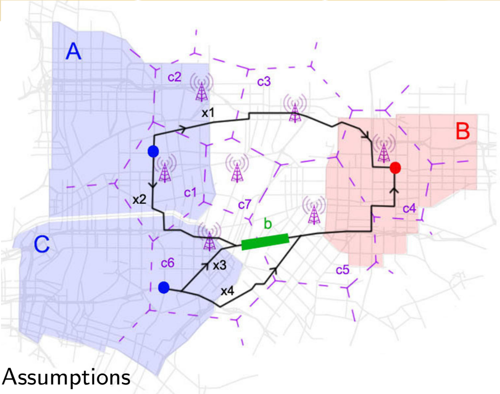
- Road network, origins, cells
- Top routes between OD pairs
- Cellpath flows, f
- OD flows, d
- Observed link flows, b

Recover

- Flow along routes, x

Cellpath flow

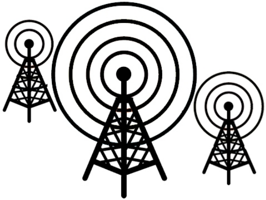
Flow along a sequence of cells



Assumptions

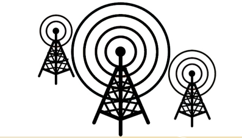
- Static, noiseless
- Cell partitioning = Voronoi
- Cellpaths contiguous
- Cellpaths well-posed

A brief note on cellular networks



- Not GPS (read-only signals)
- Cell towers spaced $\frac{1}{4} - \frac{1}{2}$ mi (urban areas) to 1-2 mi apart (suburbia)
- Cell towers collect signals as devices connect to and use the network

Related works

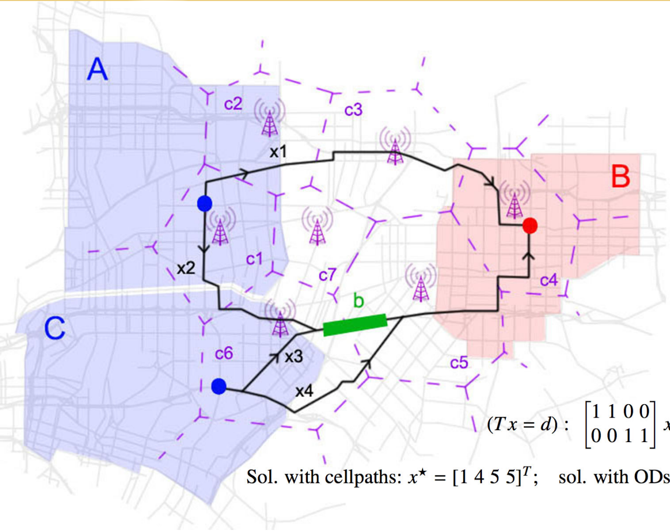


Cellular location data + transportation problems

- Excellent survey [GK09]
- OD matrix estimation (ODME) [CWB07, BHNF08, CLDLR11]
- Travel time estimation [THF06, FJS07, B07, JHVRH12]
- Congestion detection and classification [JHVRH12]
- Link density estimation [YTWPB14]
- Route choice modeling [TDV12]

This work: cellular location data + route flow estimation

Example problem setup



All flows are in 1000 vehicles/hour.

cellpath flows: $\begin{cases} f_{p1234} = 1 = x_1 \\ f_{p1654} = 4 = x_2 \\ f_{p654} = 10 = x_3 + x_4 \end{cases}$

OD demands: $\begin{cases} d_{AB} = 5 = x_1 + x_2 \\ d_{CB} = 10 = x_3 + x_4 \end{cases}$

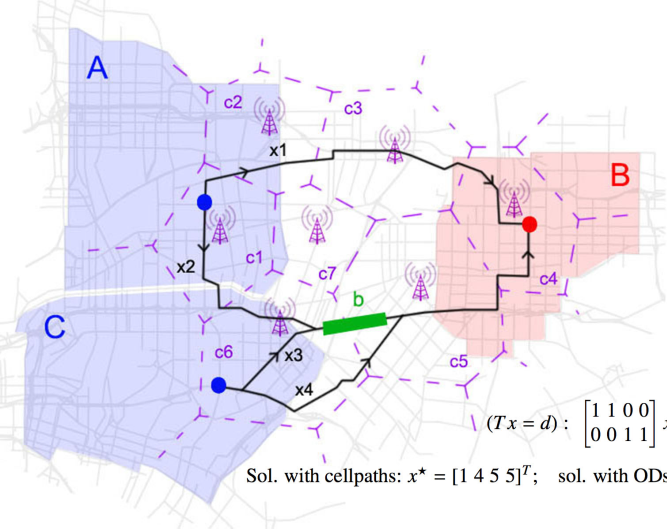
Link flow: $b = 9 = x_2 + x_3$

$$(Ux = f): \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} f_{p1234} \\ f_{p1654} \\ f_{p654} \end{bmatrix}$$

$$(Tx = d): \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} d_{AB} \\ d_{CB} \end{bmatrix} \quad ; \quad (Ax = b): \ [0\ 1\ 1\ 0]\, x = b$$

Sol. with cellpaths: $x^\star = [1\ 4\ 5\ 5]^T$; sol. with ODs: $x = x^\star + \left[1\ -1\ 1\ -1\right]^T t, \ \forall t \in [-1, 4]$

Example problem setup



All flows are in 1000 vehicles/hour.

cellpath flows: $\begin{cases} f_{p1234} = 1 = x_1 \\ f_{p1654} = 4 = x_2 \\ f_{p654} = 10 = x_3 + x_4 \end{cases}$

OD demands: $\begin{cases} d_{AB} = 5 = x_1 + x_2 \\ d_{CB} = 10 = x_3 + x_4 \end{cases}$

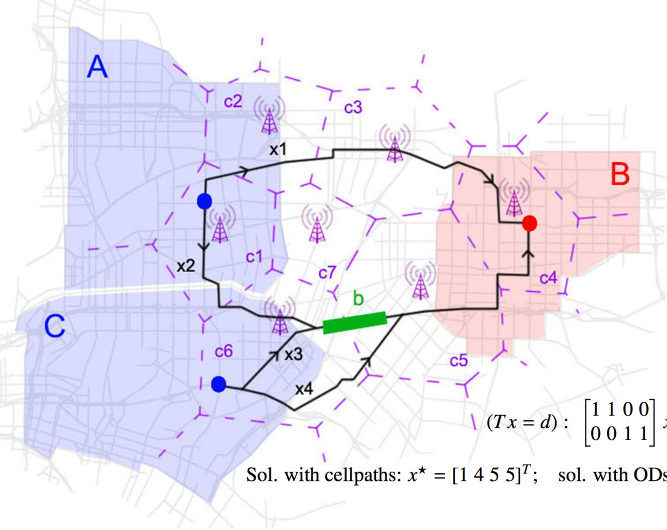
Link flow: $b = 9 = x_2 + x_3$

$$(Ux = f) : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} f_{p1234} \\ f_{p1654} \\ f_{p654} \end{bmatrix}$$

$$(Tx = d) : \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} d_{AB} \\ d_{CB} \end{bmatrix} \; ; \quad (Ax = b) : \; [0\ 1\ 1\ 0]\, x = b$$

Sol. with cellpaths: $x^\star = [1\ 4\ 5\ 5]^T$; sol. with ODs: $x = x^\star + \left[1\ -1\ 1\ -1\right]^T t, \ \forall t \in [-1, 4]$

Example problem setup



All flows are in 1000 vehicles/hour.

cellpath flows: $\begin{cases} f_{p1234} = 1 \;\; = x_1 \\ f_{p1654} = 4 \;\; = x_2 \\ f_{p654} \;\, = 10 = x_3 + x_4 \end{cases}$

OD demands: $\begin{cases} d_{AB} = 5 \;\; = x_1 + x_2 \\ d_{CB} = 10 = x_3 + x_4 \end{cases}$

Link flow: $b = 9 = x_2 + x_3$

$$(Ux = f): \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} f_{p1234} \\ f_{p1654} \\ f_{p654} \end{bmatrix}$$

$$(Tx = d): \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} d_{AB} \\ d_{CB} \end{bmatrix} \; ; \quad (Ax = b): \; [0\;1\;1\;0]\, x = b$$

Sol. with cellpaths: $x^\star = [1\;4\;5\;5]^T$; sol. with ODs: $x = x^\star + \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}^T t, \; \forall t \in [-1, 4]$

Convex optimization formulation

Block simplex constrained quadratic program (QP):

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} & Ux = f,\ x \geq 0 \end{array}$$

- link-route: $A_{lr} = \begin{cases} 1 & \text{if } l \in r \\ 0 & \text{else} \end{cases}$; cellpath-route: $U_{pr} = \begin{cases} 1 & \text{if } r \in \mathcal{R}^p \\ 0 & \text{else} \end{cases}$
- $b \in \mathbb{R}_+^{|\mathcal{L}|}$ observed link flow vector, $b = (b_l)_{l \in \mathcal{L}}$
- $f \in \mathbb{R}_+^{|\mathcal{P}|}$ cellpath flow vector $f = (f_p)_{p \in \mathcal{P}}$
- $x \in \mathbb{R}_+^{|\mathcal{R}|}$ route flow vector, $x = (x_r)_{r \in \mathcal{R}}$

Convex optimization formulation

Block simplex constrained quadratic program (QP):

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} & Ux = f,\ x \geq 0 \end{array}$$

- link-route: $A_{lr} = \begin{cases} 1 & \text{if } l \in r \\ 0 & \text{else} \end{cases}$; cellpath-route: $U_{pr} = \begin{cases} 1 & \text{if } r \in \mathcal{R}^p \\ 0 & \text{else} \end{cases}$
- $b \in \mathbb{R}_+^{|\mathcal{L}|}$ observed link flow vector, $b = (b_l)_{l \in \mathcal{L}}$
- $f \in \mathbb{R}_+^{|\mathcal{P}|}$ cellpath flow vector $f = (f_p)_{p \in \mathcal{P}}$
- $x \in \mathbb{R}_+^{|\mathcal{R}|}$ route flow vector, $x = (x_r)_{r \in \mathcal{R}}$

Convex optimization formulation

Block simplex constrained quadratic program (QP):

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} \quad & Ux = f,\; x \geq 0 \end{aligned}$$

- n routes
- m link flow measurements
- q cellpath flow constraints
- Separable scaled simplex constraint
- Weakly convex, underdetermined, $rk(A) \leq m \ll n, q \leq n$

Flexibility of our formulation

Cellpath + observed link flows

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} \quad & Ux = f,\ x \geq 0 \end{aligned}$$

OD + observed link flows

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} \quad & Tx = d,\ x \geq 0 \end{aligned}$$

Cellpath + OD + observed link flows

$$\begin{aligned} \min \quad & \tfrac{1}{2}\left\|\begin{bmatrix} A \\ T \end{bmatrix} x - \begin{bmatrix} b \\ d \end{bmatrix}\right\|_2^2 \\ \text{s.t.} \quad & Ux = f,\ x \geq 0 \end{aligned}$$

And more: turning ratio data and traffic assignment problem

Flexibility of our formulation

Cellpath + observed link flows

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} & Ux = f,\ x \geq 0 \end{array}$$

OD + observed link flows

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} & Tx = d,\ x \geq 0 \end{array}$$

Cellpath + OD + observed link flows

$$\begin{array}{ll} \min & \frac{1}{2}\left\| \begin{bmatrix} A \\ T \end{bmatrix} x - \begin{bmatrix} b \\ d \end{bmatrix} \right\|_2^2 \\ \text{s.t.} & Ux = f,\ x \geq 0 \end{array}$$

And more: turning ratio data and traffic assignment problem

Flexibility of our formulation

Cellpath + observed link flows

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} \quad & Ux = f,\ x \geq 0 \end{aligned}$$

OD + observed link flows

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} \quad & Tx = d,\ x \geq 0 \end{aligned}$$

Cellpath + OD + observed link flows

$$\begin{aligned} \min \quad & \tfrac{1}{2}\left\| \begin{bmatrix} A \\ T \end{bmatrix} x - \begin{bmatrix} b \\ d \end{bmatrix} \right\|_2^2 \\ \text{s.t.} \quad & Ux = f,\ x \geq 0 \end{aligned}$$

And more: turning ratio data and traffic assignment problem

Outline

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.

while stopping criteria not met **do**
  Determine a descent direction Δx
  Step in that direction: $x^+ := x + \alpha \Delta x$
  Projection: $x := \Pi_{\mathcal{X}}(x^+)$
end while
return x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.

```
while stopping criteria not met do
    Determine a descent direction Δx
    Step in that direction: x⁺ := x + αΔx
    Projection: x := Π_X(x⁺)
end while
return x
```

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.
1: **while** stopping criteria not met **do**
2: Determine a descent direction Δx
3: Step in that direction: $x^+ := x + \alpha\Delta x$
4: Projection: $x := \Pi_{\mathcal{X}}(x^+)$
5: **end while**
6: **return** x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.
1: **while** stopping criteria not met **do**
2: Determine a descent direction Δx
3: Step in that direction: $x^+ := x + \alpha \Delta x$
4: Projection: $x := \Pi_{\mathcal{X}}(x^+)$
5: **end while**
6: **return** x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.

while stopping criteria not met **do**
  Determine a descent direction Δx
  Step in that direction: $x^+ := x + \alpha \Delta x$
  Projection: $x := \Pi_{\mathcal{X}}(x^+)$
end while
return x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.
1: **while** stopping criteria not met **do**
2: Determine a descent direction Δx
3: Step in that direction: $x^+ := x + \alpha \Delta x$
4: Projection: $x := \Pi_{\mathcal{X}}(x^+)$
5: **end while**
6: **return** x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$

Require: initial point x in the feasible set $\mathcal{X}$.

while stopping criteria not met **do**
    Determine a descent direction Δx
    Step in that direction: $x^+ := x + \alpha \Delta x$
    Projection: $x := \Pi_{\mathcal{X}}(x^+)$
end while
return x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.

while stopping criteria not met **do**
    Determine a descent direction Δx
    Step in that direction: $x^+ := x + \alpha \Delta x$
    Projection: $x := \Pi_{\mathcal{X}}(x^+)$
end while
return x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.
while stopping criteria not met **do**
  Determine a descent direction Δx
  Step in that direction: $x^+ := x + \alpha \Delta x$
  Projection: $x := \Pi_{\mathcal{X}}(x^+)$
end while
return x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Summary of analysis: block simplex quadratic program

Goal: design an efficient first-order projected descent method

General projected descent method

Algorithm 1 `Proj-descent(·)`

Require: minimizing function $f(x)$
Require: initial point x in the feasible set $\mathcal{X}$.
while stopping criteria not met **do**
    Determine a descent direction Δx
    Step in that direction: $x^+ := x + \alpha \Delta x$
    Projection: $x := \Pi_{\mathcal{X}}(x^+)$
end while
return x

1. Equivalence to separable **standard** simplex constraints
2. Transformable to order constraints via equality constraint elimination
3. Design an efficient projection under a coordinate change

(1/3) Block standard simplex

Proposition: Equivalence of scaled and standard simplices

For C a well-posed block simplex matrix:

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} & Cx = \mathbb{1},\ x \geq 0 \end{array} \iff \begin{array}{ll} \min & \frac{1}{2}\|\tilde{A}\tilde{x} - b\|_2^2 \\ \text{s.t.} & C\tilde{x} = d,\ \tilde{x} \geq 0 \end{array}$$

where

$$E := \text{diag}(C^T d) \quad A := \tilde{A}E \quad x := E^{-1}\tilde{x}$$

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{s.t.} & Ux = \mathbb{1},\ x \geq 0 \end{array} \iff \begin{array}{ll} \min & \frac{1}{2}\|\tilde{A}\tilde{x} - b\|_2^2 \\ \text{s.t.} & U\tilde{x} = f,\ \tilde{x} \geq 0 \end{array}$$

New interpretation: $\tilde{x}$ is route flow vector $\to$ x is route split vector

$$A \in \mathbb{R}_+^{|\mathcal{L}|\times|\mathcal{R}|} \ : \ A_{lr} = \begin{cases} f_p & \text{if } l \in r \in \mathcal{R}^p \\ 0 & \text{else} \end{cases}$$

(1/3) Block standard simplex

Proposition: Equivalence of scaled and standard simplices

For C a well-posed block simplex matrix:

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax-b\|_2^2 \\ \text{s.t.} & Cx = \mathbb{1},\ x \geq 0 \end{array} \quad \Longleftrightarrow \quad \begin{array}{ll} \min & \frac{1}{2}\|\tilde{A}\tilde{x}-b\|_2^2 \\ \text{s.t.} & C\tilde{x} = d,\ \tilde{x} \geq 0 \end{array}$$

where

$$E := \mathrm{diag}(C^T d) \quad A := \tilde{A}E \quad x := E^{-1}\tilde{x}$$

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax-b\|_2^2 \\ \text{s.t.} & Ux = \mathbb{1},\ x \geq 0 \end{array} \quad \Longleftrightarrow \quad \begin{array}{ll} \min & \frac{1}{2}\|\tilde{A}\tilde{x}-b\|_2^2 \\ \text{s.t.} & U\tilde{x} = f,\ \tilde{x} \geq 0 \end{array}$$

New interpretation: $\tilde{x}$ is route flow vector $\rightarrow$ x is route split vector

$$A \in \mathbb{R}_+^{|\mathcal{L}|\times|\mathcal{R}|} \ : \ A_{lr} = \begin{cases} f_p & \text{if } l \in r \in \mathcal{R}^p \\ 0 & \text{else} \end{cases}$$

(1/3) Block standard simplex

Proposition: Equivalence of scaled and standard simplices

For C a well-posed block simplex matrix:

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax-b\|_2^2 \\ \text{s.t.} & Cx = \mathbb{1},\ x \geq 0 \end{array} \iff \begin{array}{ll} \min & \frac{1}{2}\|\tilde{A}\tilde{x}-b\|_2^2 \\ \text{s.t.} & C\tilde{x} = d,\ \tilde{x} \geq 0 \end{array}$$

where

$$E := \text{diag}(C^T d) \quad A := \tilde{A}E \quad x := E^{-1}\tilde{x}$$

$$\begin{array}{ll} \min & \frac{1}{2}\|Ax-b\|_2^2 \\ \text{s.t.} & Ux = \mathbb{1},\ x \geq 0 \end{array} \iff \begin{array}{ll} \min & \frac{1}{2}\|\tilde{A}\tilde{x}-b\|_2^2 \\ \text{s.t.} & U\tilde{x} = f,\ \tilde{x} \geq 0 \end{array}$$

New interpretation: $\tilde{x}$ is route flow vector $\rightarrow$ x is route split vector

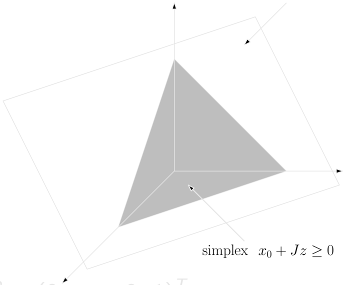
$$A \in \mathbb{R}_+^{|\mathcal{L}|\times|\mathcal{R}|} \ : \ A_{lr} = \begin{cases} f_p & \text{if } l \in r \in \mathcal{R}^p \\ 0 & \text{else} \end{cases}$$

(2/3) Transforming standard simplex into ordering constraint

Proposition: Ordering constraint

$$\mathbf{1}^T x^p = 1, \quad x^p \geq 0 \quad \Longleftrightarrow \quad 0 \leq z_1^p \leq \cdots \leq z_{n_p-1}^p \leq 1$$

for an appropriate change of variables $x \to z$.



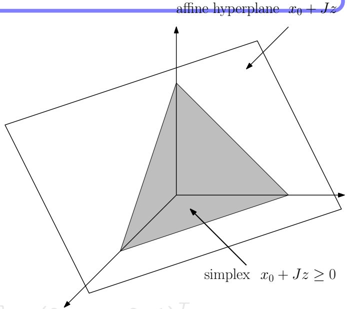
Constraint elimination trick: $x^p = x_0^p + J^p z^p$

Choose $J^p = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & \ddots & \\ & & \ddots & \end{pmatrix} \in \mathbb{R}^{n_p \times (n_p - 1)}$ and $x_0^p = (0, \cdots, 0, 1)^T$

Proposition: Ordering constraint

$$\mathbf{1}^T x^p = 1, \quad x^p \geq 0 \quad \Longleftrightarrow \quad 0 \leq z_1^p \leq \cdots \leq z_{n_p-1}^p \leq 1$$

for an appropriate change of variables $x \to z$.



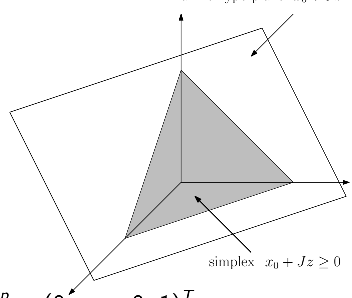
Constraint elimination trick: $x^p = x_0^P + J^p z^p$

Choose $J^p = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & \ddots & \\ & & \ddots & \end{pmatrix} \in \mathbb{R}^{n_p \times (n_p - 1)}$ and $x_0^p = (0, \cdots, 0, 1)^T$

Proposition: Ordering constraint

$$\mathbf{1}^T x^p = 1, \quad x^p \geq 0 \quad \Longleftrightarrow \quad 0 \leq z_1^p \leq \cdots \leq z_{n_p-1}^p \leq 1$$

for an appropriate change of variables $x \to z$.



Constraint elimination trick: $x^p = x_0^P + J^p z^p$

Choose $J^p = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & \ddots & \\ & & \ddots & \end{pmatrix} \in \mathbb{R}^{n_p \times (n_p - 1)}$ and $x_0^p = (0, \cdots, 0, 1)^T$

(3/3) Isotonic regression

Related work

Isotonic regression with complete order (ABERS55, BC90, LHM09)

$$\begin{array}{rl} \text{minimize} & \sum_{i=1}^{n} w_i(y_i - x_i)^2 \\ \text{subject to} & x_1 \leq x_2 \leq \cdots \leq x_n \end{array}$$

Solvable via pool adjacent violators (PAV) algorithm in $O(n)$ (BB72, GW84)

Our work: $\Pi_\Delta(y)$

Box-constrained isotonic regression with complete order

$$\begin{array}{rl} \text{minimize} & \sum_{i=1}^{n} w_i(y_i - x_i)^2 \\ \text{subject to} & t \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq u \end{array}$$

Solvable via PAV algorithm, then projection onto $[t, u]^n$, also $O(n)$

Previous work

Direct simplex projection in $\mathrm{O}(n \log n)$ (DGK08, WC13)

(3/3) Isotonic regression

Related work

Isotonic regression with complete order (ABERS55, BC90, LHM09)

$$\begin{array}{rl} \text{minimize} & \sum_{i=1}^{n} w_i (y_i - x_i)^2 \\ \text{subject to} & x_1 \leq x_2 \leq \cdots \leq x_n \end{array}$$

Solvable via pool adjacent violators (PAV) algorithm in $O(n)$ (BB72, GW84)

Our work: $\Pi_\Delta(y)$

Box-constrained isotonic regression with complete order

$$\begin{array}{rl} \text{minimize} & \sum_{i=1}^{n} w_i (y_i - x_i)^2 \\ \text{subject to} & t \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq u \end{array}$$

Solvable via PAV algorithm, then projection onto $[t, u]^n$, also $O(n)$

Previous work

Direct simplex projection in $O(n \log n)$ (DGK08, WC13)

(3/3) Isotonic regression

Related work

Isotonic regression with complete order (ABERS55, BC90, LHM09)

$$\begin{array}{rl} \text{minimize} & \sum_{i=1}^{n} w_i(y_i - x_i)^2 \\ \text{subject to} & x_1 \leq x_2 \leq \cdots \leq x_n \end{array}$$

Solvable via pool adjacent violators (PAV) algorithm in $O(n)$ (BB72, GW84)

Our work: $\Pi_\Delta(y)$

Box-constrained isotonic regression with complete order

$$\begin{array}{rl} \text{minimize} & \sum_{i=1}^{n} w_i(y_i - x_i)^2 \\ \text{subject to} & t \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq u \end{array}$$

Solvable via PAV algorithm, then projection onto $[t, u]^n$, also $O(n)$

Previous work

Direct simplex projection in $\mathrm{O}\left(n \log n\right)$ (DGK08, WC13)

(3/3) Box-constrained isotonic regression – proof summary

Proposition: Optimal solution via Euclidean projection

Solution x^* to BCIR(t, u) is the Euclidean projection of the solution x^{iso} to IR onto $[t, u]^n$.

IR: isotonic regression
BCIR: block-constrained isotonic regression

(3/3) Box-constrained isotonic regression – proof summary

Proposition: Optimal solution via Euclidean projection

Solution x^* to BCIR(t, u) is the Euclidean projection of the solution x^{iso} to IR onto $[t, u]^n$.

Lemma: Independent subproblems

Given a solution x^{iso} to IR, if there exists k s.t. $x_k^{\text{iso}} < x_{k+1}^{\text{iso}}$,
then IR reduces to two independent subproblems $\text{IR}_{1:k}$ and $\text{IR}_{k+1:n}$.

IR: isotonic regression
BCIR: block-constrained isotonic regression

(3/3) Box-constrained isotonic regression – proof summary

Proposition: Optimal solution via Euclidean projection

Solution x^* to BCIR(t, u) is the Euclidean projection of the solution x^{iso} to IR onto $[t, u]^n$.

Lemma: Independent subproblems

Given a solution x^{iso} to IR, if there exists k s.t. $x_k^{\text{iso}} < x_{k+1}^{\text{iso}}$,
then IR reduces to two independent subproblems $\text{IR}_{1:k}$ and $\text{IR}_{k+1:n}$.

Lemma: Uniformly lower bound solution

Given a solution x^{iso} to IR, if $x_i^{\text{iso}} \leq t \, \forall i$, then $x_i^* = t \, \forall i$ for BCIR(t, u).

IR: isotonic regression
BCIR: block-constrained isotonic regression

Outline

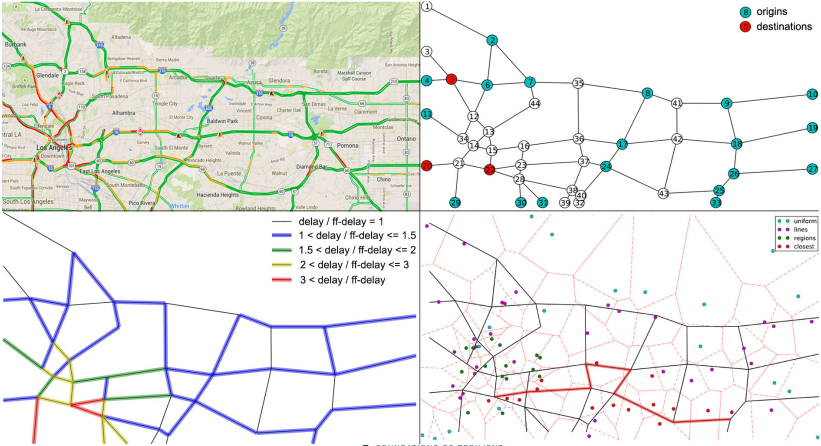
Route flow estimation problem and formulation

Block simplex constrained quadratic program: analysis and algorithms

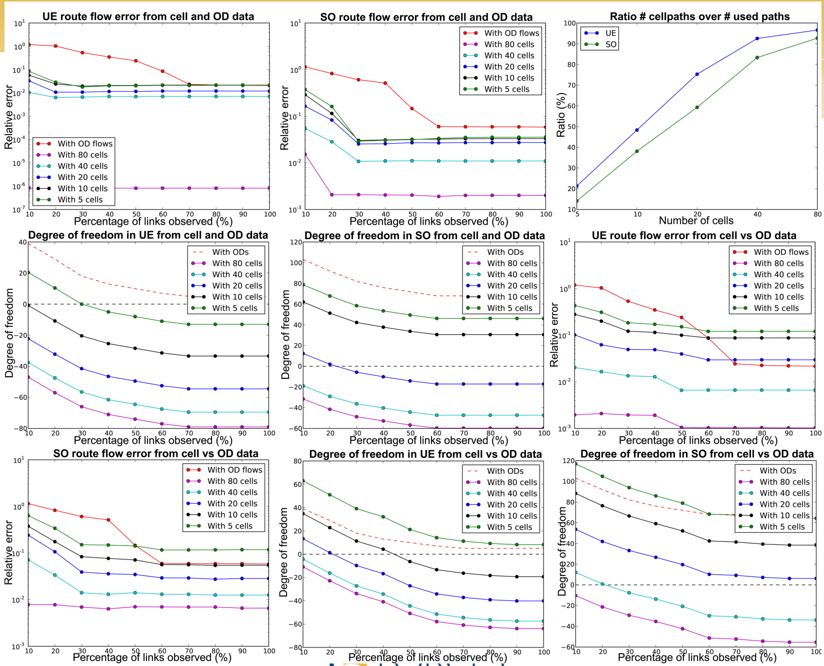
Experiments and conclusions

Experiment setup: Los Angeles highway network

- Network: 144 links, 44 nodes
- ODs: 21 origins, 3 destinations $\Longrightarrow$ 42 OD pairs
- Routes: 275 (UE), 300 (SO)
- Route flow: 91 (UE), 153 (SO) positive flows
- Link sensors: 5-100% static sensor coverage (most congested)
- Cellpath sensors (cells): $\{N^B, N^S, N^L\} \propto \{20, 40, 20\}$ (5-80 total)



Results: Los Angeles highway network



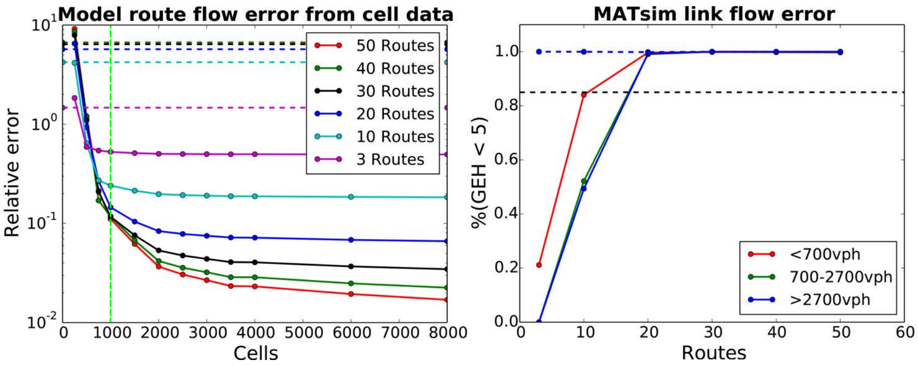
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Experiment: Los Angeles full network → **90% accuracy**

- Network: 20K links, 11K nodes
- 296K routes (up to 50 routes per OD pair)
- 1K observed links (5% coverage)
- 1K cells ⟹ 203K cellpaths
- 500K agents, trajectories simulated via MATSIM
- 32K origin-destination (OD) pairs; 321 ODs

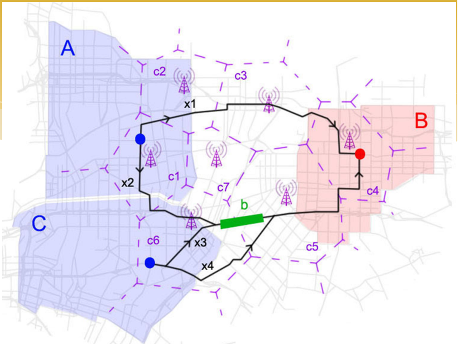
Cellpath + OD + observed link flows + model error

$$\begin{aligned} \min \quad & \frac{1}{2} \left\| \begin{bmatrix} A \\ T \end{bmatrix} x - \begin{bmatrix} b^{(50)} \\ d \end{bmatrix} \right\|_2^2 + \lambda \|x\|_2^2 \\ \text{s.t.} \quad & Ux = f, \; x \geq 0 \end{aligned}$$

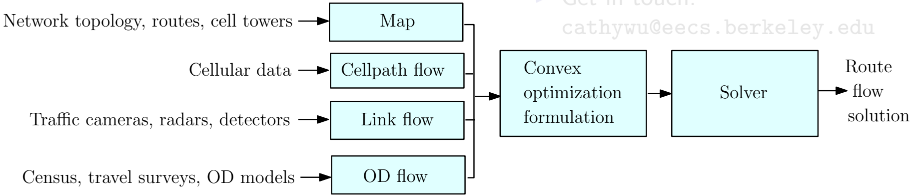


Conclusions

- Cellpath: fusion of cellular and traffic sensor data for route flow estimation via convex optimization
- Route flow estimation has received little attention due to data limitations
- Cellular data is a promising data source
- Route flow estimates will enable short time horizon applications, e.g. prediction and control
- Future work: noisy and dynamic settings, experiments with AT&T data



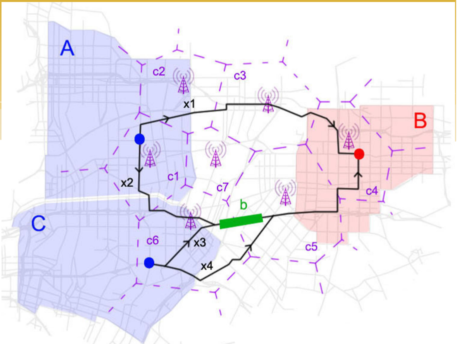
- Website: megacell.github.io
- Implementation: github.com/megacell
- Get in touch: cathywu@eecs.berkeley.edu



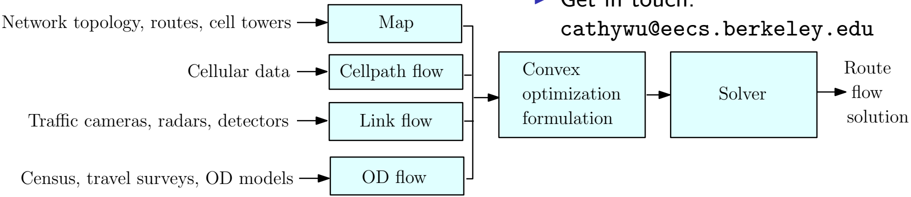
FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Conclusions

- Cellpath: fusion of cellular and traffic sensor data for route flow estimation via convex optimization
- Route flow estimation has received little attention due to data limitations
- Cellular data is a promising data source
- Route flow estimates will enable short time horizon applications, e.g. prediction and control
- Future work: noisy and dynamic settings, experiments with AT&T data



- Website: `megacell.github.io`
- Implementation: `github.com/megacell`
- Get in touch: `cathywu@eecs.berkeley.edu`



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS