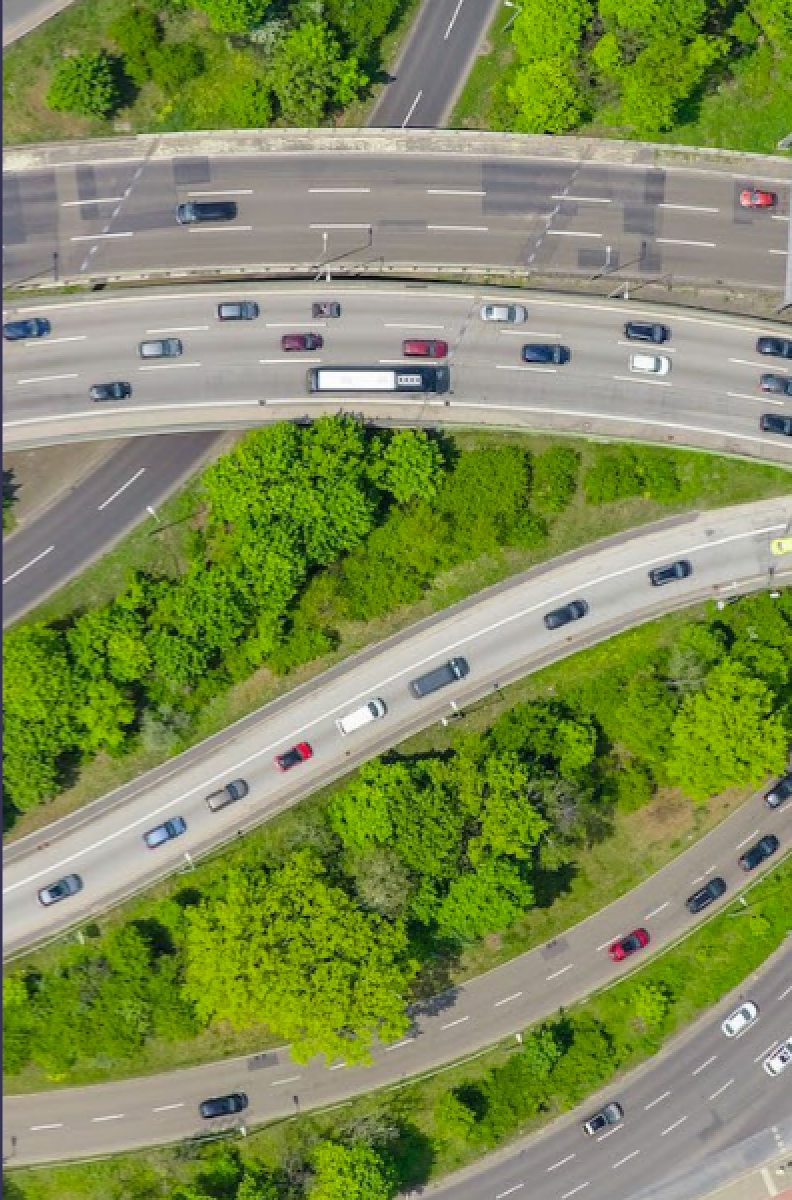
PRIPOSE

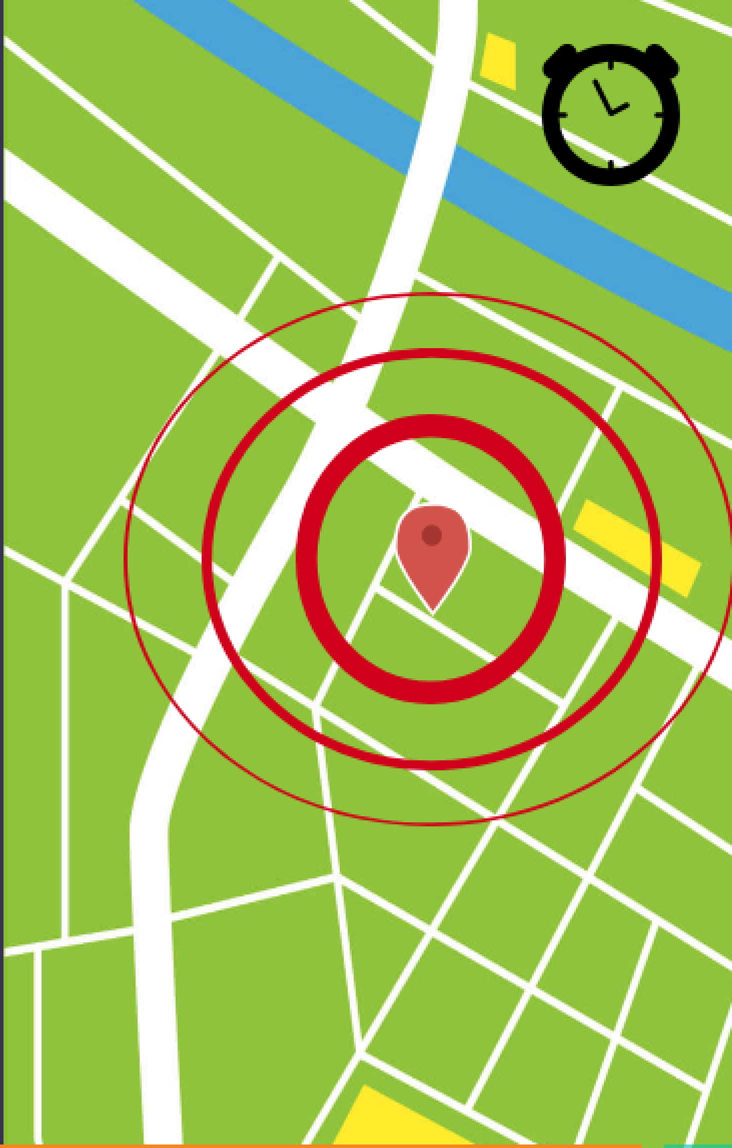
Predictive Rerouting in Presence of Scheduled Events

Anirudh Madhusudan | Brian Thompson | Chenhao Wu | Yuntae Kim



01 MAIN IDEA

- Researchers have found that traffic jams occur spontaneously when the average vehicle density exceeds a certain critical value
- Planned Special Events(PSE) often push traffic density above this critical value
- PRIPOSE Predicts increase in traffic congestion in spatio-temporal proximity to PSE
- PSE: non-periodic events with an expected large attendance
- 150+ routing experiments have shown that PRIPOSE is better at predicting traffic congestion "ahead of time" in presence of PSE when compared to Google Maps



02 HOW PRIPOSE ENHANCES EXISTING METHODS

Google Maps

1B+ Users

54% of all smartphone users have used GM at least once

13 Years worth of data

- Live traffic data from phones - Google knows in real time, how many people are at a given road segment!
- User entered information such as traffic jams, accidents, cops from Waze
- Uses historical data and live traffic data to predict traffic at a given time and date

PRIPOSE

- Current applications only notify nearby users about incidents such as accidents or severe weather conditions, but not PSEs!
- Historical data can only provide suggestions based on the average traffic at a given location and time. PSEs are bursty and won't be evident in typical historical data
- PRIPOSE is novel because it takes PSEs and its spatio-temporal effect on traffic into account while routing

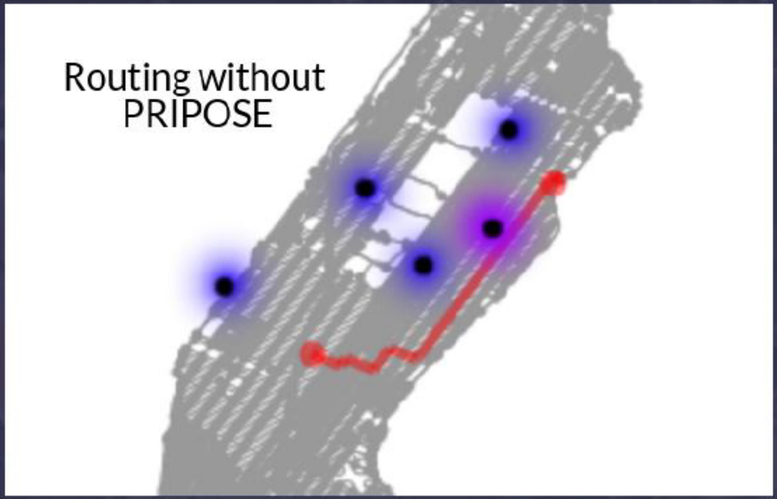
Better Prediction, Better Routing!

03 WHY SHOULD YOU CARE?

Knowledge of future congestion caused by PSEs along a given route will improve route planning

Better routing with knowledge of future congestion will:

- Saves time, energy and more evenly distributes traffic volume
- Helps better decide the time of departure
- Attendees of the PSEs will be able to arrive and depart the venue during less congestion



Not actual results. These maps present a demonstrative example of how the algorithm will potentially work.

04 THE DATA

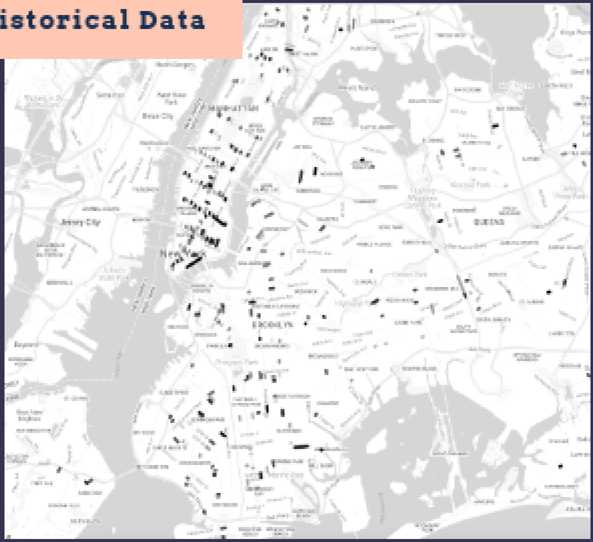
Initial Attempts on Austin, New York, Chicago, San Fransisco was unsuccessful, because:-

- Spatio-temporal sparse data of street segment traffic volume
- Street segment traffic volume data only available for 2012 and before
- Little or no information from past events of events before 2012

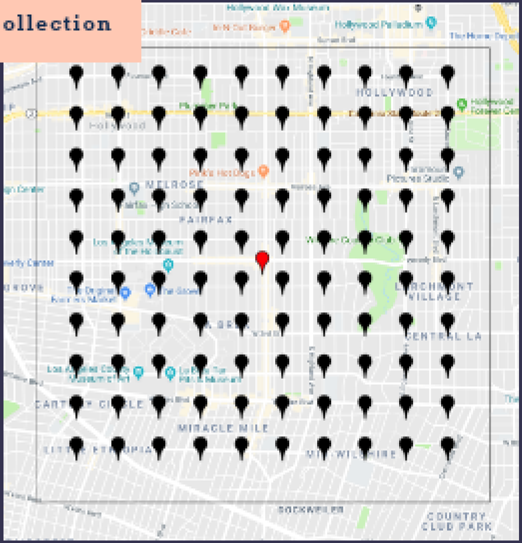
Data Collection using Tom-Tom API

- Hand-picked variety of PSEs using the Eventful website
- Used TomTom API for collecting traffic flow data at 50 points in a 1km radius from the event for a large time window (before, during, after the event)

Historical Data



Live Data Collection

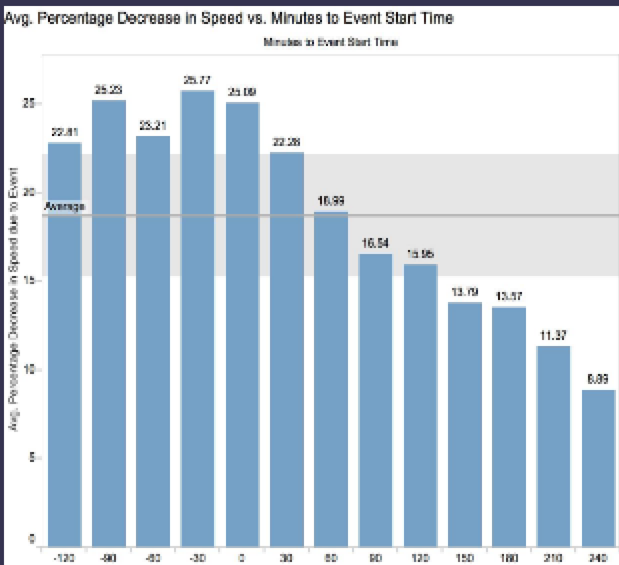


05 EVENTS

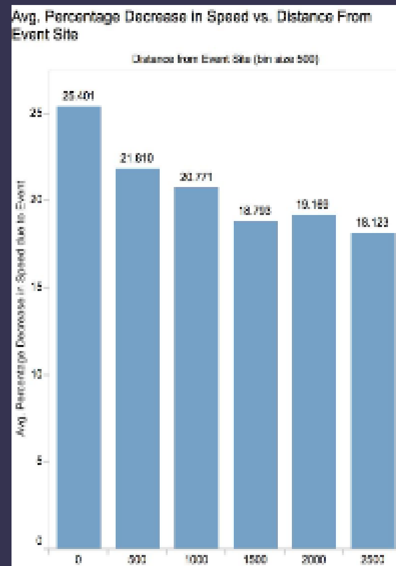
Events Collected For Training

Event ID	Event Name	Category	
E0-001-103317913-9	Hamlet	Performing Arts	March 24, 2018
E0-001-103857672-6	Laugh Factory Long Beach	Nightlife	April 12, 2018
E0-001-104932362-5	Khullam Khulla - Rishi Kapoor	Performing Arts	March 24, 2018
E0-001-108286887-9	Los Angeles Dodgers vs. Arizona Diamondbacks	Sports	April 14, 2018
E0-001-108651019-2	Our Last Night	Concerts	April 7, 2018
E0-001-110558861-4	Gloria Trevi vs. Alejandra Guzman	Concerts	April 14, 2018
E0-001-110565032-8	Matt and Kim	Concerts	April 4, 2018
E0-001-110961095-7	Circa Survive - Hail the Sun & Foxing	Concerts	April 8, 2018
E0-001-110961124-8	Once in a Whale	Performing Arts	March 23, 2018
E0-001-111784050-8	Synapse Animation	Concerts	April 1, 2018
E0-001-111825012-8	The Dustbowl Revival - Shook Twins	Concerts	April 11, 2018
E0-001-112293407-8	Kurt Carr Live ft. Tasha Paige-Lockhart	Performing Arts	April 8, 2018
E0-001-112323987-3	Rotations: Shiva DJ Set	Concerts	March 23, 2018
E0-001-112436039-2	Jamrock Thursdays	Concerts	April 12, 2018
E0-001-112887201-7	LOL IRL: An Improv Comedy Show	Comedy	March 26, 2018
E0-001-112893006-5	Nollywood in Hollywood	Film	March 23, 2018
E0-001-113386241-2	Rakim	Concerts	April 12, 2018
E0-001-113414956-7	TigerHeat	Concerts	April 5, 2018
E0-001-113515176-3	"We Are One!" (LAUSD Benefit Concert)	Performing Arts	April 12, 2018

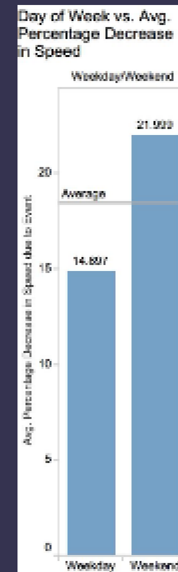
06 FEATURE SELECTION



Avg. percentage of decrease in speed vs. Minutes to event



Avg. percentage of decrease in speed vs. Distance from event site



Avg. percentage of decrease in speed vs. Weekday

Popularity

Popularity of the Event: Popularity Score is a fuzzy number that gives an indication of an event popularity but is not an exact number. In general as the number goes up the more popular the event is.

07 MACHINE LEARNING

- We identify that *distance from the event*, DE and *time to start of event*, TE are the two main independent variables that affect the *percentage change in traffic speed*, S, on non-event day

- 80/20 split for Training/Testing

- Random Forest Regressor gave least error on testing data

- Due to sparsity of information on events (only 19 events), our ML model relies wholly *distance from the event* and *time to start of event.*

- We do want to account for the influence of event parameters on the model. So we heuristically add them on the parameter S.

- S = 0.895*regressor(DE, TE) + 0.005*popularity + 0.1*W +1

08 DESIGN & IMPLEMENTATION

PSEUDO-CODE

Pseudo-code

- Input City, OD pair, time of departure, T_D
- Create OSMnx graph for given City
- Extract current date from the computer, and extract events from Eventful API for current date. Event information will have Time of Event,T_E and Popularity, P.
- W = 1 if date shows it is a friday, saturday or sunday (weekend), 0 otherwise
- Use Google Maps API, obtain Time of Journey, T_J. Comparison Time $T_C = T_D + 0.5 * T_J$
- Loop over all events of the day

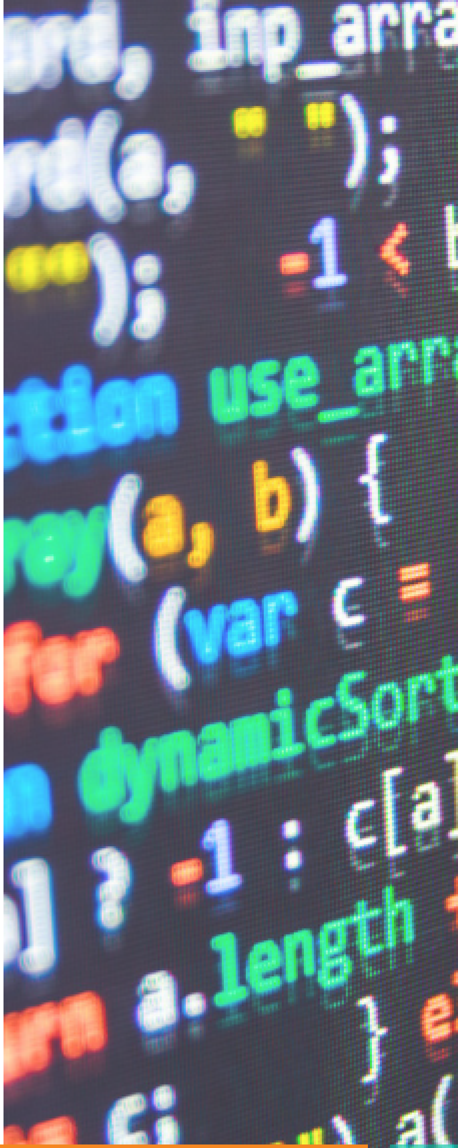
    Time Difference in minutes, Diff = T_E - T_C

    Select event IF (-120 min < Diff < 240 min):

    - ❑ Compute S = 0.895*Regressor(D_E, T_E) + 0.005*P + 0.1*W +1, where D_E is computed for 100,200,300,400,500 m from event site.
    - ❑ Find the distance $D_{E,max}$ in the set (100,200,300,400,500) that has a maximum corresponding S value, S_{max}
    - ❑ For all distances smaller than $D_{E,max}$, set their corresponding S value to S_{max}
    - ❑ For all distances larger than $D_{E,max}$, iteratively set their corresponding S values to 0.9 * $D_{E-1.}$ (i.e S corresponding to 500 m is 0.9 * S corresponding to 400 m)
    - ❑ For distances D_E = 100 to 500:

        Define S_E as the S corresponding to D_E . For all street segments within D_E distance from the event site, update the corresponding length of street segment, i.e segment length = segment length* S_E

- Given that the city graph has been altered to reflect the congested traffic routes, compute the shortest distance path between origin and destination. The algorithm will intelligently circumnavigate through the traffic prone areas caused by scheduled events.



09 DESIGN & IMPLEMENTATION

User Input

Origin
Destination
Time of Departure (Td)

Create OSMnx graph for the city

For current date, extract event info (event start time (Te), popularity)

Loop over all events from this step

Comparsion Time
Tc = Td + 0.5*Tj

Diff = Te - Tc

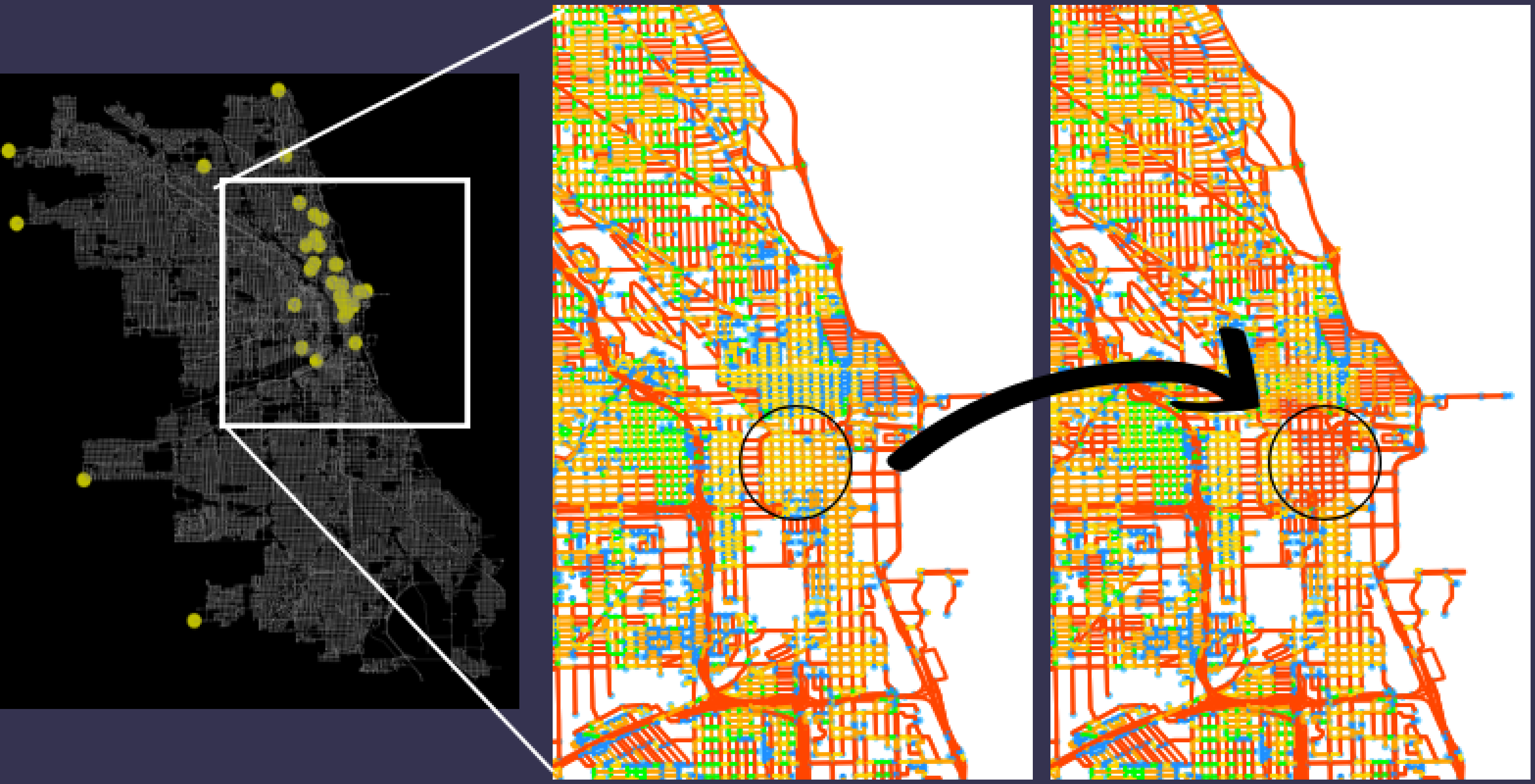
If -120 < Diff < 240
Compute S and extend length of street segments L = L*S
as a function of distance of segment from event site

#Penalty Step

On the new penalized map, use the shortest path algorithm, and output the path

This explanation has been simplified for ease of a higher-level understanding. The actual pseudo-code can be found the previous slide

10 PENALIZED SEGMENTS EXAMPLE



G0: Unpenalized Map

G1: Penalized Map

Color Scheme: Rainbow: More red = longer route

10 ROUTING EXAMPLE

Shortest Route (Suggest by Google)
Uses G0: Unpenalized Map

Events on given Date

PRIPOSE Route
Uses G1: Penalized Map

11 CHALLENGES

- Good spatio-temporal coverage, but sparsity in number of event: Live Traffic Data Collection is **Expensive** & **Time Consuming!**

- Not many explanatory event parameters
  - Model wasn't good when including popularity or day of week as explanatory variables
  - Needed to come up with some heuristics

- To get factual evaluation, real drive tests are needed

12 EVALUATION

- G0: Unpenalized Map & G1: Penalized Map

- Naive route: Shortest route, often the choice with Google's future congestion predictions
- PRIPOSE route: The best route that into consideration traffic congestion from PSEs

- For each OD Pair, get route from PRIPOSE and get a Naive route
  - Get estimated time of the Naive route on G0
  - Calculate the average speed of the Naive route in G0
  - Calculate estimated time for Naive route in G1
  - Calculate estimated time for PRIPOSE's route in G1
  - Calculate the difference in estimated times in G1
  - Calculate the percentage decrease in time from the Naive route to PRIPOSE's route
  - Perform this for many random routes, calculate the *average* percentage decrease on affected routes

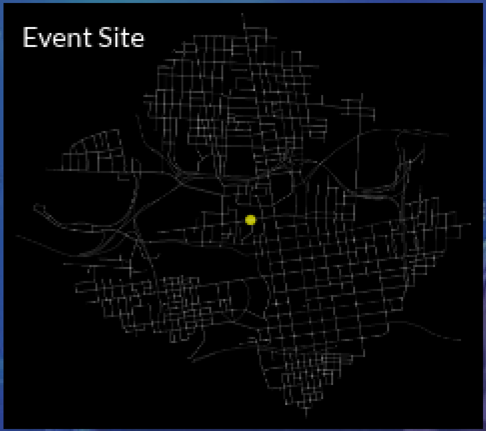
6%

Reduction
in Travel Time

Result: PRIPOSE achieves 6% decrease in travel time with its routing algorithm

13 CASE STUDY

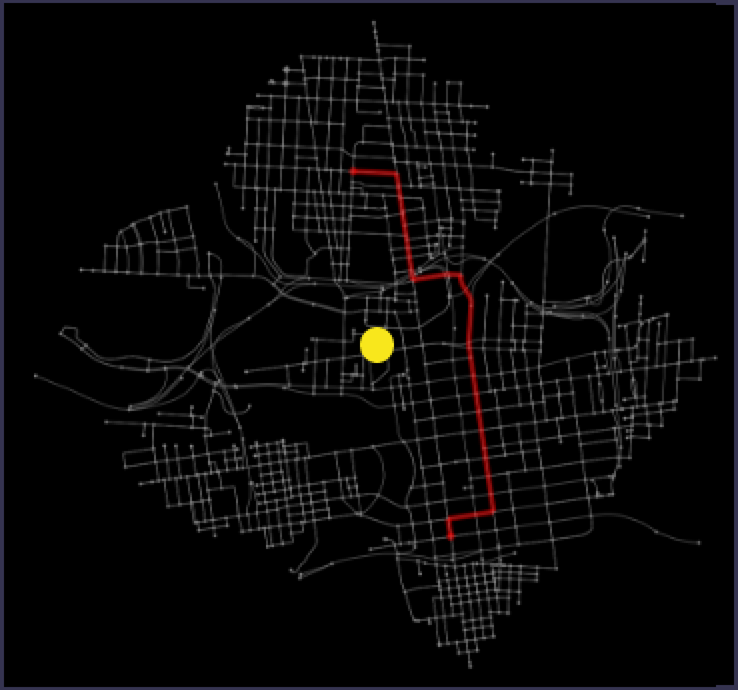
PRIPOSE VS. GOOGLE MAPS



Justin Timberlake Concert
May 7th, 7:30pm (EDT), Columbus, Ohio

13 CASE STUDY

Shortest Route: Based on Road Segment Length

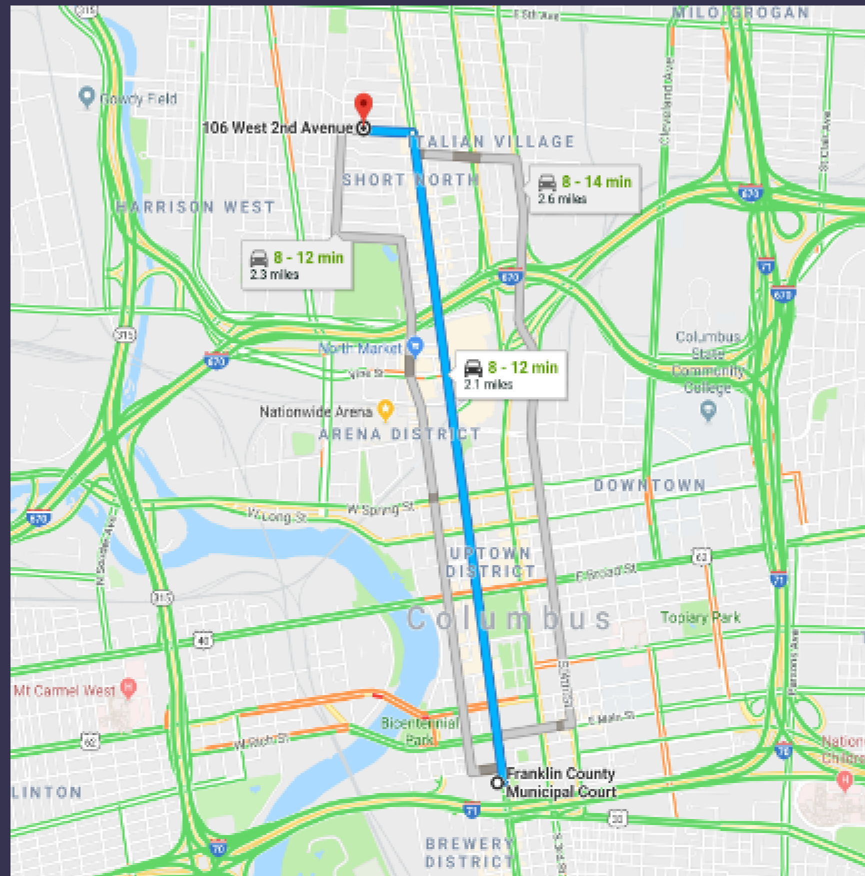


Best Route on Weighted Graph
Weighted Graph Created by PRIPOSE

13 CASE STUDY

Google Maps Predicted Route

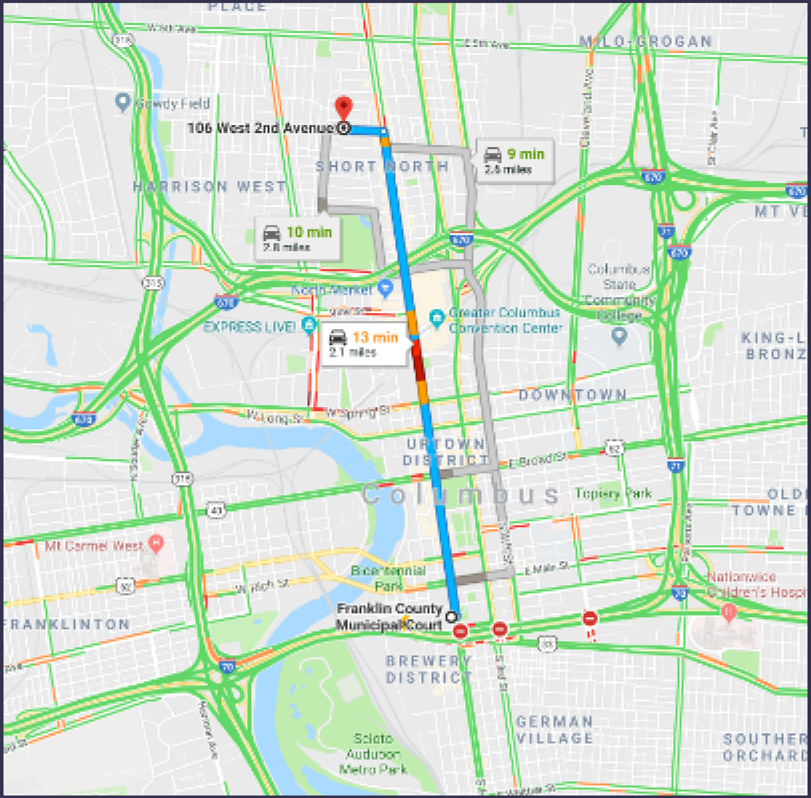
Real Time: 7:30pm
Time of Departure: 10:30 pm



13 CASE STUDY

Google Maps Predicted Route through Live Traffic

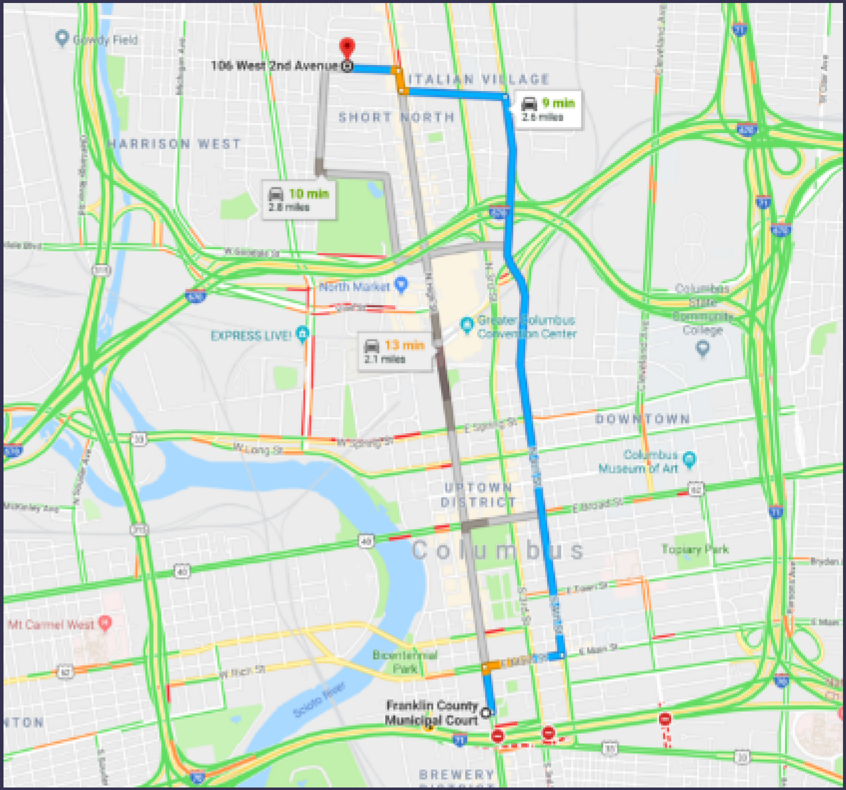
Route predicted at 7:30pm
Overlayed on the Live 10:30 map



13 CASE STUDY

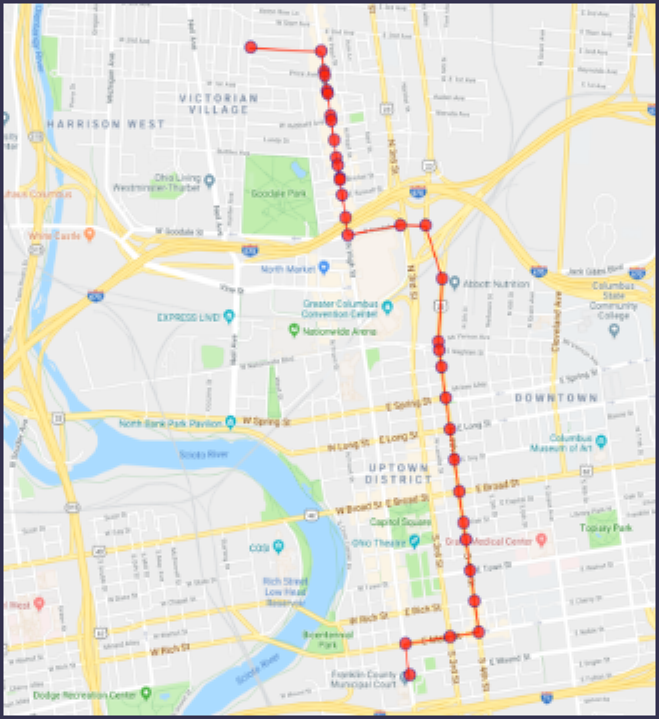
Google Maps Suggest Route through Live Traffic

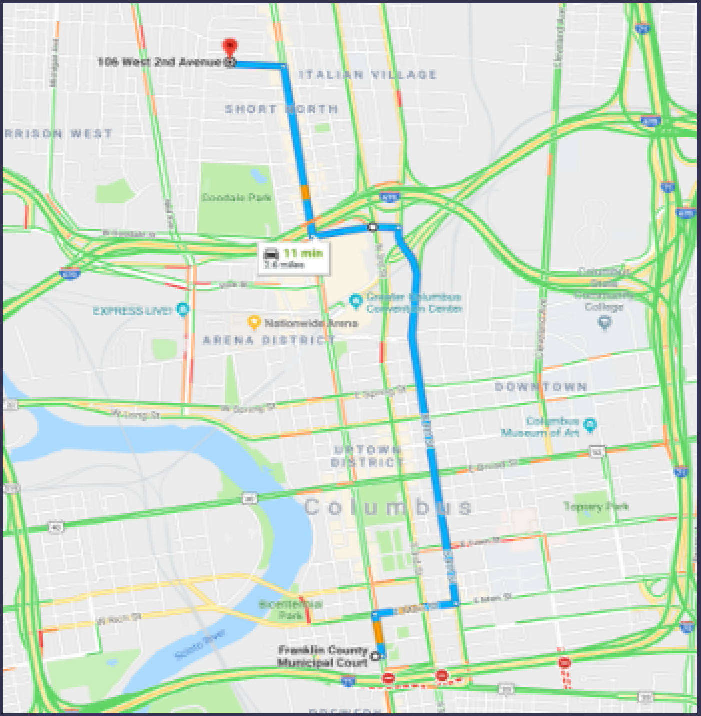
At 10:30pm



13 CASE STUDY

Pripose predicted route in GM





13 CASE STUDY: RESULTS

11 mins

PRIPOSE Route

13 mins

Google Maps Route

PRIPOSE route bypassed congestion by rerouting to the East

14 CONCLUSION

- Our technique shows promise
- Great Potential for future work

  - More Event Parameters
  - Cluster Data
  - Overlay with Live Traffic

- Questions?