

# Algorithms for Real-Time Gathering and Analysis of Continuous-Flow Traffic Data

Jeffrey Miller

Department of Computer Science

University of Southern California

[Jeffrey.Miller@usc.edu](mailto:Jeffrey.Miller@usc.edu)

Ellis Horowitz

Department of Computer Science

University of Southern California

[horowitz@usc.edu](mailto:horowitz@usc.edu)

# Current Applications

- Mapping Applications
  - Mapquest [1]
  - Yahoo Maps [2]
- Navigation Systems



# The Problem

---

- Traffic is a real-time constantly-changing variable
- I consider how communicating real-time traffic data to/from vehicles via wireless, mobile devices can be efficiently processed and used to dynamically produce and adjust optimal traffic paths

# Real-Time Traffic Analysis Today

---

- CalTrans has loop detectors placed between 1/3 and 1/2 mile apart on highways that report the number of vehicles that cross that point and the average amount of time that a vehicle is over that point [22]
  - From this data, an estimate of the speed can be calculated
- The CHP exposes locations of incidents and any additional information they may have
- Sigalert.com [4] provides a nice interface to the data exposed by CalTrans and the CHP by allowing a user to see the traffic conditions along specific highways



# Limitations

---

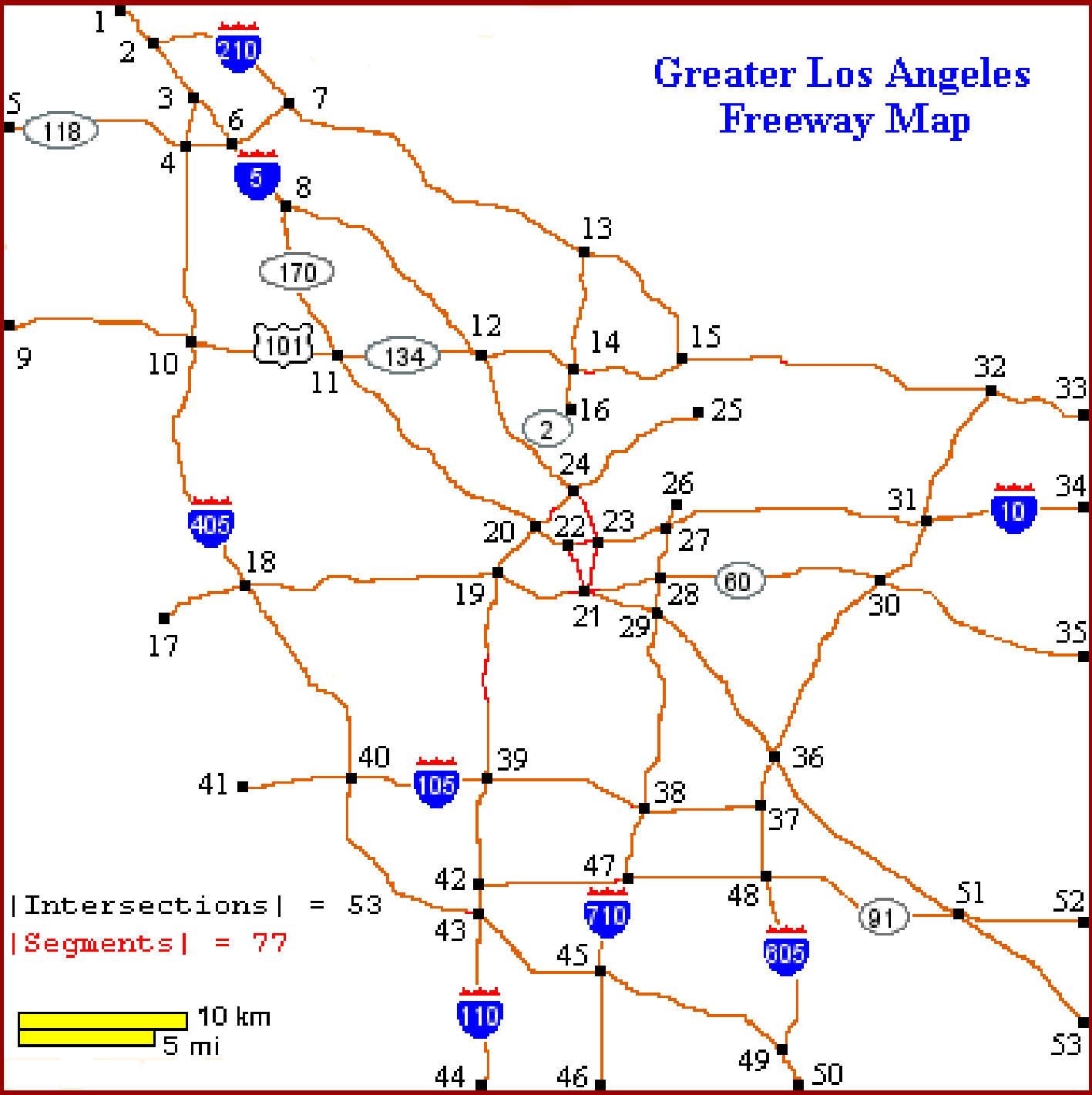
- The data on Sigalert.com is not up-to-minute though, as there is *at least* a 15 minute delay because of CalTrans and the CHP sites
- The sensors owned by CalTrans are at specific points (generally near off-ramps, though not always, and at different distances from each other), and no data is gathered between the points

# The Refined Problem

---

- Given a static map of highways and constantly updated information about the speed and location of all cars on the highways, monitor and optimally determine the fastest route from a car's current location to its destination
  - What processing power is required to compute optimal paths in real-time?

# Greater Los Angeles Freeway Map





# Vertex and Path Definitions

---

- A vertex is defined as an on-ramp or an off-ramp of a highway, as shown on the previous slide
- Initially, a path will be defined as follows:

A path  $P$  on a graph  $G=(V, E)$  is a sequence of vertices  $V_p=\{v_1, \dots, v_j\}$  such that for  $1 \leq i < j$ ,  $(v_i, v_{i+1}) \in E$

# Los Angeles Freeway Specifics

---

- Considering the entire Los Angeles freeway system, similar to the partial graph shown on slide 9, there are a total of 1053 on-ramps and 1088 off-ramps, giving 2141 vertices with 2401 edges in the entire Los Angeles freeway system depicted on slide 8

# Gathering the Speed and Location

---

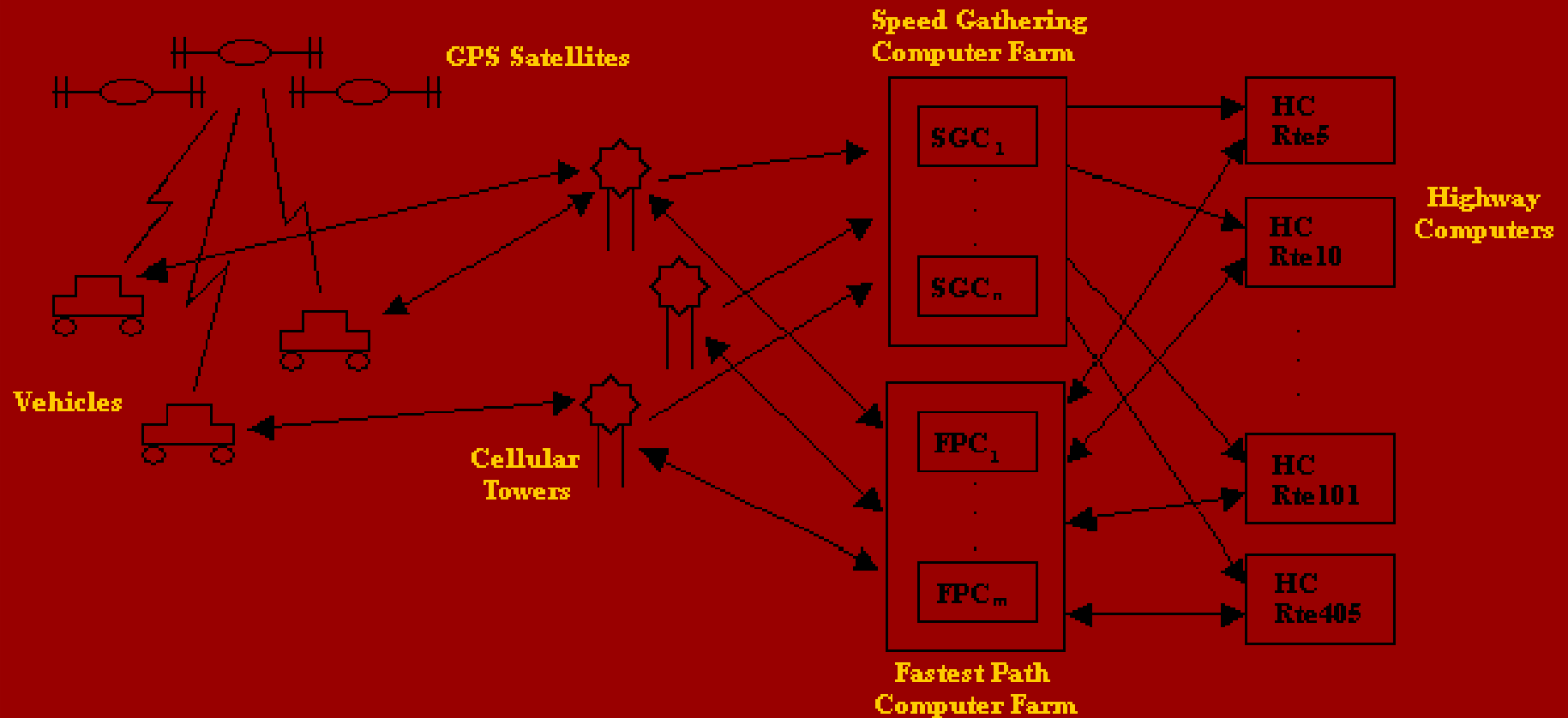
- Assume the vehicles will transmit their speed and location (latitude, longitude) at regular intervals through a cellular link to cell towers that are already positioned along highways
- Using CalTrans AADT data [21], at rush hour in the worst case, there are potentially  $10^6$  vehicles in the Los Angeles freeway system

# Fastest Path Application

---

- Determine the fastest path from the vehicle's current location to its desired destination
- Input parameters
  - current location of a vehicle as supplied by the GPS
  - location of the desired destination as supplied by the driver
- Output
  - the path with the minimum overall time

# Proposed Architecture



3 types of computers: Speed Gathering, Fastest Path, Highway

# Speed Gathering Computer Description

---

- Input
  - (latitude, longitude) pair
  - Speed of the vehicle at that point
- Output
  - Nothing is required to be returned
- Functionality
  - The (latitude, longitude, speed) data from all cars are forwarded to the Speed Gathering Computers
  - These computers map the (latitude, longitude) pair to a highway
  - They then forward the (latitude, longitude, speed) data to the corresponding Highway Computer

# Fastest Path Computer Description

---

- Input
  - Source (latitude, longitude) pair
  - Destination (latitude, longitude) pair
- Output
  - Optimal path from source to destination
- Functionality
  - Map the source (latitude, longitude) and destination (latitude, longitude) to (highway, on-ramp/off-ramp) pairs
  - Assume
    - $d_s$  = distance of shortest path
    - $t_s(0)$  = time to traverse the shortest path with optimal speeds
    - $t_s(t)$  = time to traverse the shortest path with current speeds
    - $z = t_s(t) / t_s(0)$
  - Determine all of the paths with distance less than  $z * d_s$  between the two locations and request speed information from each Highway Computer that is traversed in each path
  - Based on the current speeds, determine the fastest path between the source and destination and return that path

*Note: Fast algorithms are needed here*

# Justification of $z * d_s$ Value

---

- Taking  $z = t_s(t) / t_s(0)$  guarantees that I do not consider alternate paths that, even under optimal conditions, do not have the chance to be faster than the shortest path under the current traffic conditions

# Speed Update and Fastest Path Approaches

---

- Naïve approach
  - Fastest path is recomputed using Johnson's Algorithm [10] when an updated speed is received, and the fastest path is returned when requested
- Dynamic all-pairs shortest path approach
  - Update an edge when a speed is received, query for the fastest path when requested [15]
- All-pairs all-paths pre-computed – Constant Update
- All-pairs all-paths pre-computed – Constant Query
- All-pairs all-paths pre-computed – Hybrid

# Naïve Approach

---

- Floyd [9] and Johnson [10] both presented algorithms for all-pairs shortest paths, though the algorithms had to be re-executed if an edge weight changed (with running times of  $O(V^3)$  and  $O(V^2 \lg V + VE)$ , respectively)

## Dynamic All-Pairs Shortest Path Approach

---

- Demetrescu and Italiano added dynamic behavior in [15], showing that with an edge update cost of  $O(V^2 \log^3 V)$ , they can make the query cost  $O(1)$ , which is slightly faster than other proposed dynamic algorithms [13, 16]

## All-Pairs All-Paths Pre-Computed Approach

- The pre-processing step determines all of the paths between all points in the graph
- All of the paths between any two nodes can be looked up in constant time

## All-Pairs All-Paths Pre-Computed Running Times

---

- Pre-computing the number of paths from one vertex to every other vertex –  $O(E!)$
- ALL-PAIRS-ALL-PATHS running time - # pairs of vertices \* time to compute # paths from one vertex =  $V^2 * E = O(V^2 E!)$

# All-Pairs All-Paths Pre-Computed Constant Update

---

- Update Edge –  $O(1)$ 
  - Take average of last 667 vehicles to transmit speed for the edge or the last 5 minutes (whichever comes first)
  - If the average speed has changed by  $s$  mph (i.e.  $s=10$ ), update the time to traverse the edge
- Retrieve Fastest Path –  $O(mV)$ 
  - Determine the time to traverse all of the  $m$  paths from the source to the destination that have distance less than  $zd_s$  (where  $d_s$  is the length of the shortest path and  $z = t_s(t) / t_s(0)$ ) and return the path with the minimum time

# All-Pairs All-Paths Pre-Computed Constant Query

---

- Update Edge –  $O(V^2 \log m)$ 
  - Take average of last 667 vehicles to transmit speed for the edge or the last 5 minutes (whichever comes first)
  - If the average speed has changed by  $s$  mph (i.e.  $s=10$ ), update the time to traverse the edge and recalculate the fastest paths for all pairs of points that have a path containing that edge
- Retrieve Fastest Path –  $O(1)$ 
  - Retrieve list of all paths for requested pair of nodes and return the fastest path

# All-Pairs All-Paths Pre-Computed Hybrid

---

- Update Edge –  $O(V^2 m)$ 
  - Take average of last 667 vehicles to transmit speed for the edge or the last 5 minutes (whichever comes first)
  - If the average speed has changed by  $s$  mph (i.e.  $s=10$ ), update the time to traverse the edge and all paths containing that edge
- Retrieve Fastest Path –  $O(m)$ 
  - Compare the times to traverse all of the  $m$  paths from the source to the destination that have distance less than  $zd_s$  (where  $d_s$  is the length of the shortest path and  $z = t_s(t) / t_s(0)$ ) and return the path with the minimum time

# Running Time Comparison

	Pre-computation	Update Edge	Retrieve Fastest Path
<b>Naïve (Johnson)</b>	N/A	$O(V^2 \log V + VE)$	$O(1)$
<b>Dynamic All-Pairs Shortest Path (Demetrescu, Italiano)</b>	N/A	$O(V^2 \log^3 V)$	$O(1)$
<b>All-Pairs All-Paths Pre-Computed – Constant Update</b>	$O(V^2 E!)$	$O(1)$	$O(Vm)$
<b>All-Pairs All-Paths Pre-Computed – Constant Query</b>	$O(V^2 E!)$	$O(V^2 m \log m)$	$O(1)$
<b>All-Pairs All-Paths Pre-Computed - Hybrid</b>	$O(V^2 E!)$	$O(V^2 m)$	$O(m)$

$m = \#$  paths between a source and a destination

Note: Updating edges occurs more frequently than computing fastest paths

# Testing the Algorithms Using Simulation

---

- I have conducted some preliminary work creating a simulator to test the previous algorithms to determine whether the actual running time corresponds to the algorithmic running time
- Using the raw data provided by Caltrans' loop detectors, I simulate the number of vehicles entering and exiting the freeway at any given time from any ramp
- The PeMS group at Berkeley estimates the speed at each of the Caltrans' loop detectors [23], enabling me to derive the speeds between detectors

# Future Work

---

- What percentage of vehicles need to use the system to allow accurate fastest paths to be generated?
- If more cars than the percentage from the previous question transmit data, is the additional data necessary, or can the data be aggregated from the cars that are close in proximity?
- If all of the vehicles are routed along fastest paths, will the overall amount of time spent on the freeways by all vehicles decrease?

## Future Work (cont.)

---

- If some of the vehicles are routed along non-optimal paths, will the overall amount of time spent on the freeways by all vehicles be greater than if the optimal path was always returned?
- With such a large amount of data, will exact algorithms be able to execute fast enough to make the system usable, or will heuristic algorithms need to be used?
- Can fastest paths be determined at such a granular level as lanes rather than just freeway segments?

# Future Work (cont.)

---

- Can incidents (accidents, road hazards, lane closures, etc.) be detected and vehicles alerted to the hazards?
- Can the probability of an incident occurring be determined and vehicles alerted to decrease the likelihood of an incident?
- What additional complications arise when considering traffic-regulated streets instead of only free-flowing highways?

Questions?

# Related Work

---

- Methods for monitoring traffic
  - Loop detectors [24]
  - Sensor networks [25] [26]
  - Airsage [27]
- Travel time estimation and prediction [28] [31]
  - Using loop detectors [29] [30]
- Traffic flow [17] [32]

# References

---

- [1] Mapquest. <http://www.mapquest.com>.
- [2] Yahoo Maps. <http://maps.yahoo.com>.
- [3] Shrank, David and Tim Lomax. “2004 Urban Mobility Report.” Texas Transportation Institute’s Annual Urban Mobility Report, September 2004.
- [4] Sigalert.com. <http://www.sigalert.com>.
- [5] Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. Introduction to Algorithms – 2<sup>nd</sup> Edition, The MIT Press, 2001, 527.
- [6] Bellman, Richard. “On a Routing Problem.” Quarterly of Applied Mathematics, Volume 16, 1958.
- [7] Ford, Lestor, D R Fulkerson. Flows in Networks, Princeton University Press, 1962.
- [8] Dijkstra, E.W. “A note on two problems in connexion with graphs.” Numerische Mathematik, 1959.
- [9] Floyd, Robert. “Algorithm 97 (SHORTEST PATH).” Communications of the ACM, 1962.

## Related Work (cont.)

---

- Fastest path and incident identification using loop detectors [19, 33]
- Necessary amount of data for accurate routing [34]
- V2V and V2R communication [35, 36], C2P2 [37]
- Cellular probing [38, 39, 40]

## Related Work (cont.)

---

- Single Source Shortest Path – Floyd [9], Dijkstra [8], Bellman-Ford [6, 7]
- All Pairs Shortest Path [14, 41, 42] – Johnson [10]
- Dynamic All Pairs Shortest Path [15, 43]

# References

---

- [10] Johnson, Donald. “Efficient Algorithms for Shortest Paths in Sparse Networks.” Journal of the ACM, 1977.
- [11] Romeijn, H. Edwin and Robert Smith. “Parallel Algorithms for Solving Aggregated Shortest Path Problems.” ACM Computers and Operations Research, Volume 26, September 1999.
- [12] Djidjev, H.N., G.E. Pantziou, and C.D. Zaroliagis. “Improved Algorithms for Dynamic Shortest Paths.” Algorithmica, Volume 28, 2000.
- [13] Klein, P.N. and S. Subramanian. “A Fully Dynamic Approximation Scheme for Shortest Paths in Planar Graphs.” Algorithmica, Volume 22, 1998.
- [14] Seidel, Raimund. “On the All-Pairs-Shortest-Path Problem.” 24<sup>th</sup> Annual ACM Symposium on Theory of Computing, 1992.
- [15] Demetrescu, Camil and Giuseppe Italiano. “A New Approach to Dynamic All Pairs Shortest Paths.” ACM Symposium on Theory of Computing, June 2003.
- [16] King, Valerie. “Fully Dynamic Algorithms for Maintaining All-Pairs Shortest Paths and Transitive Closure in Digraphs.” IEEE Symposium on Foundations of Computer Science, 1999.

# References

---

- [17] Li, Kim and Petros Ioannou. “Modeling of Traffic Flow of Automated Vehicles.” IEEE Transactions on Intelligent Transportation Systems, Volume 5, Number 2, June 2004.
- [18] Srinivasan, Dipti, Xin Jin, and Ruey Long Cheu. “Evaluation of Adaptive Neural Network Models for Freeway Incident Detection.” IEEE Transactions on Intelligent Transportation Systems, Volume 5, Number 1, March 2004.
- [19] Lin, Wei-Hua and Carlos Daganzo. “A Simple Detection Scheme for Delay-Inducing Freeway Incidents.” Transportation Research A, Volume 31, Number 2, 1997.
- [20] Antonakos, James L. and Kenneth C. Mansfield. Practical Data Structures Using C/C++. Prentice Hall College Div, January 1999.
- [21] CalTrans 2003 AADT – Average Annual Daily Traffic, 2003, <http://www.dot.ca.gov/hq/traffops/saferesr/trafdata/>.
- [22] California Center for Innovative Transportation.  
[http://www.calccit.org/itsdecision/serv\\_and\\_tech/Archived\\_Data/TRB00443.pdf](http://www.calccit.org/itsdecision/serv_and_tech/Archived_Data/TRB00443.pdf)

# References

---

- [23] PeMS – Performance Measurement System. <http://pems.eecs.berkeley.edu>.
- [24] Jia, Zhanfeng, Chao Chen, Ben Coifman, Pravin Varaiya. “The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors.” *IEEE 4<sup>th</sup> International Intelligent Transportation Systems Conference*, February 12, 2001.
- [25] Coleri, Sinem, Sing Yiu Cheung, Pravin Varaiya. “Sensor Networks for Monitoring Traffic.” *42<sup>nd</sup> Annual Allerton Conference on Communication, Control, and Computing*, September 2004.
- [26] Ding, Jiagen, Sing-Yiu Cheung, Chin-Woo Tan, Pravin Varaiya. “Signal Processing of Sensor Node Data for Vehicle Detection.” *IEEE 7<sup>th</sup> International Intelligent Transportation Systems Conference*, October 2004.
- [27] Airsage. <http://www.airsage.com>.
- [28] Rice, John, Erik van Zwet. “A Simple and Effective Method for Predicting Travel Times on Freeways.” *IEEE 5<sup>th</sup> International Intelligent Transportation Systems Conference*, August 2002.

# References

---

- [29] Kwon, Jaimyoung, Benjamin Coifman, Peter Bickel. “Day-to-Day Travel Time Trends and Travel Time Prediction from Loop Detector Data.” Transportation Research Board #1717, 2000.
- [30] Petty, Karl F., Peter Bickel, Jiming Jiang, Michael Ostland, John Rice, Ya’acov Ritov, Frederic Schoenberg. “Accurate estimation of travel times from single-loop detectors.” Transportation Research, Part A (Policy and Practice), Volume 32A, #1, January 1998.
- [31] Choe, Tom, Alexander Skabardonis, Pravin Varaiya. “Freeway Performance Measurement System (PeMS): An Operational Analysis Tool.” *81<sup>st</sup> Annual Meeting Transportation Research Board*, January 2002.
- [32] Bickel, Peter, Chao Chen, Jaimyoung Kwon, John Rice, Pravin Varaiya, Erik van Zwet. “Traffic Flow on a Freeway Network.” *Proceeding of MSRI Workshop on Nonlinear Estimation and Classification*, March 19-29, 2001.
- [33] PeMS Group. “Freeway Performance Measures – Calculations with Loop Detectors.” August 22, 1999.

# References

---

- [34] Kim, Seongmoon, Mark E. Lewis, Chelsea C. White, III. “State Space Reduction for Nonstationary Stochastic Shortest Path Problems With Real-Time Traffic Information.” IEEE Transactions on Intelligent Transportation Systems, Volume 6, Number 3, September 2005.
- [35] Moustafa, Hasnaa, Gilles Bourdon, Yvon Gourhant. “AAA in Vehicular Communication on Highways with Ad hoc Networking Support: A Proposed Architecture.” *2<sup>nd</sup> ACM International Workshop on Vehicular Ad Hoc Networks*, September 2005.
- [36] Wu, Hao, Mahesh Palekar, Richard Fujimoto, Jaesup Lee, Joonho Ko, Randall Guensler, Michael Hunter. “Vehicular Networks in Urban Transportation Systems.” *Digital Government Conference*, May 2005.
- [37] Ghandeharizadeh, Shahram, Bhaskar Krishnamachari. “C2P2: A Peer-to-Peer Network for On-Demand Automobile Information Services.” *1<sup>st</sup> International Workshop on Grid and Peer-to-Peer Computing*, August 2004.
- [38] Yim, Youngbin. “The State of Cellular Probes.” *California Partners for Advanced Transit and Highways Research Report UCB-ITS-PRR-2003-25*, July 2003.

# References

---

- [39] Cayford, Randall, Tigran Johnson. “Operational Parameters Affecting the Use of Anonymous Cell Phone Tracking for Generating Traffic Information.” *Transportation Research Board 2003 Annual Meeting*, 2003.
- [40] Schaefer, Lisa. “Architecture Using JINI Technology for Simulation of an Agent-Based Transportation System.” *Proceedings of the 2001 Winter Simulation Conference*, 2001.
- [41] Cohen, Edith, Uri Zwick. “All-Pairs Small-Stretch Paths.” *Symposium on Discrete Algorithms*, 1997.
- [42] Takaoka, T. “Subcubic Cost Algorithms for the All Pairs Shortest Path Problem.” *Algorithmica*, Volume 20, 1998.
- [43] Roditty, Liam, Uri Zwick. “On dynamic shortest paths problems.” *Proceedings of the 12<sup>th</sup> European Symposium on Algorithms*, 2004.
- [44] Schofer, Ralph E., Franklin F. Goodyear. “Electronic computer applications in urban transportation.” *Proceedings of the 1967 22<sup>nd</sup> National Conference*, 1967.

# References

---

- [45] Clark, Jim, Gene Daigle. “The Importance of Simulation Techniques in ITS Research and Analysis.” *Proceedings of the 29<sup>th</sup> Conference on Winter Simulation*, 1997.
- [46] Xu, Jinghua, Kathleen L. Hancock, Frank Southworth. “Dynamic Freight Traffic Simulation Providing Real-Time Information.” *Proceedings of the 35<sup>th</sup> Conference on Winter Simulation*, 2003.
- [47] Brummer, Daniel, Gary Cross, Jack Levis, Catherine McGhee, Dudley Whitney. “Toward Increased Use of Simulation in Transportation.” *Proceedings of the 30<sup>th</sup> Conference on Winter Simulation*, 1998.
- [48] Wang, Paul T.R., Richard A. Glassco. “Enhanced THOREAU Traffic Simulation for Intelligent Transportation Systems (ITS).” *Proceedings of the 27<sup>th</sup> Conference on Winter Simulation*, 1995.
- [49] Fishburn, Paul T., Javad Golkar, Kevin M. Taaffe. “Simulation of Transportation Systems.” *Proceedings of the 27<sup>th</sup> Conference on Winter Simulation*, 1995.