Planning T(r)ips for Hybrid Electric Vehicles

How to Drive in the 21st Century

Andrew Wang and Peng Yu

16.S949 Student Lecture

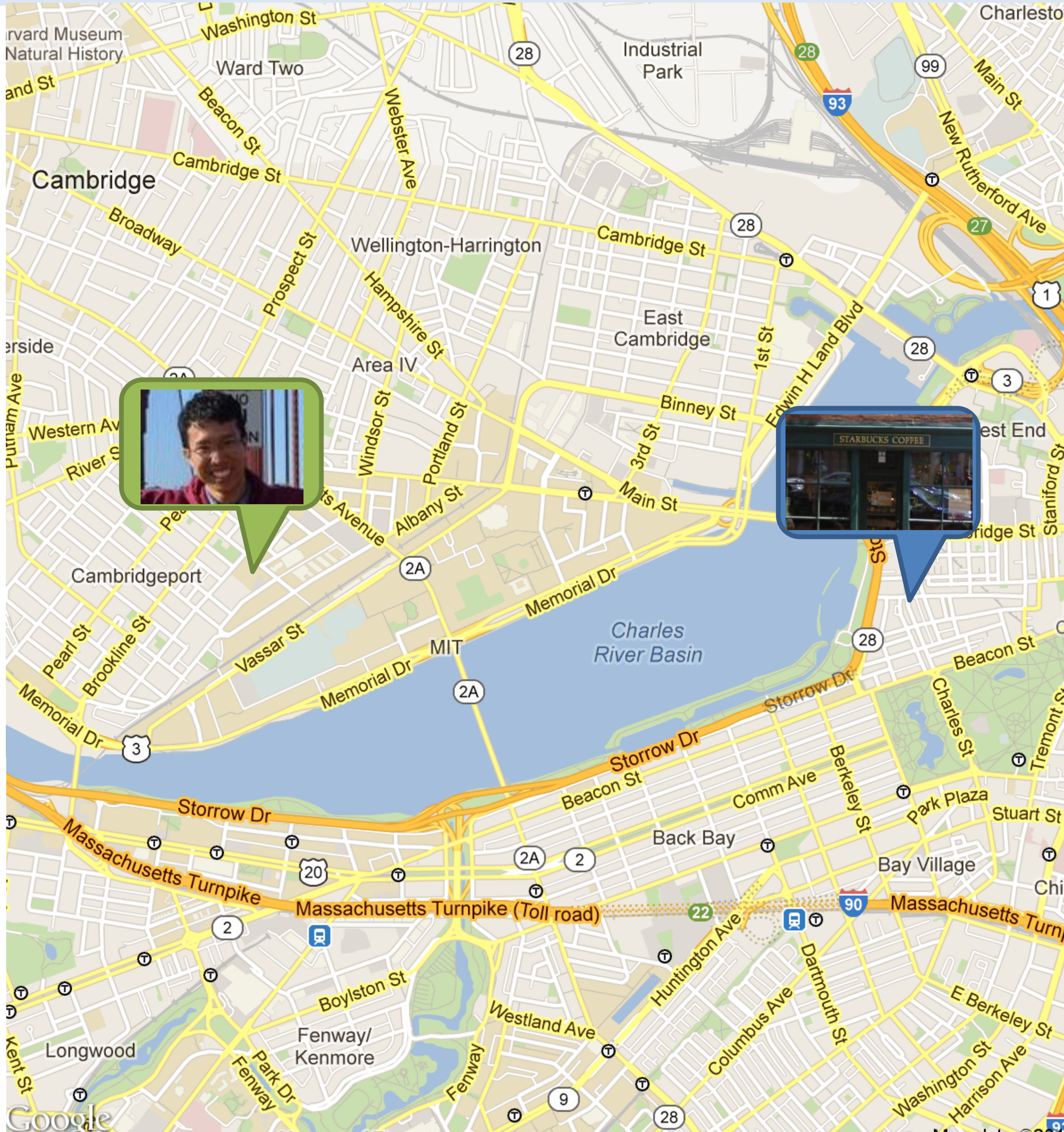
May 14th, 2012

Example

- Origin: Sid-Pac
- Destination: Revere St.

Meet Peng in 4 minutes.

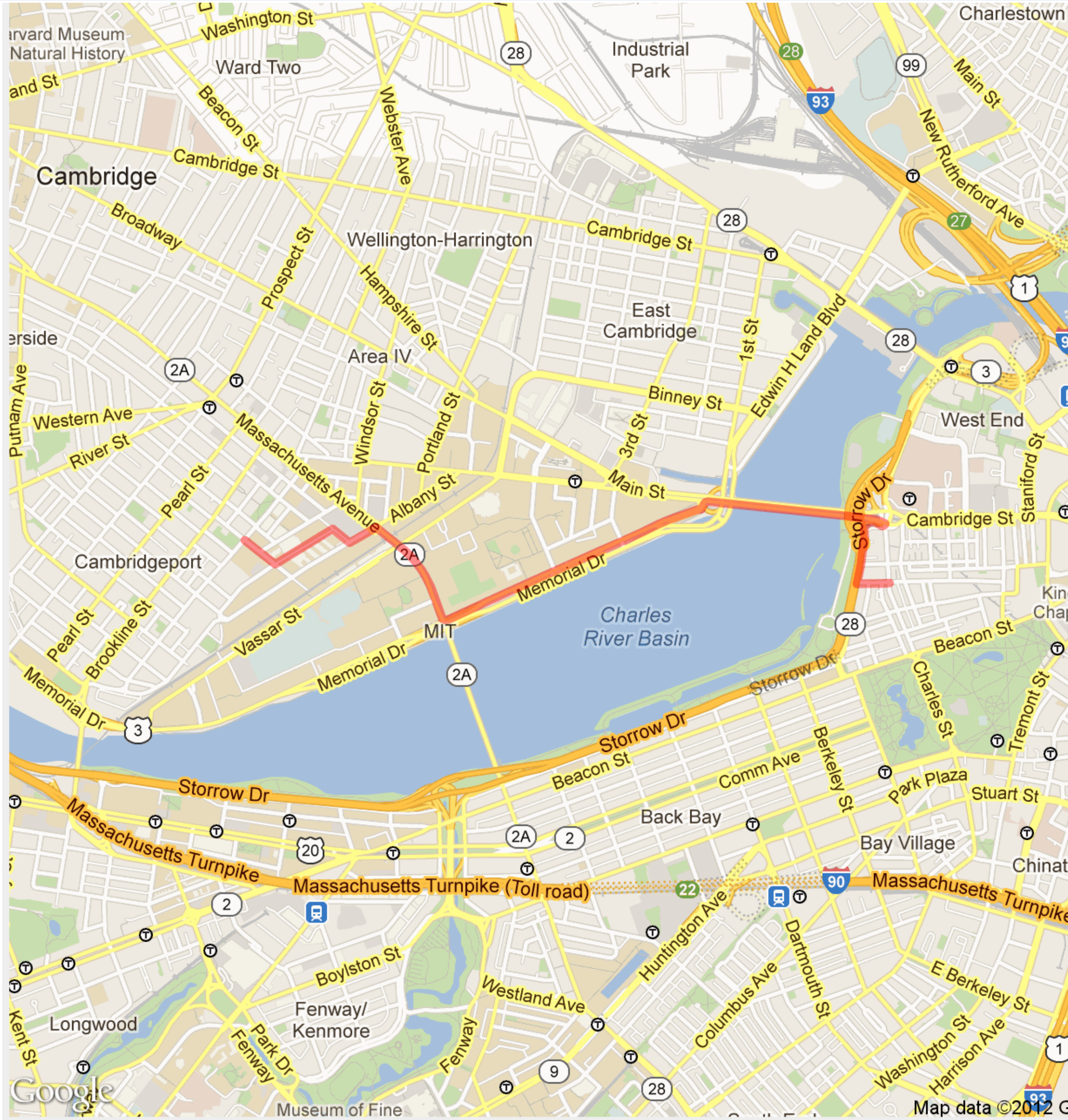
Need to find a path.



Example: least-time path

Google Maps gives the path that minimizes trip duration.

- Duration: 3.6 minutes.
- Fuel: 0.193L.



Route Time: 222.58s Route Fuel: 0.19312L Route distance: 3.4879km

Example: least-fuel path

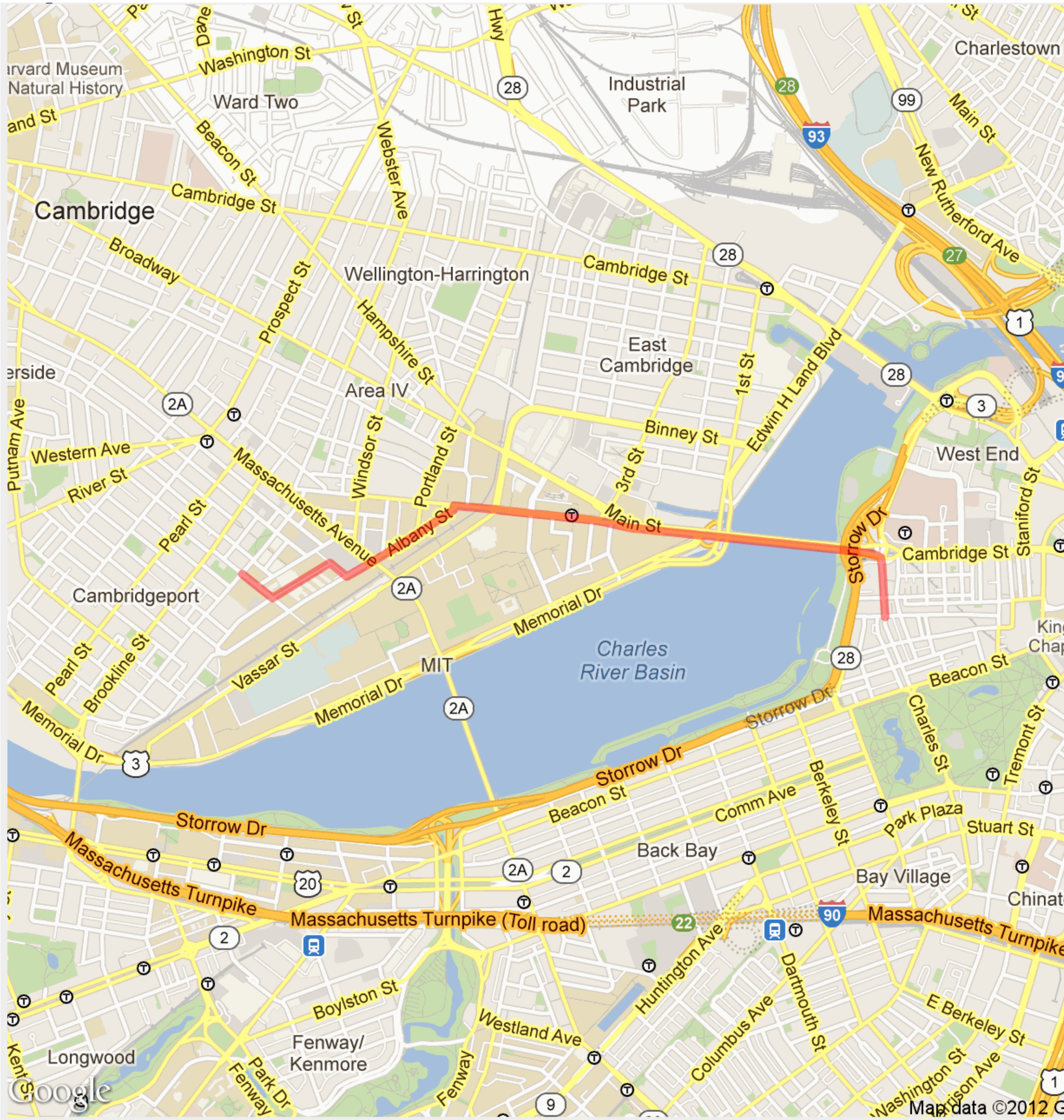
We can find a path that minimize fuel usage[1].

- Duration: 5 minutes.
- Fuel: 0.152L.

That's 25% saving!

But he will be late.

[1]R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. Greengps: a participatory sensing fuel efficient maps application. In *Proc. of MobiSys'10*, pages 151–164, San Francisco, CA, 2010.



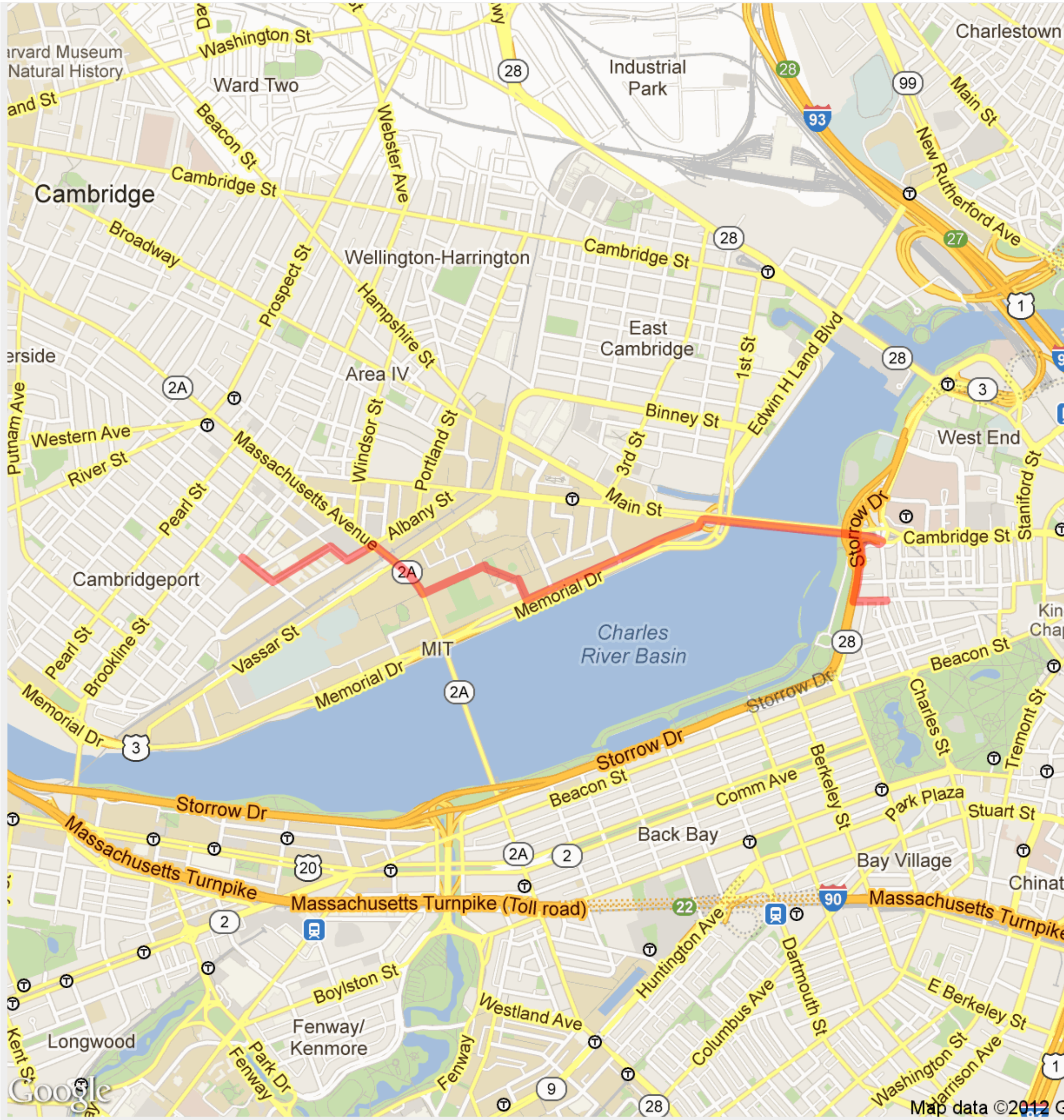
Route Time: 301.02s Route Fuel: 0.15221L Route distance: 3.0373km

Example: least-fuel path that is on-time

We want to find the path fuel usage within the timing constraint.

- Duration: 4 minutes.
- Fuel: 0.185L.

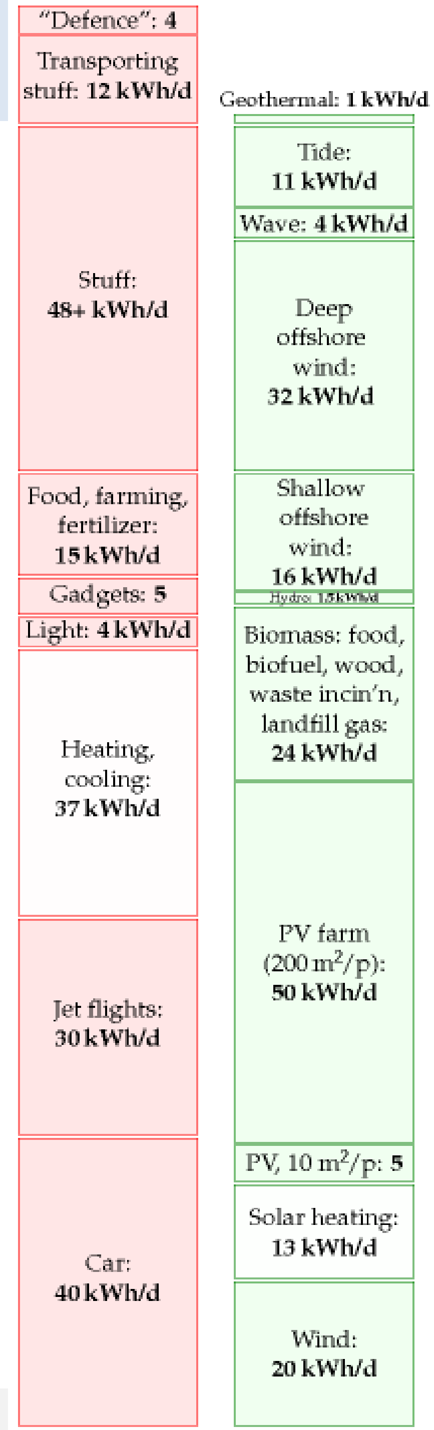
But how?



Route Time: 249.98s Route Fuel: 0.18554L Route distance: 3.4340km

Planning for Hybrid Electric Vehicles

- Motivation & Problem Formulation
- The Best Route
- The Best Driving Style
- Examples and Summary



Source: D. J. C. MacKay, *Sustainabl e Energy – Without the Hot Air*, UIT Cambridge, 2009.

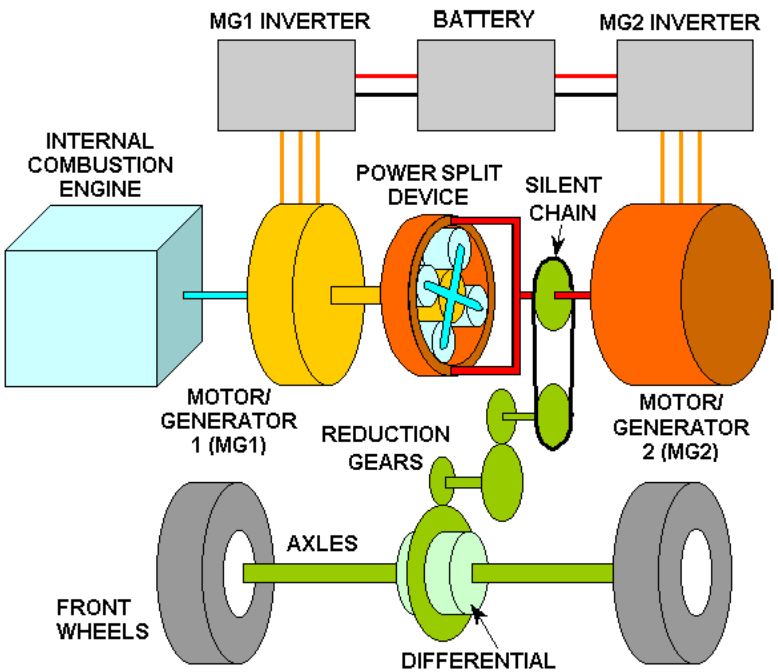
Hybrid cars are popular!

- Toyota Prius was Japan's best-selling car in 2011.
  - 252,528 units sold over the year.
- Why do people like hybrid vehicles?
  - 50% increase in City/Hwy MPG.
  - 40% reduction in carbon emission.

The hybrid system

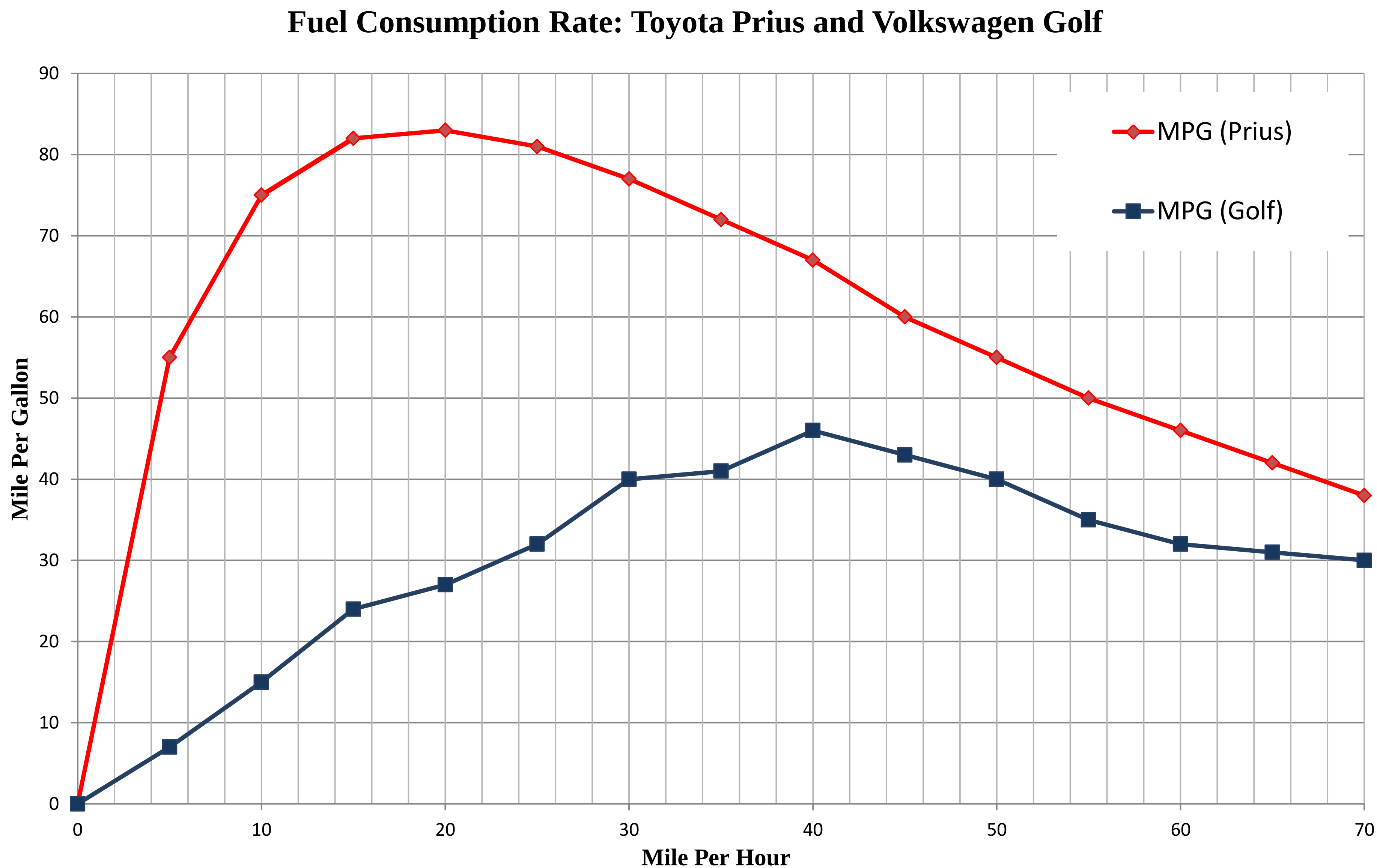
- Electric motor (batteries) and gasoline engine:

- Saving energy through:
  - Keep the engine working in its efficient zone.
  - Avoid low speed crawling with ignited engine.



Source: G. Davies, http://prius.ecrostech.com/original/Understanding/PowerTrain.htm, 2012.

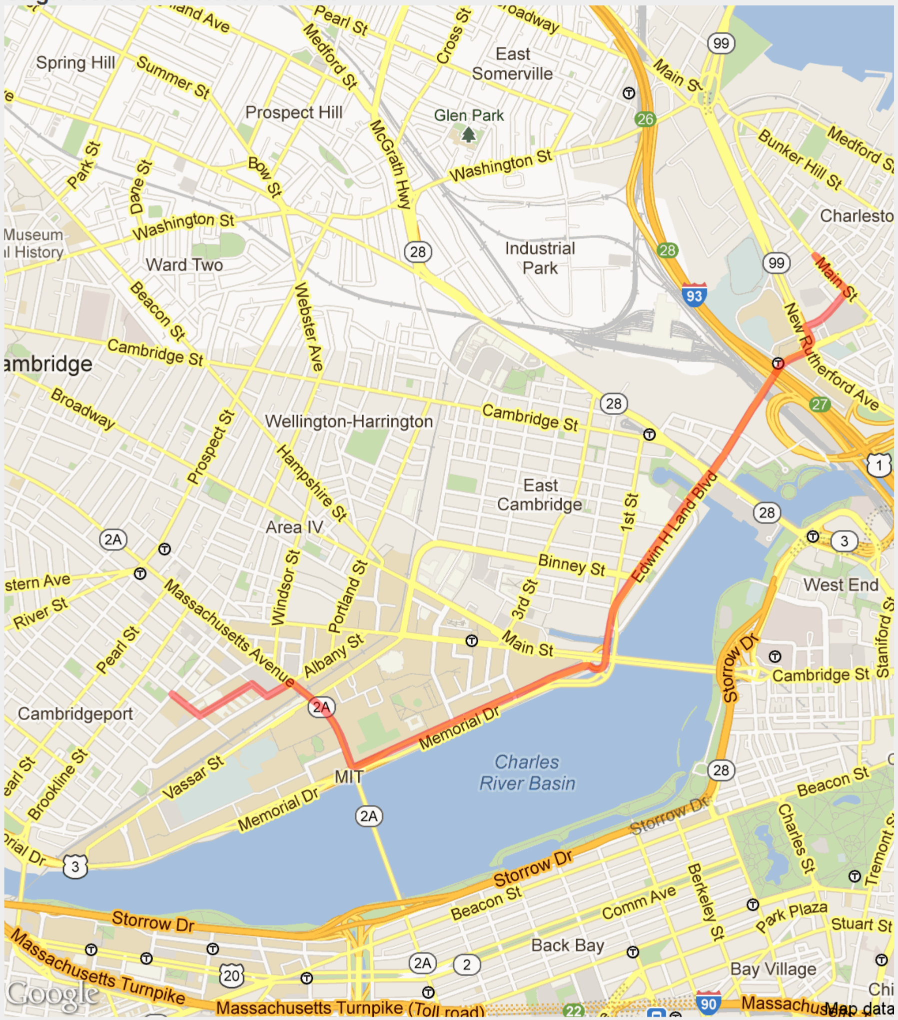
Superb low speed MPG



Source: US Environmental Protection Agency and metrompg.com

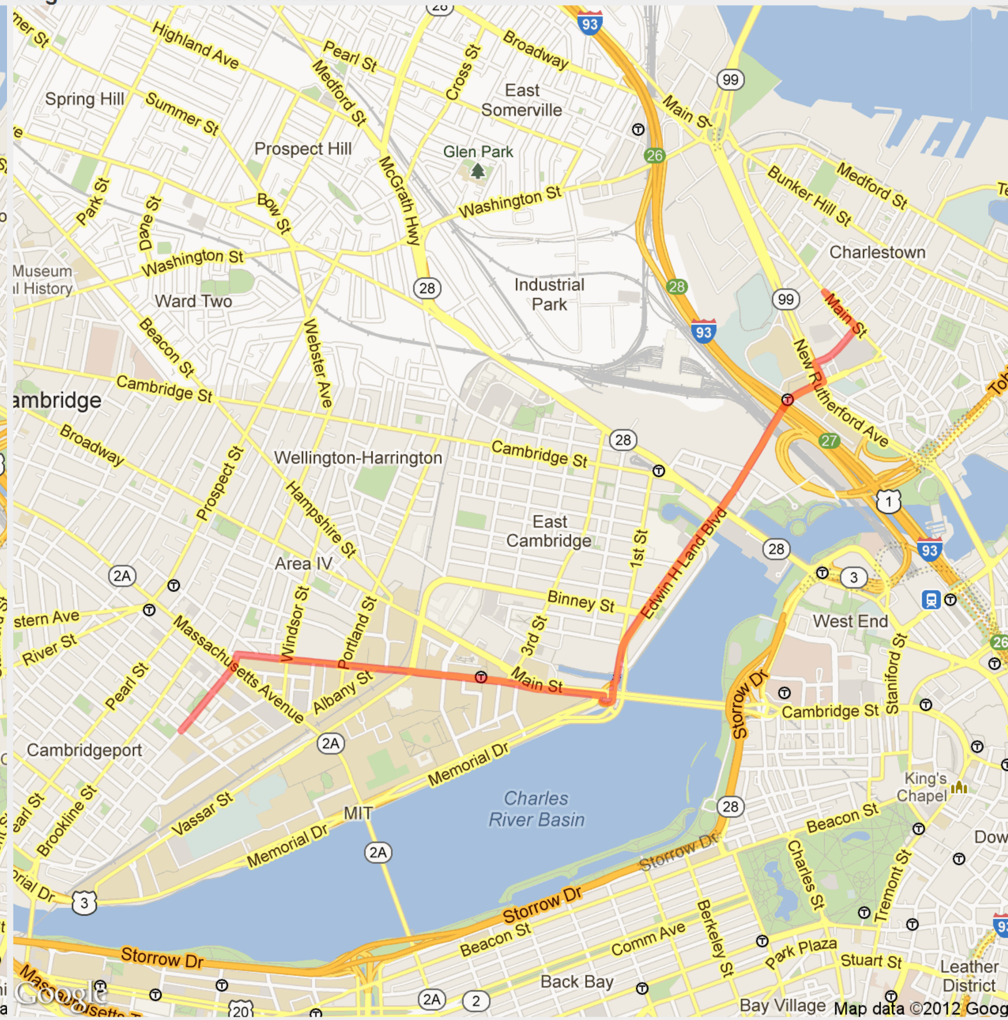
Routing affects fuel economy

Dijkstra Time Priority File: L8.osm
Nodes: 52219 Arcs: 52156 Run time: 8885ms Load Time: 538958ms
Orig: 61331888 Dest:61345626



Route Time: 350.88s Route Fuel: 0.25033L Route distance: 4.6644km

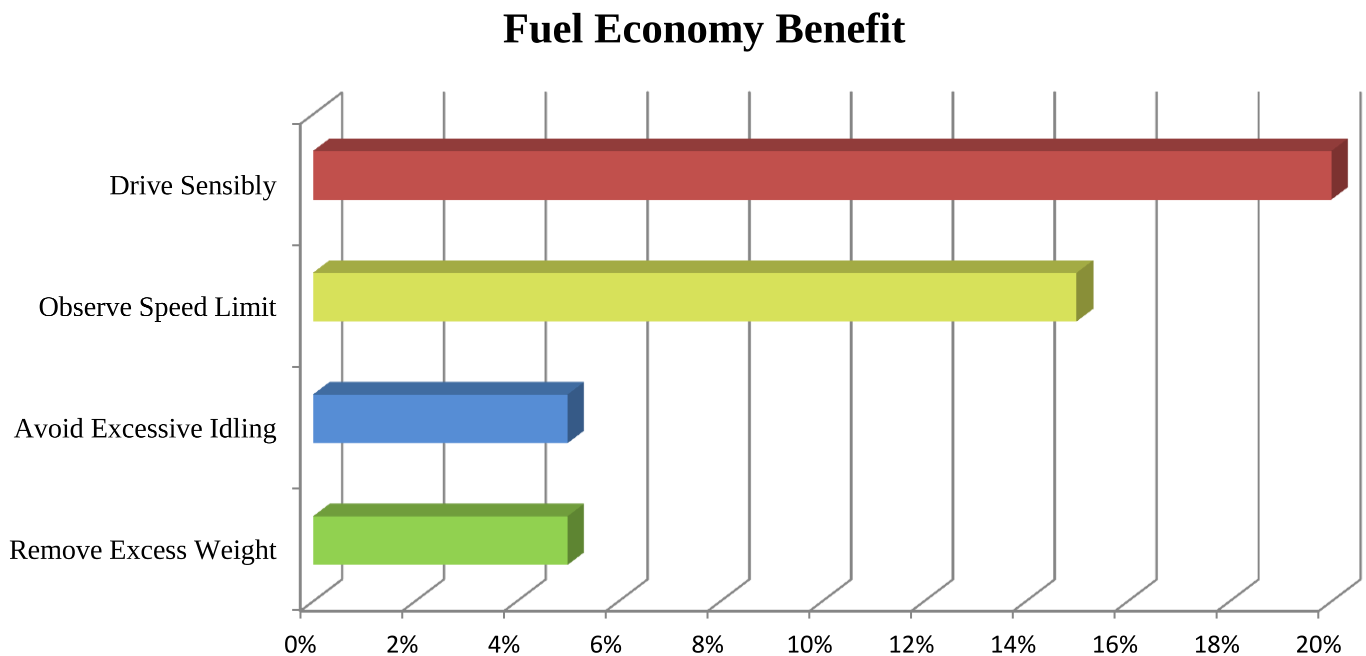
BFH 450.0s File: L8.osm
Nodes: 51577 Arcs: 51531 Run time: 31399ms Load Time: 536605ms
Orig: 61331888 Dest:61345626



Route Time: 449.91s Route Fuel: 0.22152L Route distance: 4.4477km

Driving style affects fuel economy

- Percent improvement via route-independent methods
  - Avoid aggressive driving: 5 - 33%.
  - Keep optimal fuel economy speed: 7 - 23%.
  - Remove excess weight: 1 - 2%/100lb.
  - Avoid excessive idling: 0.02 - 0.04 L/min.



Source: US Department of Energy, www.fueleconomy.gov.

A driving advisory system

- Input:
  - Where you want to go in how long.
  - A map of the road network with traffic conditions.
  - A model of the hybrid car's dynamics.

- Output:
  - **Recommend the *route*** that satisfies timing constraints while minimizing fuel consumption.
  - **Provide driving *guidance*** for avoiding aggressive driving and over-speed cruising.

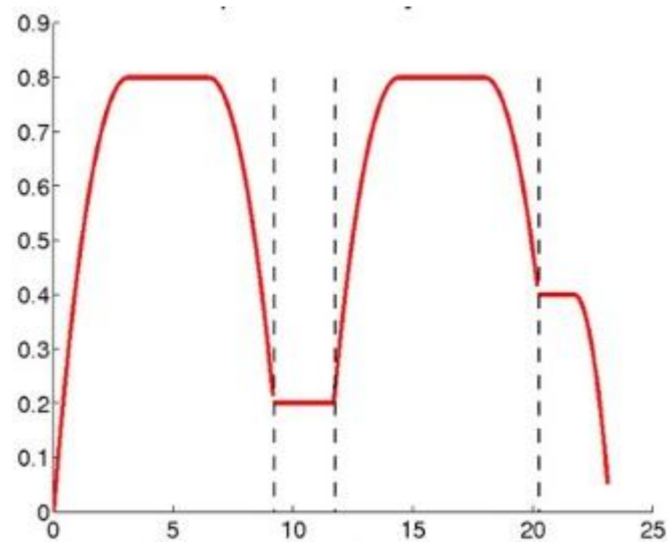


Image Source: Pratap Tokekar, Nikhil Karnad, and Volkan Isler.Energy-optimal velocity profiles for car-like robots. In *ICRA*, 2011(submitted).

The constrained optimization problem

- Objective: Minimize the fuel consumption of a trip.
  - By:

Choosing a smart route: $Seg_1, Seg_2,...,Seg_N.$

Using proper acceleration and braking: $Acc_1, Dec_1,..., Acc_N, Dec_N.$

And economy cruising speed: $Vel_1, Vel_2,...,Vel_N.$

  - To minimize:

$$TotalGas = \sum_{k=1...N} Fuel(Seg_k, Acc_k, Dec_k, Vel_k)$$

The constrained optimization problem

- Time:

$$\sum_{k=1\ldots N} Time(Seg_k, Acc_k, Dec_k, Vel_k) < TimeLimit$$

- Traffic:

$$Vel_k < SpeedLimit_k$$

- Vehicle dynamics:

$$\forall k, \qquad Acc_k < MaxAcc, \qquad Dec_k < MaxBrake$$

- A complete path connecting the origin and destination.

The constrained optimization problem

- Objective: Minimize the fuel consumption of a trip.
  - By:

Choosing a smart route: $Seg_1, Seg_2,\ldots,Seg_N$.

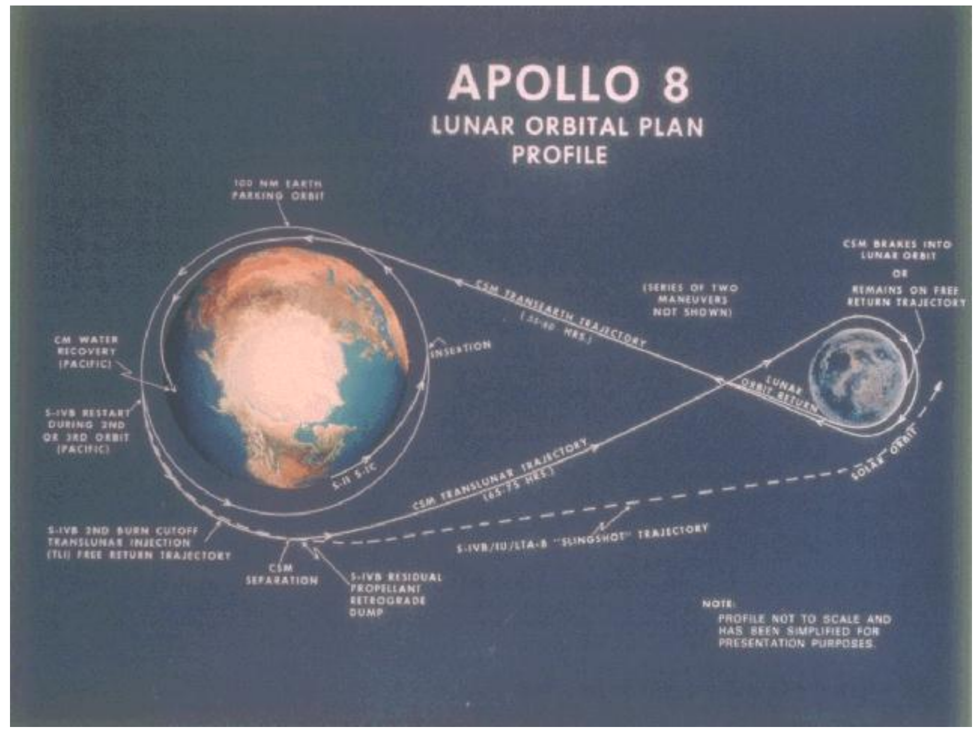
Using proper acceleration and braking: $Acc_1, Dec_1,\ldots, Acc_N, Dec_N$.

And economy cruising speed: $Vel_1, Vel_2,\ldots,Vel_N$.

1. Optimize the ***route***.
2. Optimize the ***driving strategy***.

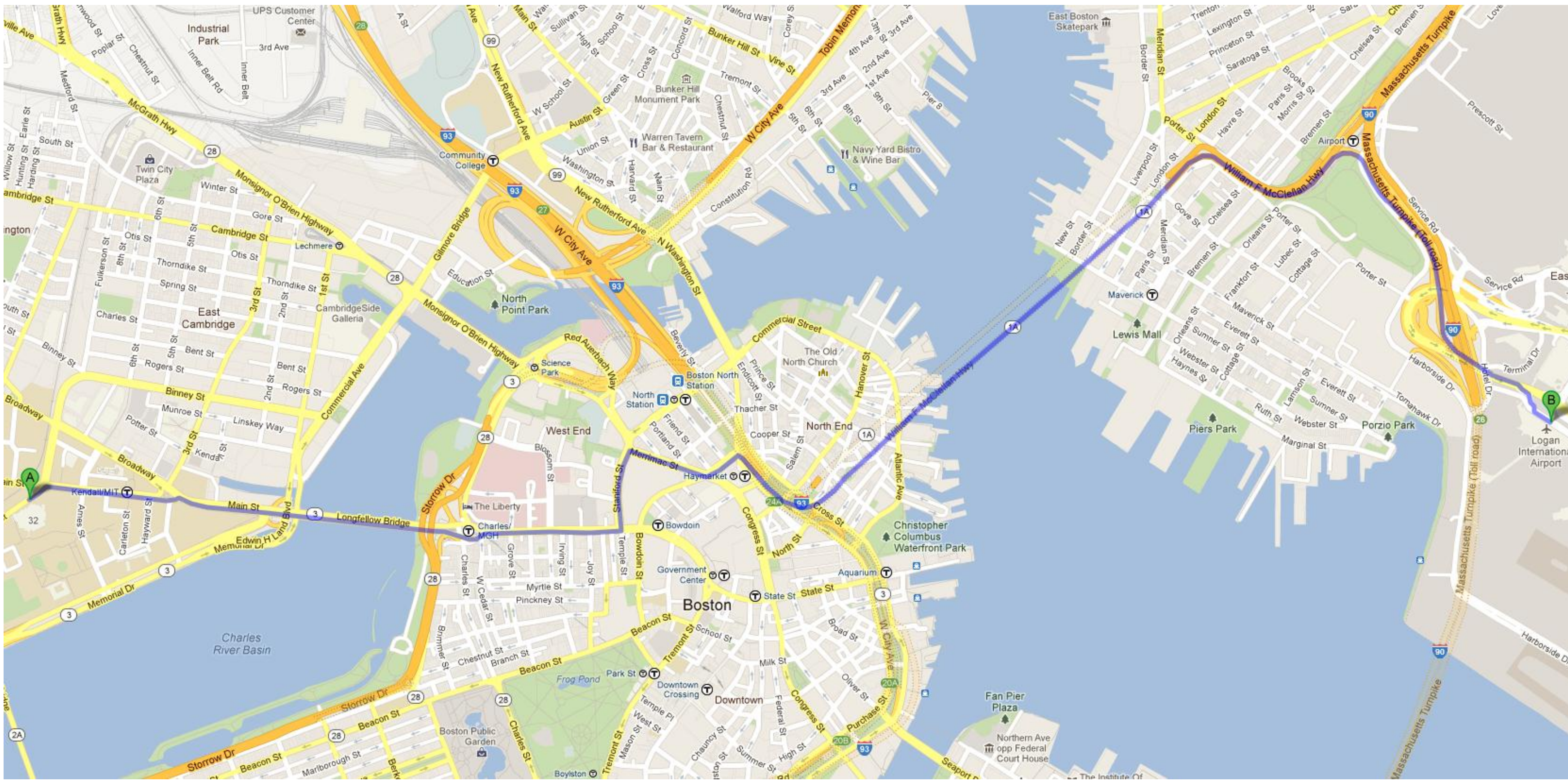
Planning for Hybrid Electric Vehicles

- Motivation & Problem Formulation
- The Best Route
- The Best Driving Style
- Examples and Summary



Vehicle routing problems

- Find the best set of road segments from the map:
  - For example, Stata Center to Logan Airport.



Vehicle routing problems

- Find the best set of road segments from the map:
  - Assume that we already know the fuel consumption and duration of each road segment.

Driving Time: $Time(Seg_k)$
e.g. 15 minutes

Fuel Consumption: $Fuel(Seg_k)$
e.g. 0.05 L

Definition

- To minimize:

$$TotalGas = \sum_{k=1\ldots N} Fuel(Seg_k)$$

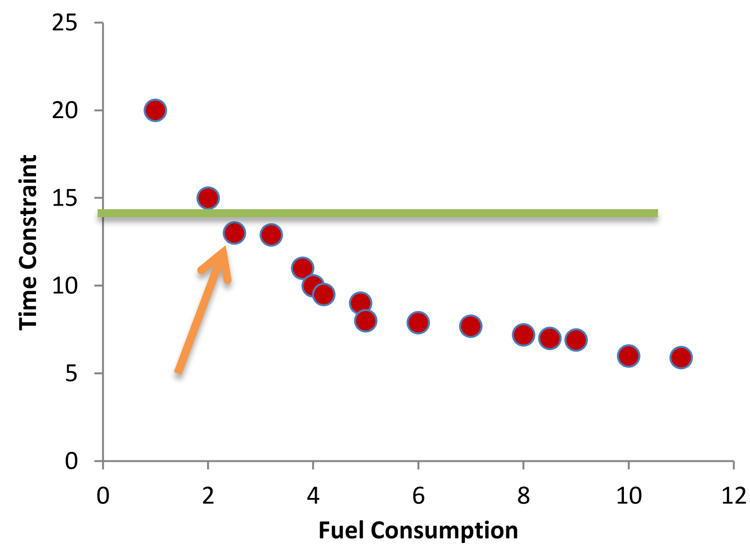
- Subject to:

$$\sum_{k=1\ldots N} Time(Seg_k) < TotalTime$$

- An optimal constraint satisfaction problem!

Exact solution

- Constrained Bellman-Ford[1] routing algorithm.
  - Treat the problem as a Multi-objective optimization.
  - Search the entire Pareto Set.
    - Using Breadth-first search and record all paths that are not dominated.

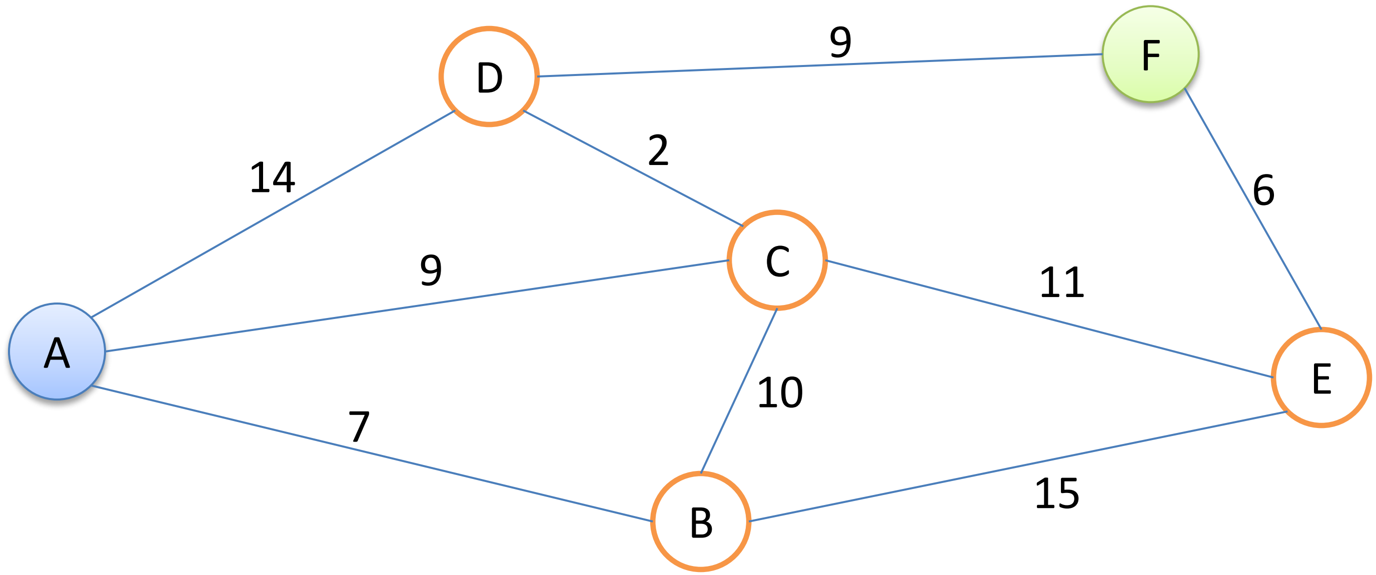


  - Exponential worst-case complexity.

[1] J. Jaffe, "Algorithms for finding path with multiple constraints", *Networks*, vol. 14, pp.95-116, 1984.

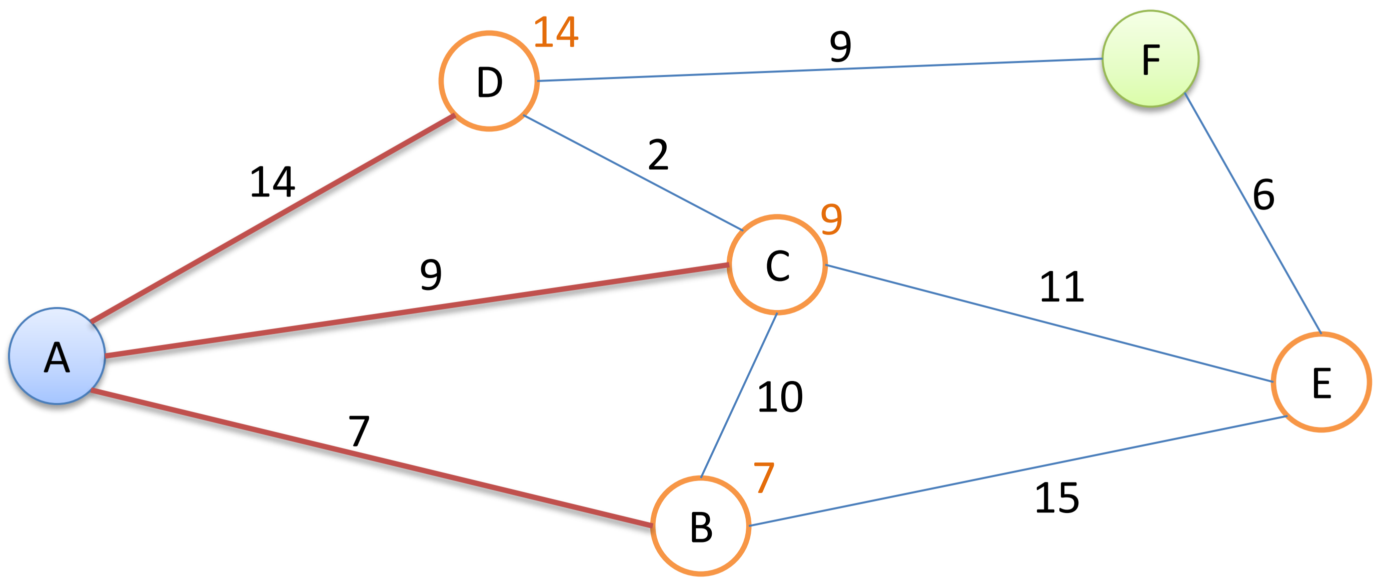
Faster methods?

- Recall: Dijkstra's algorithm.
  - Solves single-source shortest path problems.
  - Successively update the distance to vertices with newly discovered shortest route.



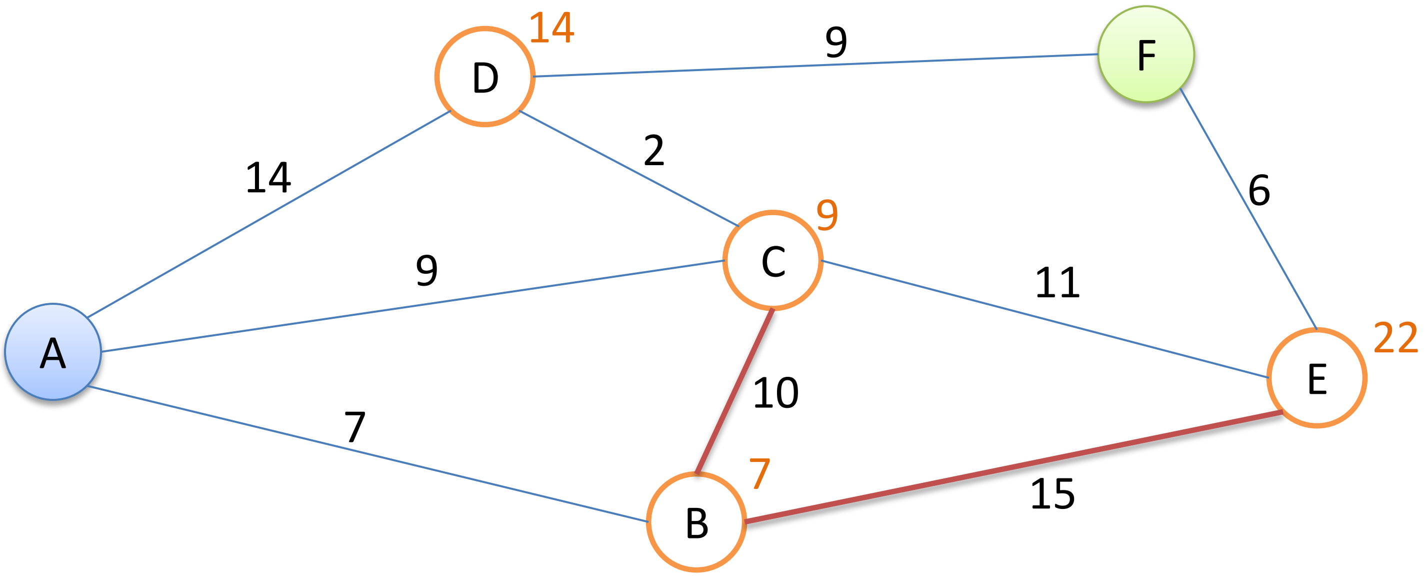
Dijkstra’s algorithm

- Recall: the Dijkstra’s algorithm.
  - Solves single-source shortest path problems.
  - Successively update the distance to vertices with newly discovered shortest route.



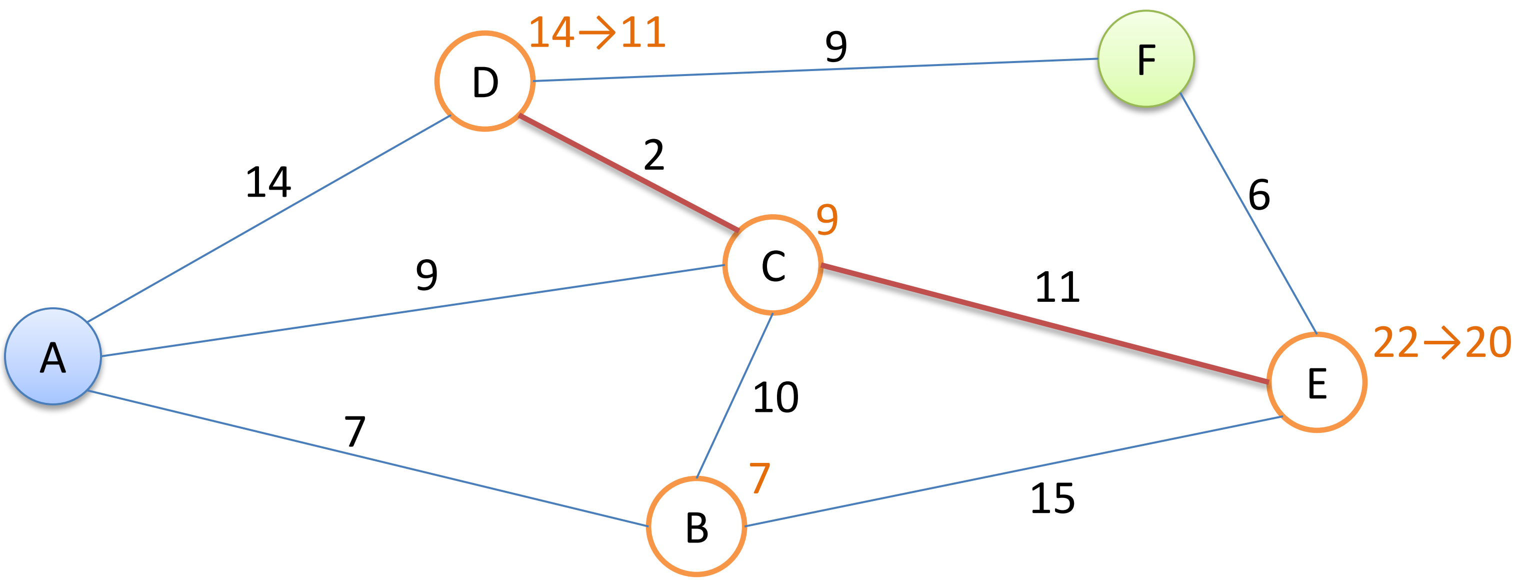
Dijkstra’s algorithm

- Recall: the Dijkstra’s algorithm.
  - Solves single-source shortest path problems.
  - Successively update the distance to vertices with newly discovered shortest route.



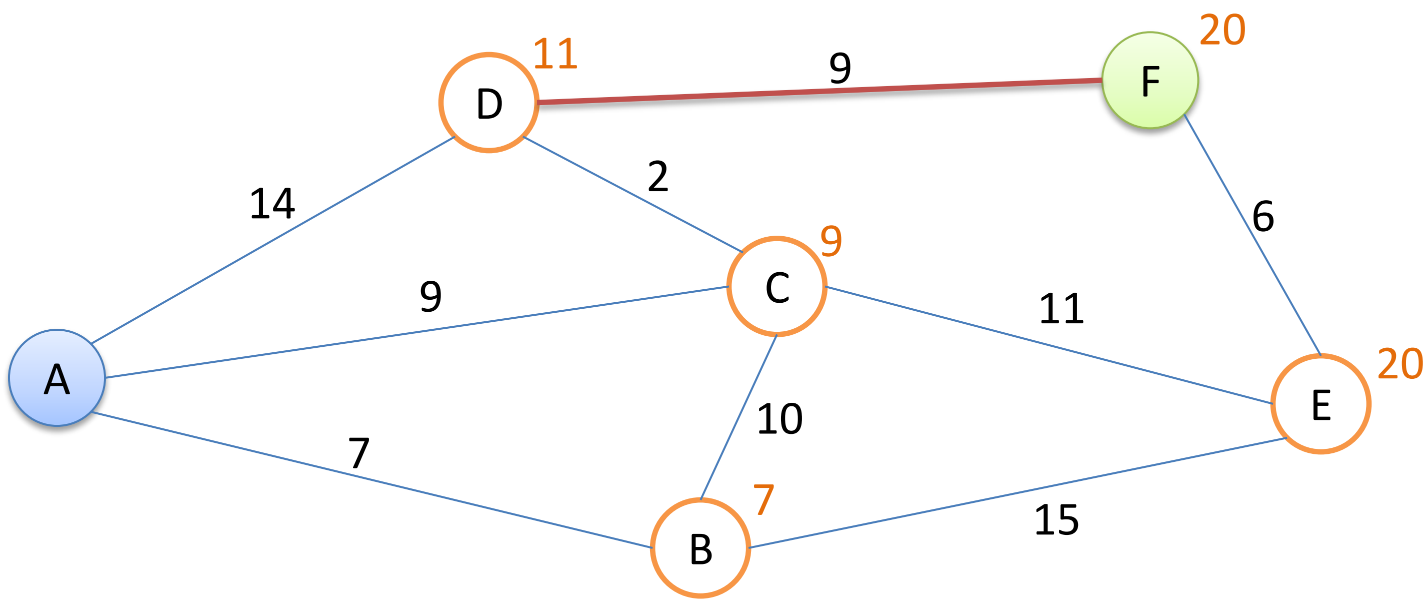
Dijkstra's algorithm

- Recall: the Dijkstra's algorithm.
  - Solves single-source shortest path problems.
  - Successively update the distance to vertices with newly discovered shortest route.



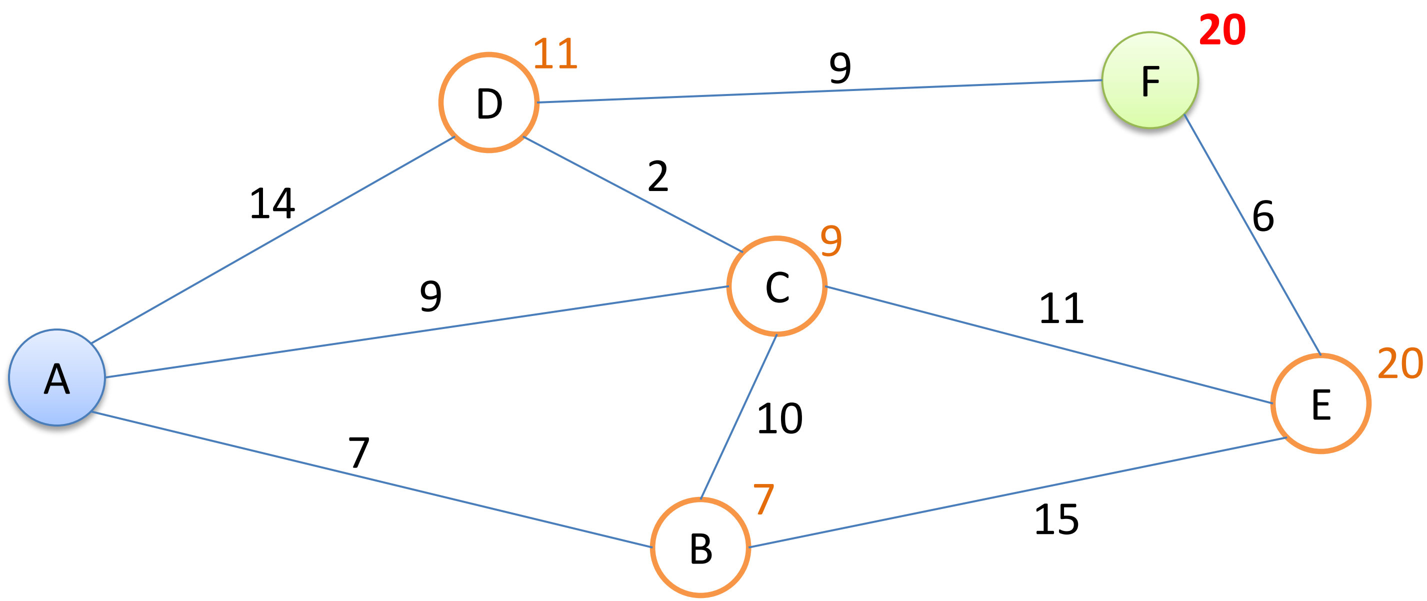
Dijkstra's algorithm

- Recall: the Dijkstra's algorithm.
  - Solves single-source shortest path problems.
  - Successively update the distance to vertices with newly discovered shortest route.



Dijkstra’s algorithm

- Recall: the Dijkstra’s algorithm.
  - Solves single-source shortest path problems.
  - Successively update the distance to vertices with newly discovered shortest route.



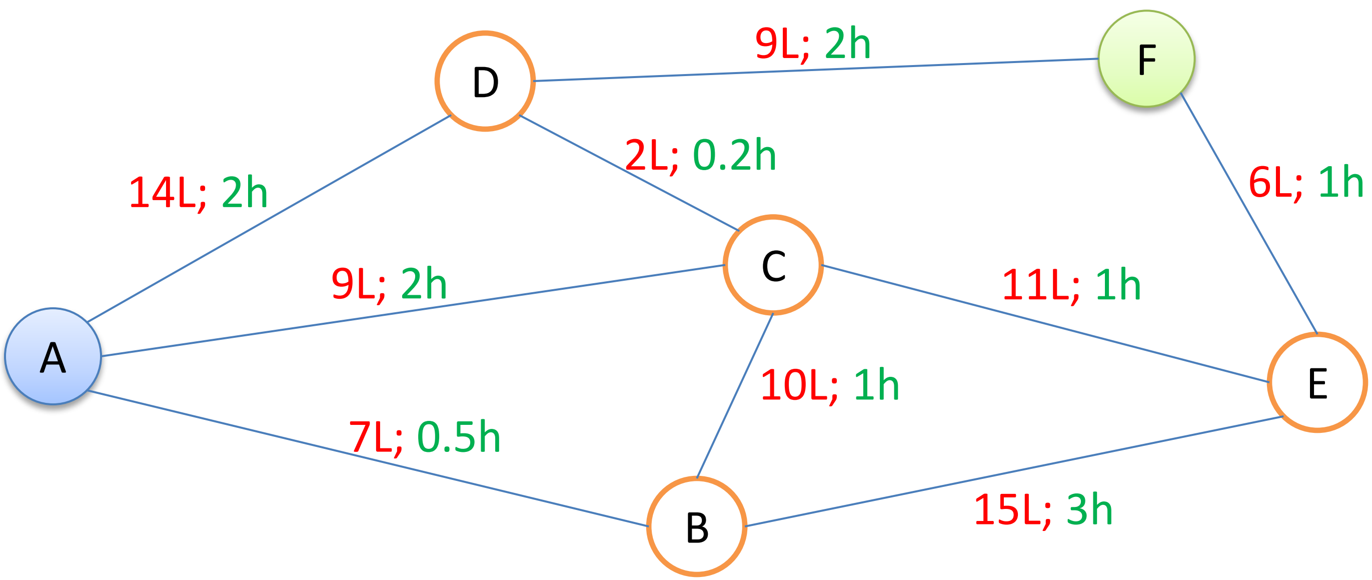
  - Runs in polynomial time: $O(|V|^2)$
  - But, it does not guarantee satisfying the timing constraint!

Backward Forward Heuristic (BFH) algorithm

- Modified from Dijkstra's algorithm.
  - Using Dijkstra's algorithm to construct shortest path trees for both fuel consumption and time, starting from the end.

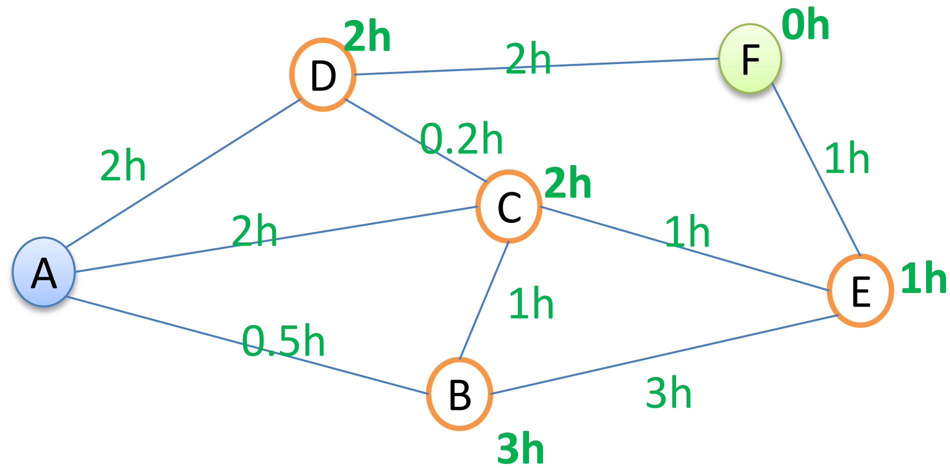
Backward Forward Heuristic (BFH) algorithm

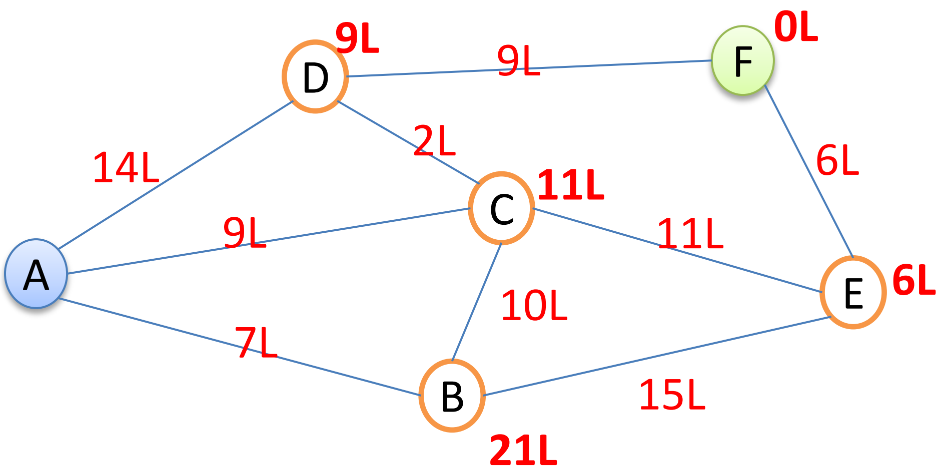
- Modified from Dijkstra's algorithm.
  - Using Dijkstra's algorithm to construct shortest path trees for both fuel consumption and time, starting from the end.



BFH: Create least-fuel tree and least-time tree

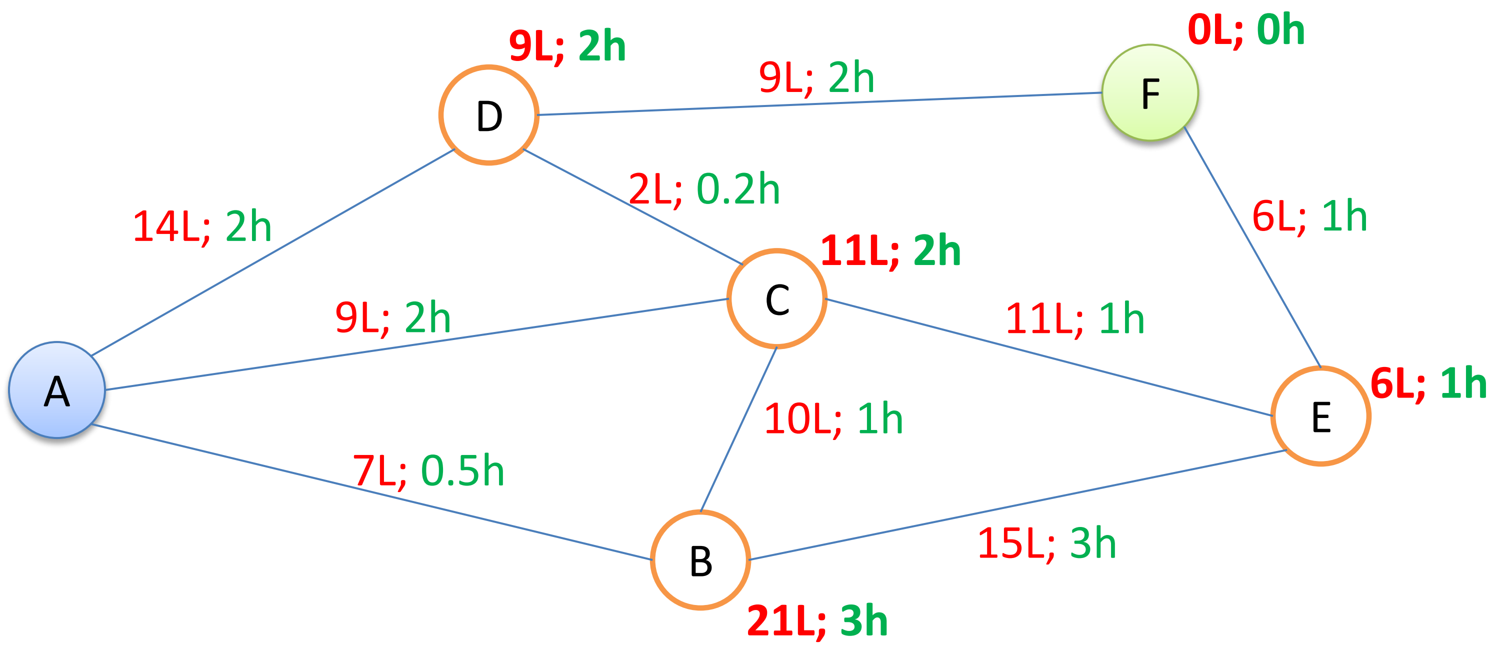
- Modified from Dijkstra's algorithm.
  - Using Dijkstra's algorithm to construct shortest path trees for both fuel consumption and time, starting from the end.





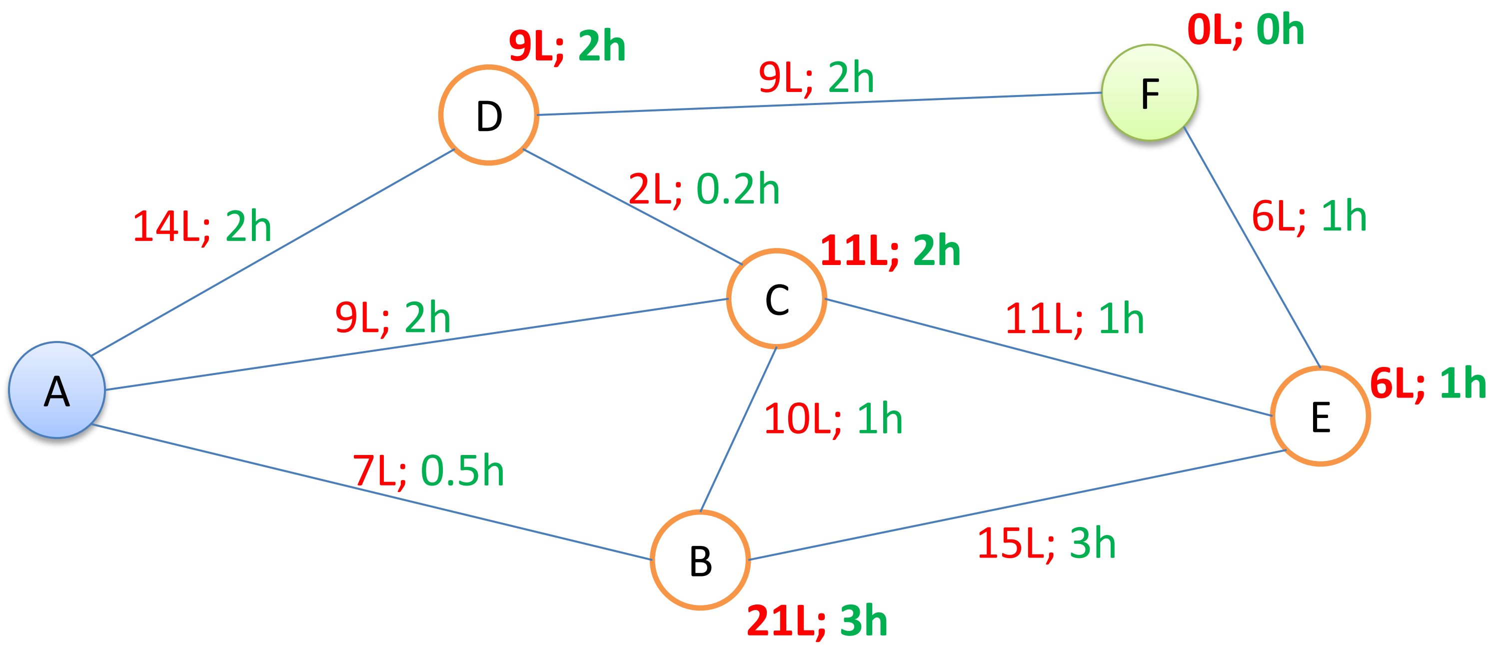
BFH: Create least-fuel tree and least-time tree

- Modified from Dijkstra's algorithm.
  - Using Dijkstra's algorithm to construct shortest path trees for both fuel consumption and time, starting from the end.



BFH: Restart from the beginning

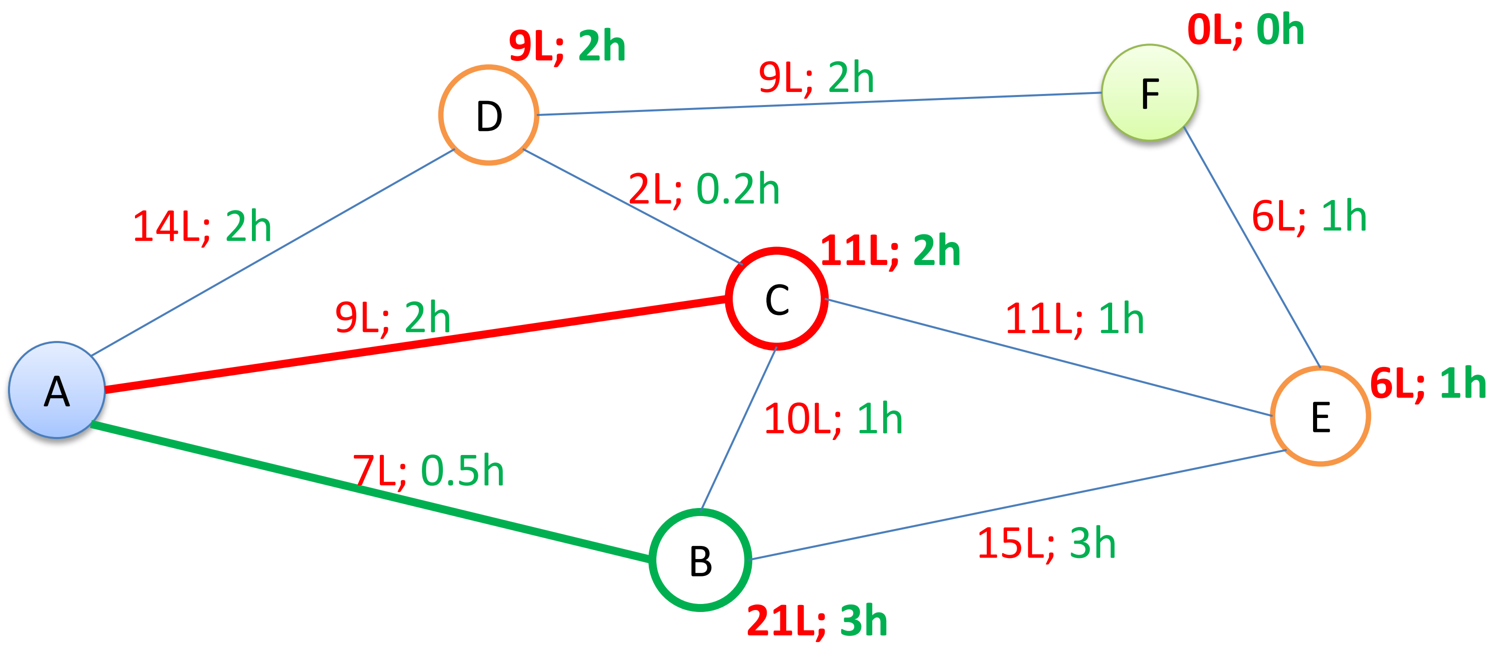
- Then start from the beginning, choose between the ***least time path*** and ***least fuel path*** to proceed.
  - If the LFP satisfies timing constraint, proceed with it.
  - Otherwise, proceed with LTP.



BFH: Propagate with timing constraint

- If the LFP satisfies timing constraint, proceed with it.
- Otherwise, proceed with LTP.

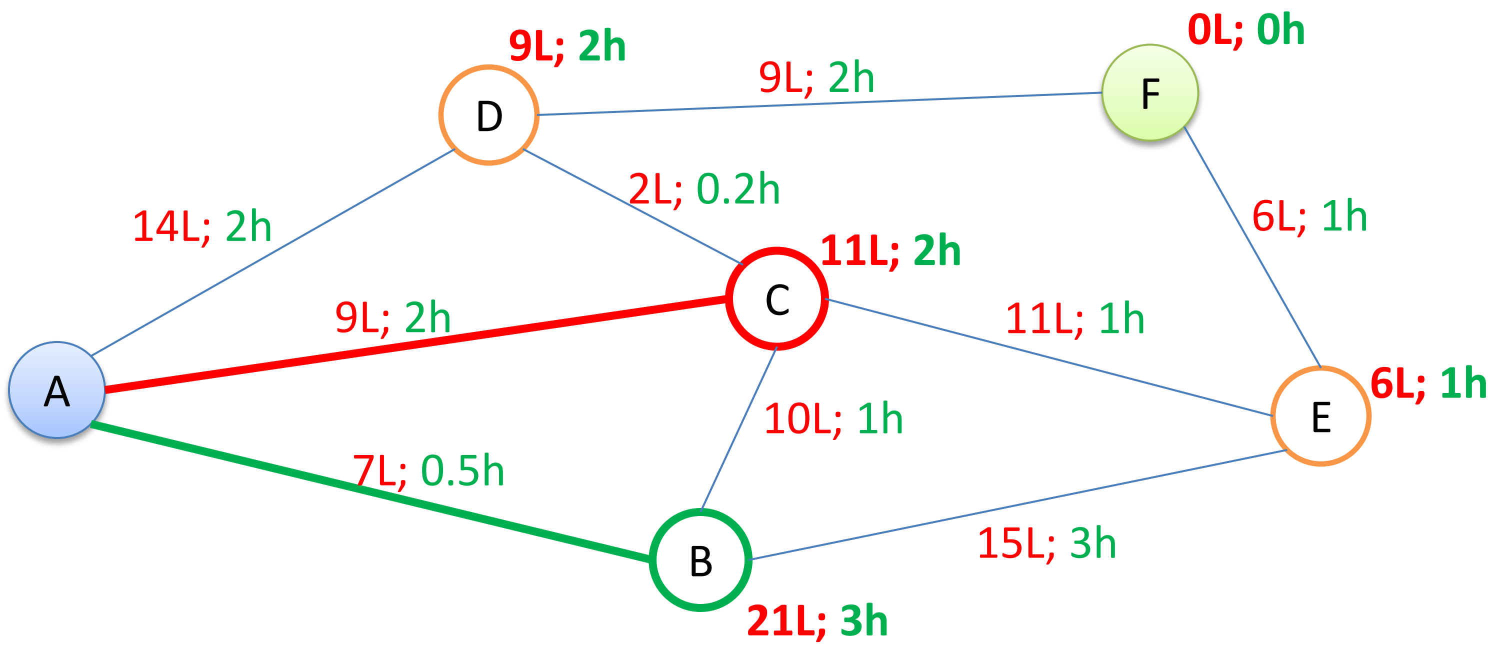
– If the timing constraint is ***4h***.



BFH: Propagate with timing constraint

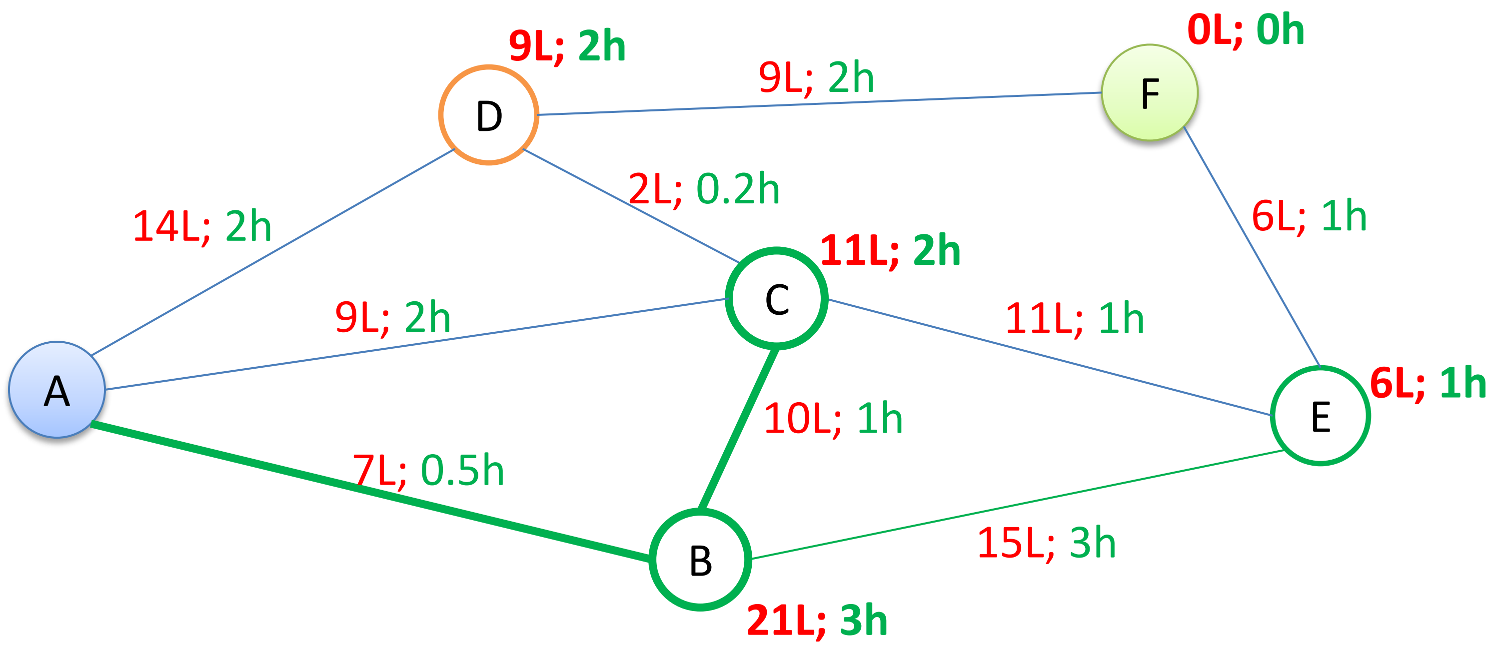
- If the LFP satisfies timing constraint, proceed with it.
- Otherwise, proceed with LTP.

– If the timing constraint is ***3.5h***.



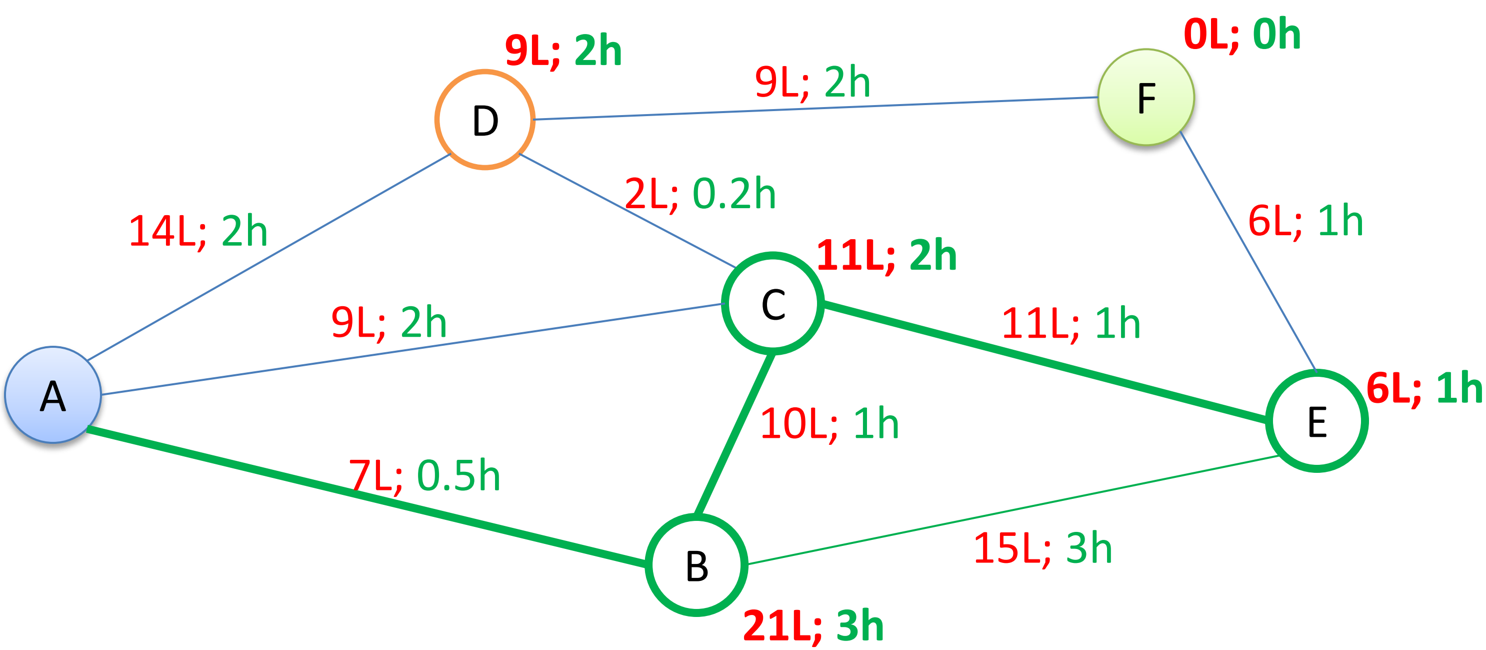
BFH: Propagate with timing constraint

– If the timing constraint is 3.5h.



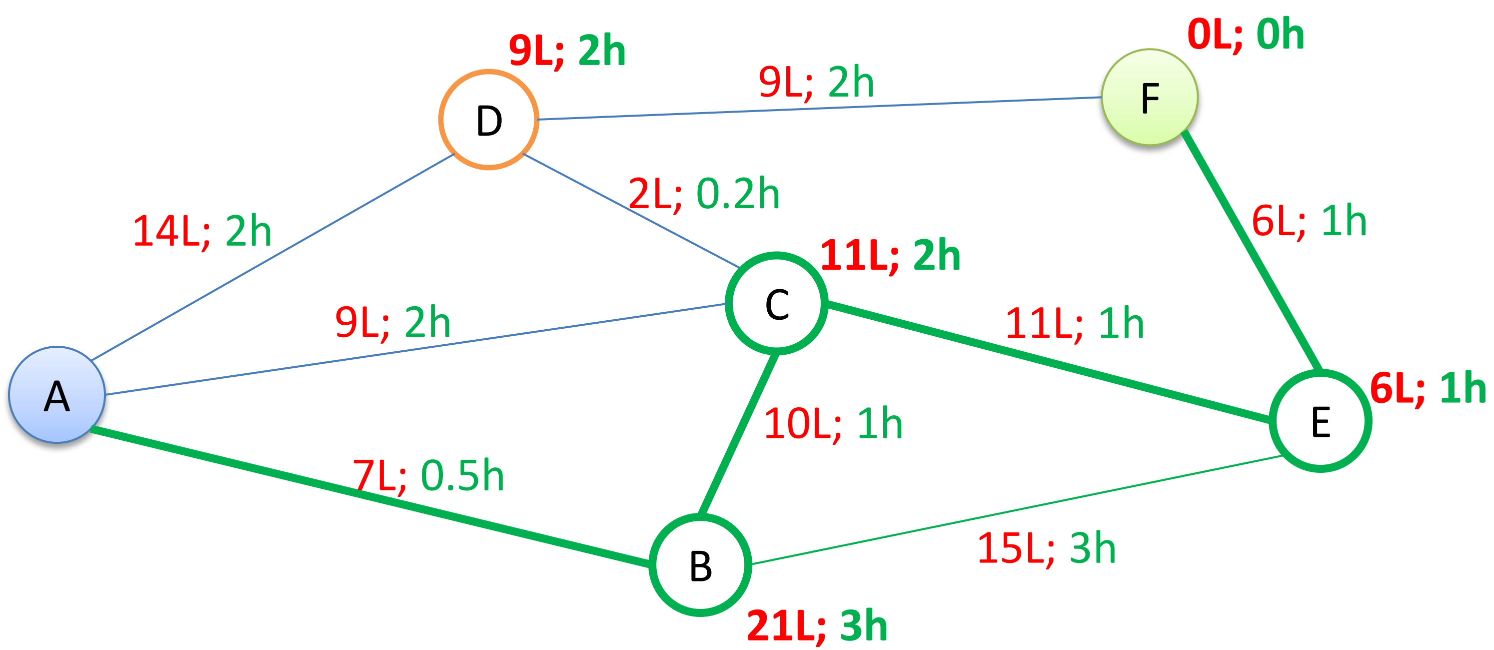
BFH: Propagate with timing constraint

- If the timing constraint is 3.5h.



BFH: Propagate with timing constraint

- If the timing constraint is 3.5h.



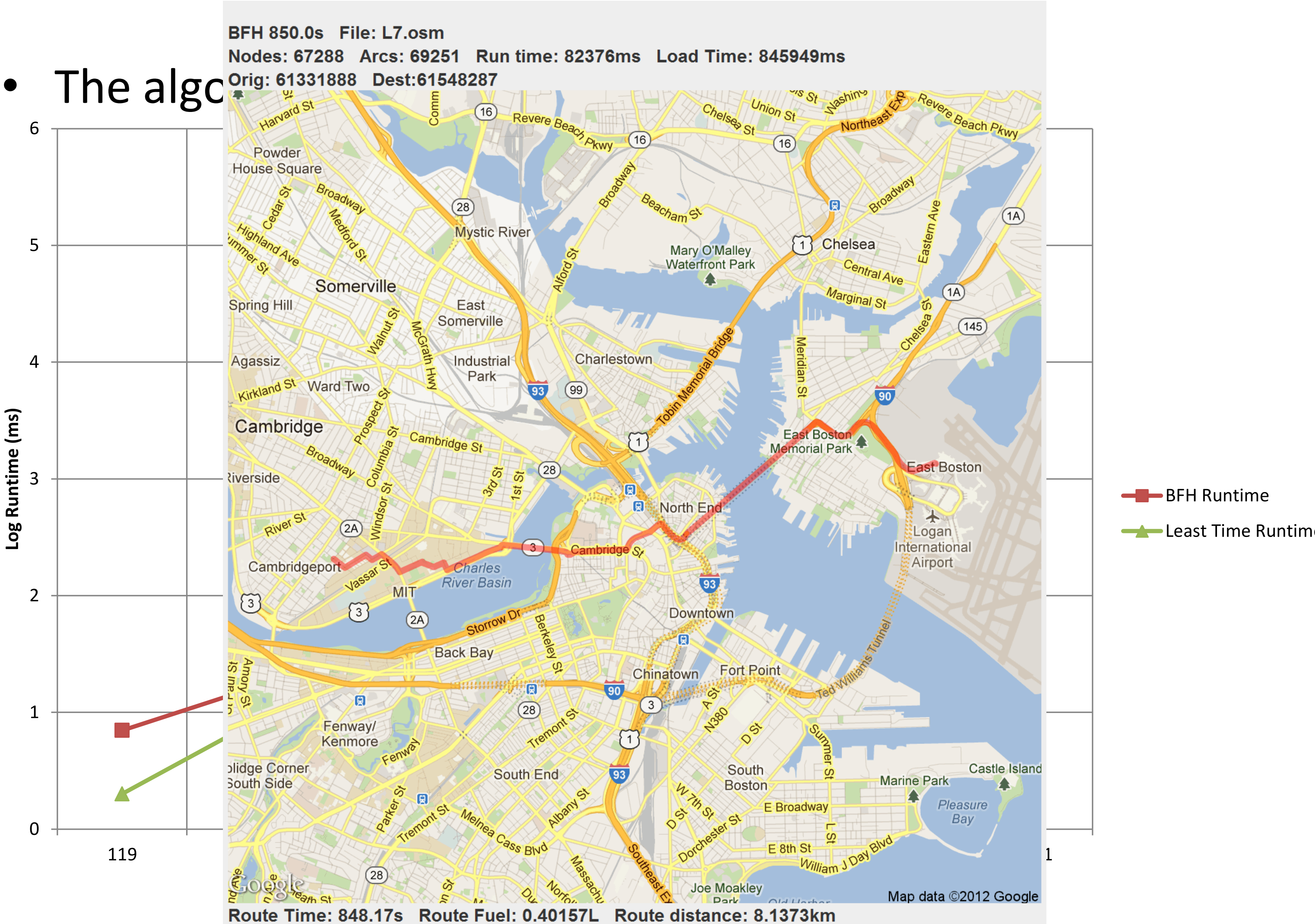
Backward Forward Heuristic (BFH) analysis

- The complexity is three times of Dijkstra’s algorithm.
- Usually generates **suboptimal** paths that satisfies timing constraint.
  - The forward procedure only makes locally optimal decision.
  - BFH paths usually less than 10% more expensive than optimal paths[1].

[1] H. F. Salama, D. S. Reeves, and Y. Viniotis, “A distributed algorithm for delay-constrained unicast routing,” in *Proc. IEEE INFOCOM’97*, Japan, pp. 84–91.

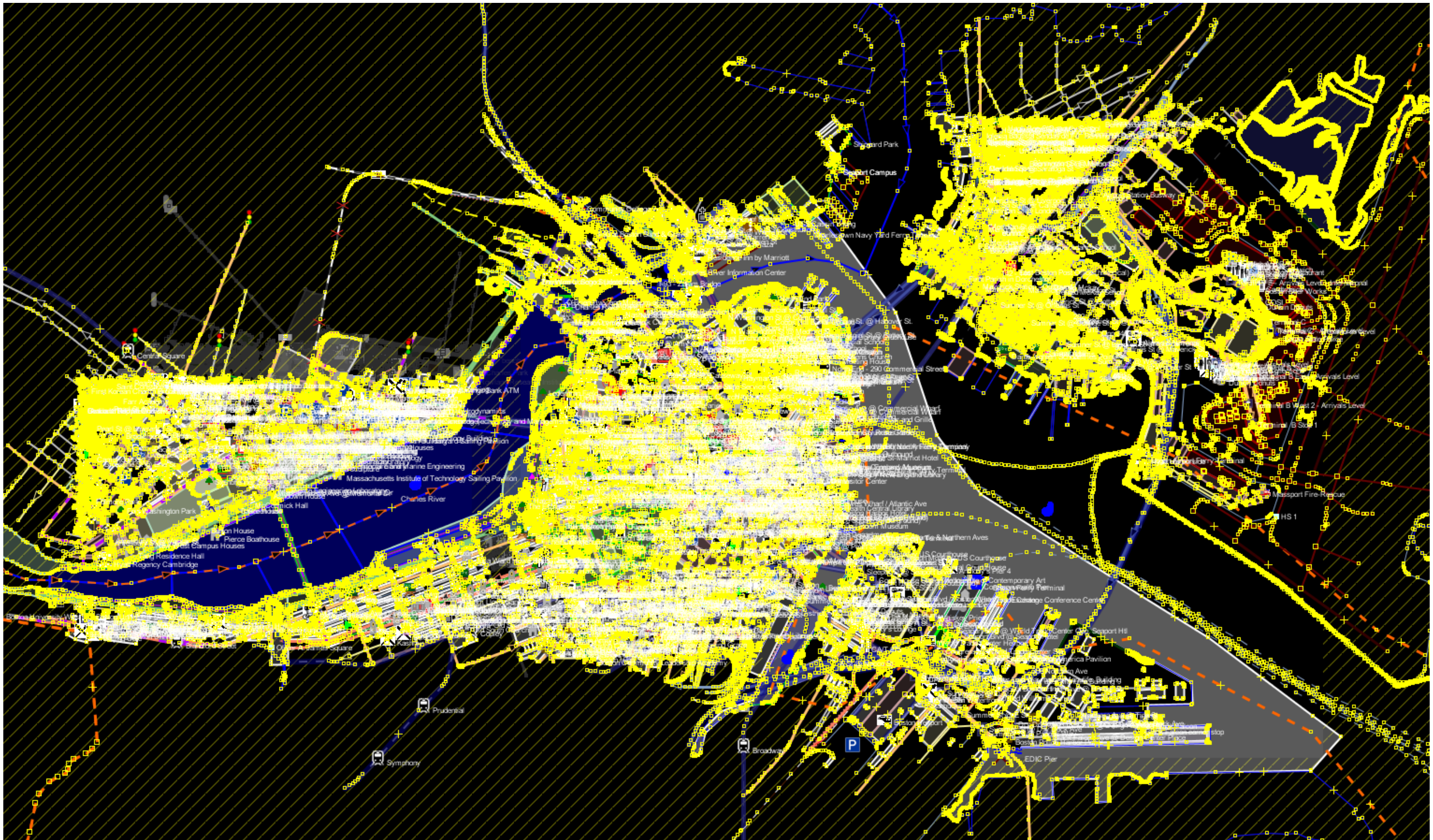
Runtime performance

- The algc



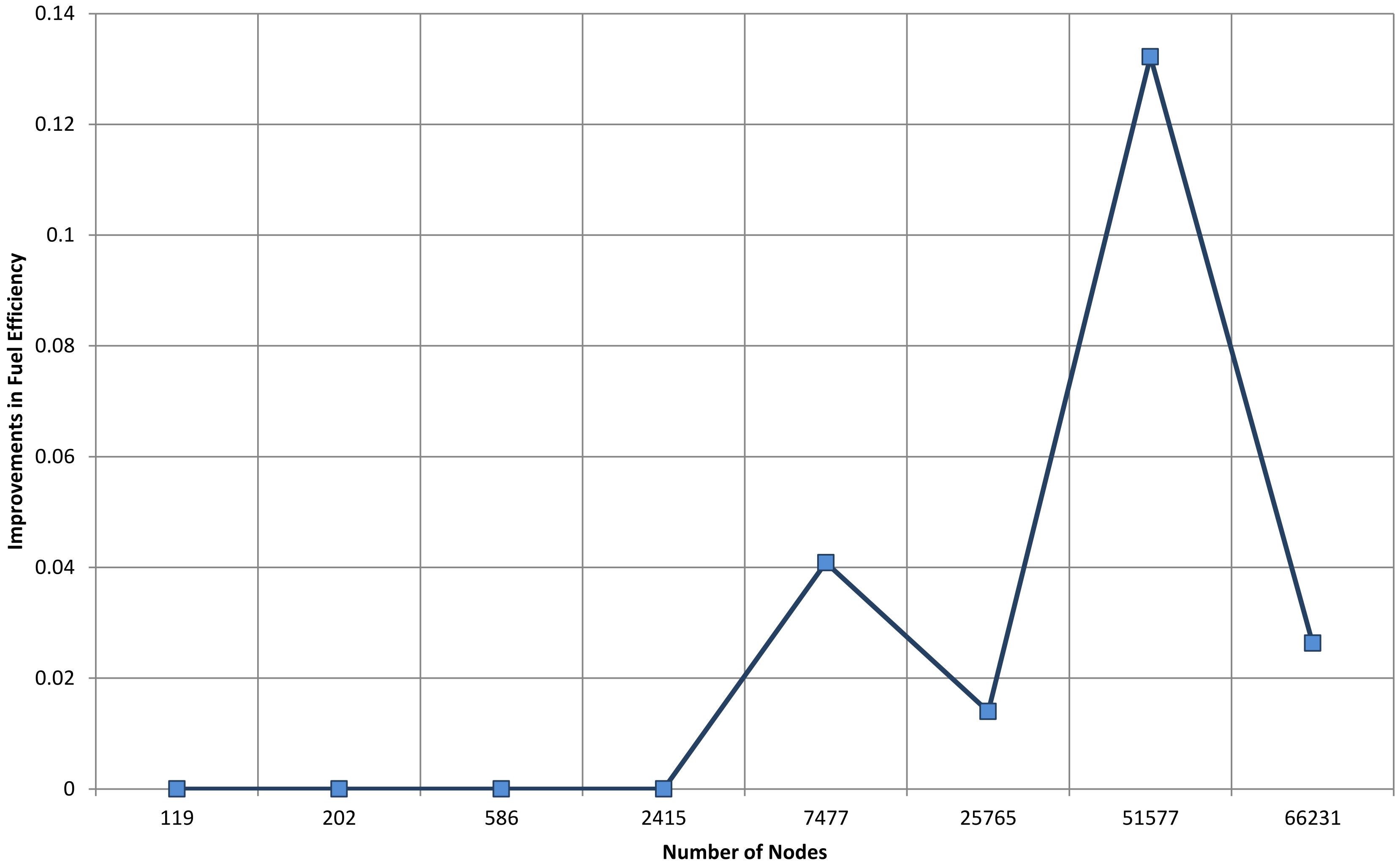
Runtime performance

- We benchmark the algorithm on various problems ranging from 100 nodes to 60000 nodes.



Quality of results

- If we want to spend 20 - 40s more on the trip:



Planning for Hybrid Electric Vehicles

- Motivation & Problem Formulation
- The Best Route
- The Best Driving Style
- Examples and Summary

Previously: Vehicle Routing Problems

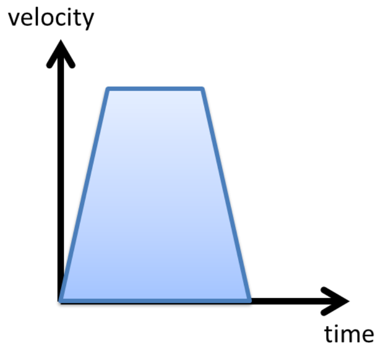
- Find the best set of road segments from the map:
  - **Assume** that we already know the fuel consumption and duration of each road segment.

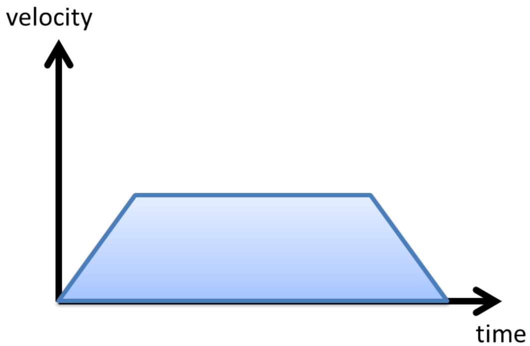
Driving Time: $\boldsymbol{Time(Seg_k)}$
e.g. 15 minutes

Fuel Consumption: $\boldsymbol{Fuel(Seg_k)}$
e.g. 0.05 L

Finding $Fuel(Seg_k)$ and $Time(Seg_k)$

- Depends on driving behavior

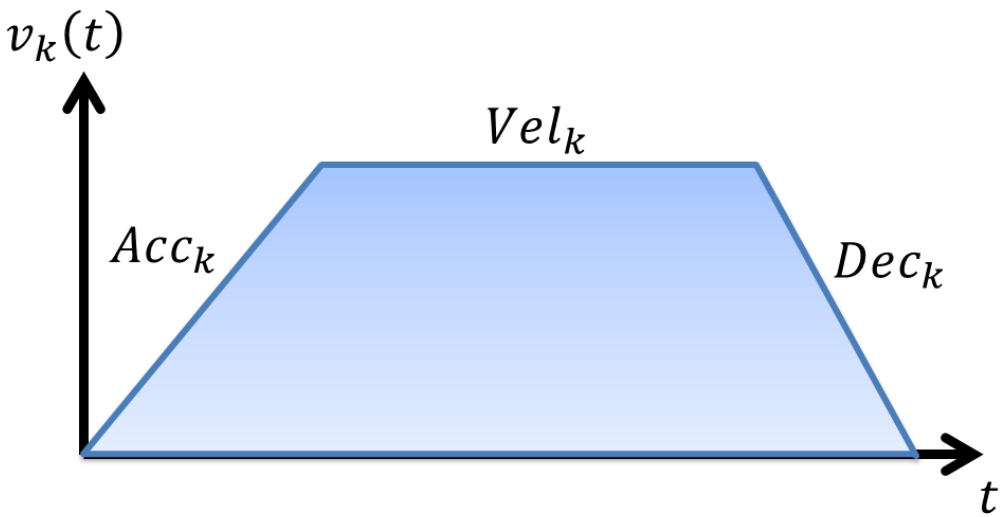




- Depends on power management
  - traditional: $P_{req} = P_{ice}$
  - hybrid: $P_{req} = P_{ice} + P_{em}$
    - additional degree of freedom in the power split
    - $P_{em} > 0$: battery **discharging**
    - $P_{em} < 0$: battery **charging**
      - ***through regenerative braking***
      - ***or ICE driving EM***

Assumptions and solution strategy

- Assume: $v_k(t) = F_{trap}(Acc_k, Vel_k, Dec_k)$.



- Strategy: optimize the fuel consumption of each segment.
  - Optimize ***a*** velocity profile (over $P_{ice,k}, P_{em,k}$)
  - Optimize ***the*** velocity profile (over Acc_k, Dec_k)

Optimize ***a*** velocity profile: problem statement

- Input: $v(\cdot)$
- Objective: $Fuel(Seg, v(\cdot)) = \min_{P_{em}(\cdot)} \Delta m_f$
- Output: $P_{ice}(\cdot)$, $P_{em}(\cdot)$, $Fuel(Seg, v(\cdot))$
- Charge-sustaining constraints:
  - $\mathrm{SOC}_{min} < \mathrm{SOC} < \mathrm{SOC}_{max}$
  - $\mathrm{SOC}(start) = \mathrm{SOC}(end) = \frac{1}{2}(\mathrm{SOC}_{min} + \mathrm{SOC}_{max})$

Optimize *a* velocity profile: modeling power flow

- Mechanical power:
  - $P_{req} = P_{ice} + P_{em}$
  - $P_{req} = F_{req} v = \left(C_D v^2 + M \frac{dv}{dt} \right) v$
- Fuel power:
  - $\frac{dm_f}{dt} = \frac{1}{H_f} \cdot \frac{P_{ice}}{\eta_{ice}(P_{ice})}$
- Electric power:
  - $\frac{d\mathrm{SOC}}{dt} = -\frac{1}{Q_{max} V_{batt}} \cdot \frac{P_{em}}{\eta_{em}(P_{em})}$

Optimize *a* velocity profile: solution by ECMS

- Fuel is to gold as battery charge is to cash.
  - fuel tank = personal stash of gold
  - charge battery = put gold/cash into the bank
  - discharge battery = take cash out of the bank

- Goal: retain as much gold as possible.

- Strategy: express everything in terms of gold.
  - Equivalent Consumption Minimization Strategy (ECMS)

Source: L. Serrao, S. Onori, and G. Rizzoni, "A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles," *Journal of Dynamic Systems, Measurement, and Control, Vol. 133, May 2011*.

Optimize *a* velocity profile: ECMS algorithm

Pontryagin's minimum principle:

- if $P_{em}(\cdot)$ is globally optimal

i.e. minimizes $\Delta m_f = \int \dot{m}_f \, dt$,

- then at each time t, $P_{em}(t)$ is locally optimal w.r.t.

$$\dot{m}_{f,e}(t) = \dot{m}_f(t) + \lambda(t)\dot{m}_{b,e}(t).$$

CURRENCY EXCHANGE

- $\dot{m}_{b,e}(t) = \frac{1}{H_f} \cdot \frac{P_{em}}{\eta_{em}(P_{em})}$

Assume λ is constant.
(actually depends on internal battery parameters)

Source: L. Serrao, S. Onori, and G. Rizzoni, "A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles," *Journal of Dynamic Systems, Measurement, and Control, Vol. 133, May 2011*.

Optimize *a* velocity profile: ECMS algorithm

Minimize:

$$\dot{m}_{f,e}(t) = \dot{m}_f(t) + \lambda\,\dot{m}_{b,e}(t).$$

- Algorithm:
  1. Select λ.
     - Optimize $P_{em}(t)$ at all times t.
  2. Simulate $P_{em}(\cdot)$ to determine $\mathrm{SOC}(end)$.
  3. Iterate on 1 and 2, using the Shooting Method to ensure $\mathrm{SOC}(start) = \mathrm{SOC}(end)$.

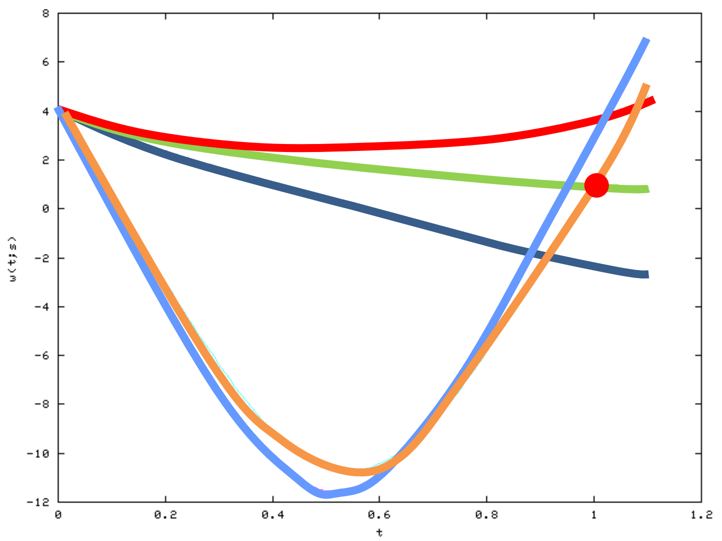
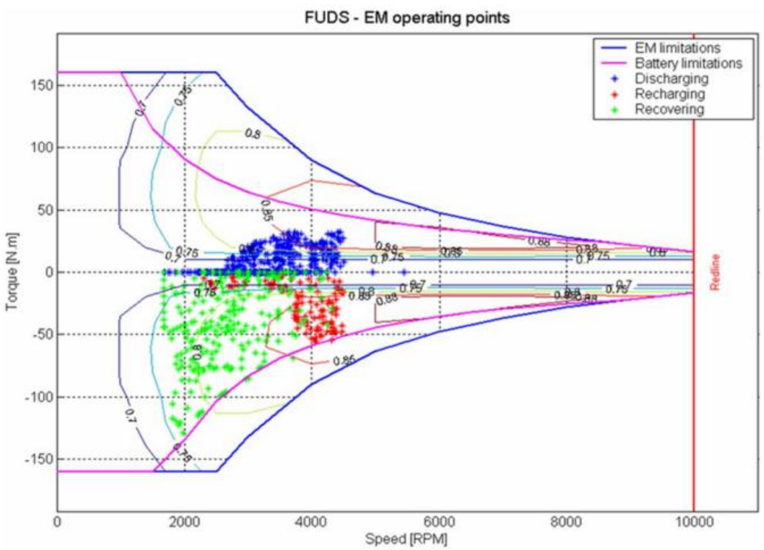


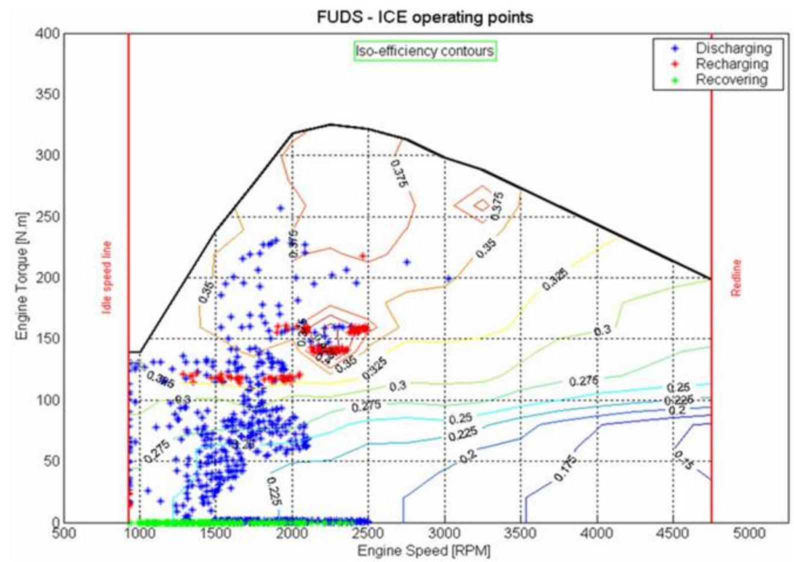
Illustration of the Shooting Method

Source: http://en.wikipedia.org/wiki/Shooting_method

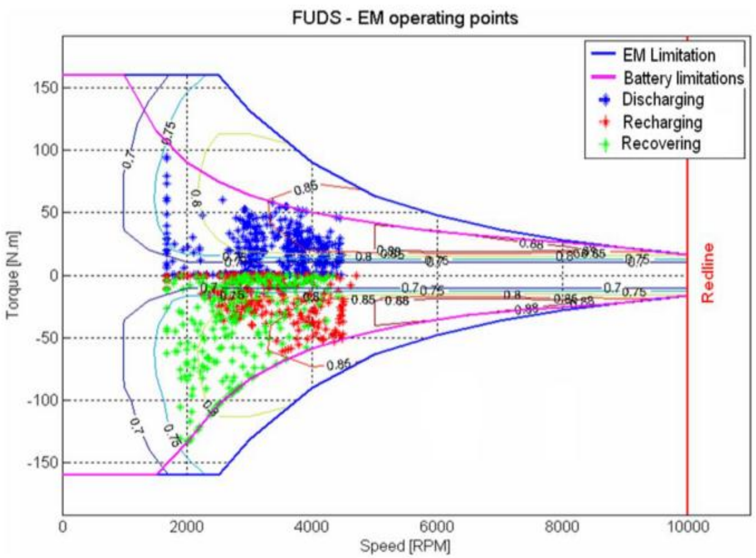
Optimize *a* velocity profile: ECMS validation

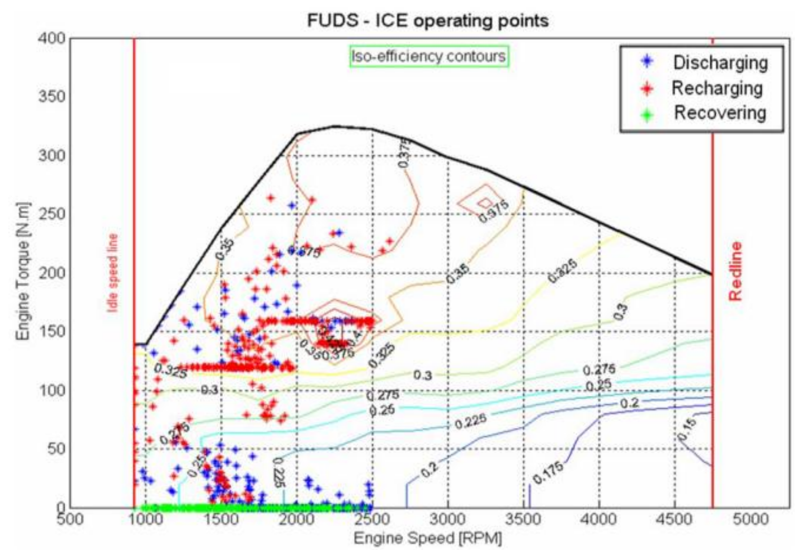
Dynamic Programming





ECMS

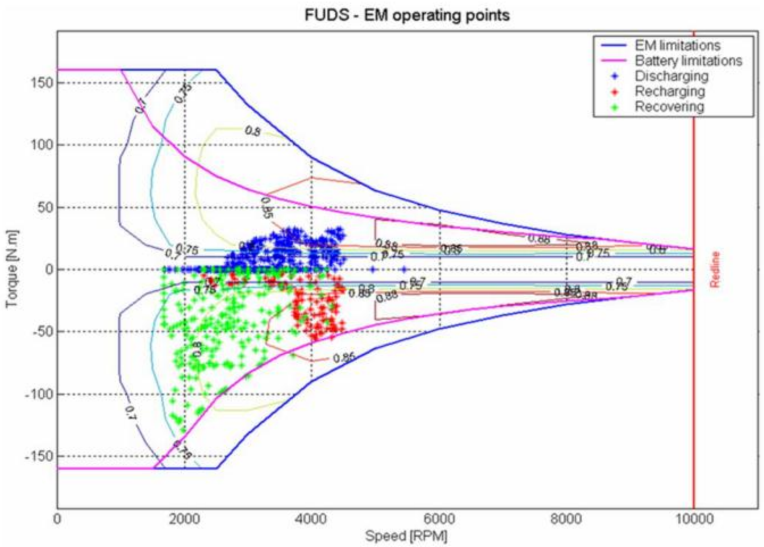


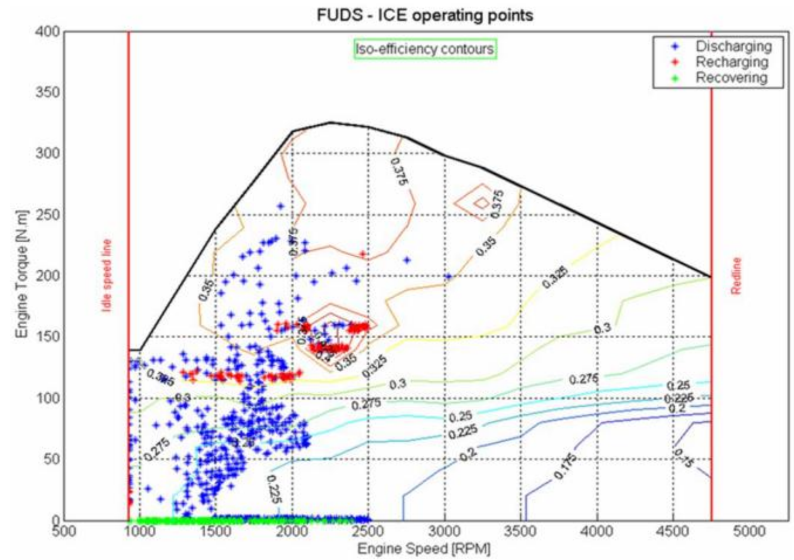


Source: P. Pisu and G. Rizzoni, “A Comparative Study of Supervisory Control Strategies for Hybrid Electric Vehicles,” *IEEE Transactions on Control Systems Technology, Vol. 15, No. 3, May 2007*.

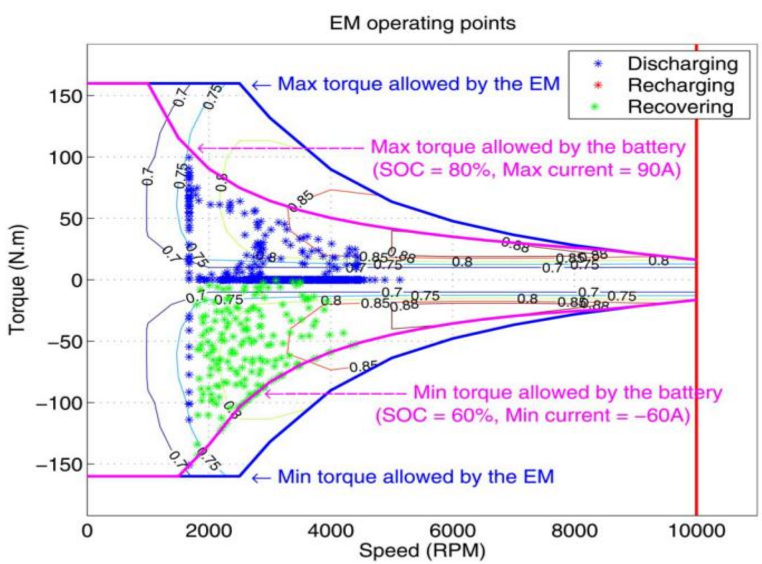
Optimize *a* velocity profile: ECMS validation

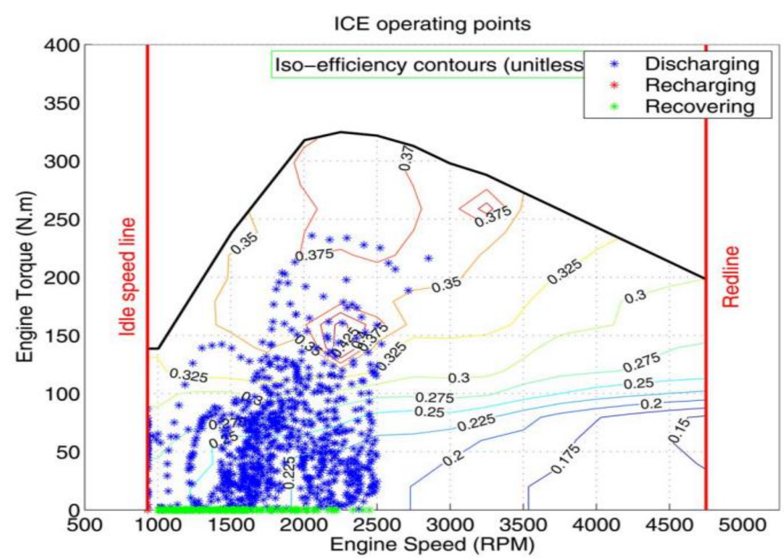
Dynamic Programming





Rule-based Finite-State Machine (FSM)





Source: P. Pisu and G. Rizzoni, "A Comparative Study of Supervisory Control Strategies for Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology, Vol. 15, No. 3, May 2007*.

Optimize *a* velocity profile: ECMS validation

Driving Cycle	Pure thermal mpg	DP mpg	DP Improv.	ECMS opt mpg	ECMS opt Improv.
FUDS	22.1	25.7	16.4%	25.7	16.3%
FHDS	24.8	26.0	4.9%	25.8	4.1%
ECE	20.8	24.5	18.2%	24.5	18.0%
EUDC	23.3	24.8	6.3%	24.7	6.2%
NEDC	22.2	24.5	10.7%	24.5	10.7%
JP1015	21.0	25.2	20.1%	25.1	19.8%

Source: C. Musardo, G. Rizzoni, and B. Staccia, "A-ECMS: An adaptive Algorithm for Hybrid Elecric Vehicle Energy Management," *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

Cycle	Pure Diesel	FSM	A-ECMS	$\mathcal{H}_\infty$	DP
FUDS	21	21(0.76)†	26 (0.7)	21 (0.72)	26* (0.7)
FHDS	25	24(0.73)	26(0.66)	24(0.72)	26*(0.7)

* Results based on simplified model.

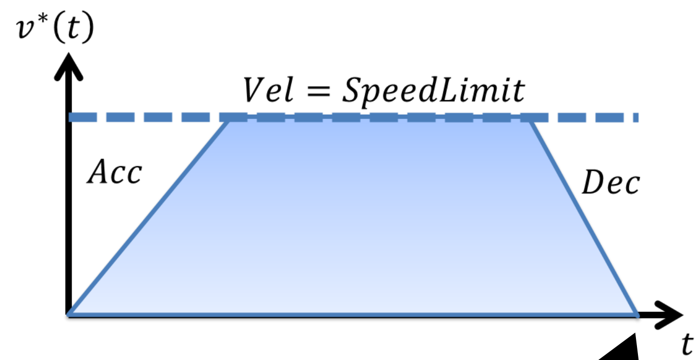
† Final SOC between parentheses.

Source: P. Pisu and G. Rizzoni, "A Comparative Study of Supervisory Control Strategies for Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology, Vol. 15, No. 3, May 2007*.

Optimize *the* velocity profile

- Input: $Dist, SpeedLimit$

- Objective: $Fuel(Seg) = \min_{Acc,Dec} Fuel(Seg, v(\cdot))$
- Subject to constraints:
  - $0 < \boldsymbol{Acc} < MaxAcc, \quad 0 < \boldsymbol{Dec} < MaxBrake$
  - $\boldsymbol{Vel} = SpeedLimit$
  - $v(\cdot) = F_{trap}(Acc, Vel, Dec)$
  - $\int v(t)\, dt = Dist$



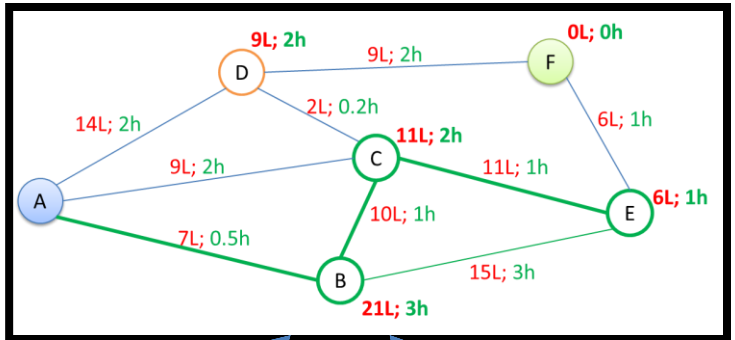
- Output: $v^*(\cdot)$

$$Fuel(Seg)$$

$$Time(Seg) = \min_{\substack{v^*(t)=0 \\ t>0}} t$$

Summary of algorithms

BFH



$v^*(t)$ optimizer

ECMS

$Time(Seg)$

$Fuel(Seg)$

$v(t)$

$P_{req} = P_{ice} + P_{em}$

Planning for Hybrid Electric Vehicles

- Motivation & Problem Formulation
- The Best Route
- The Best Driving Style
- Examples and Summary

Recall: The Driving Problem

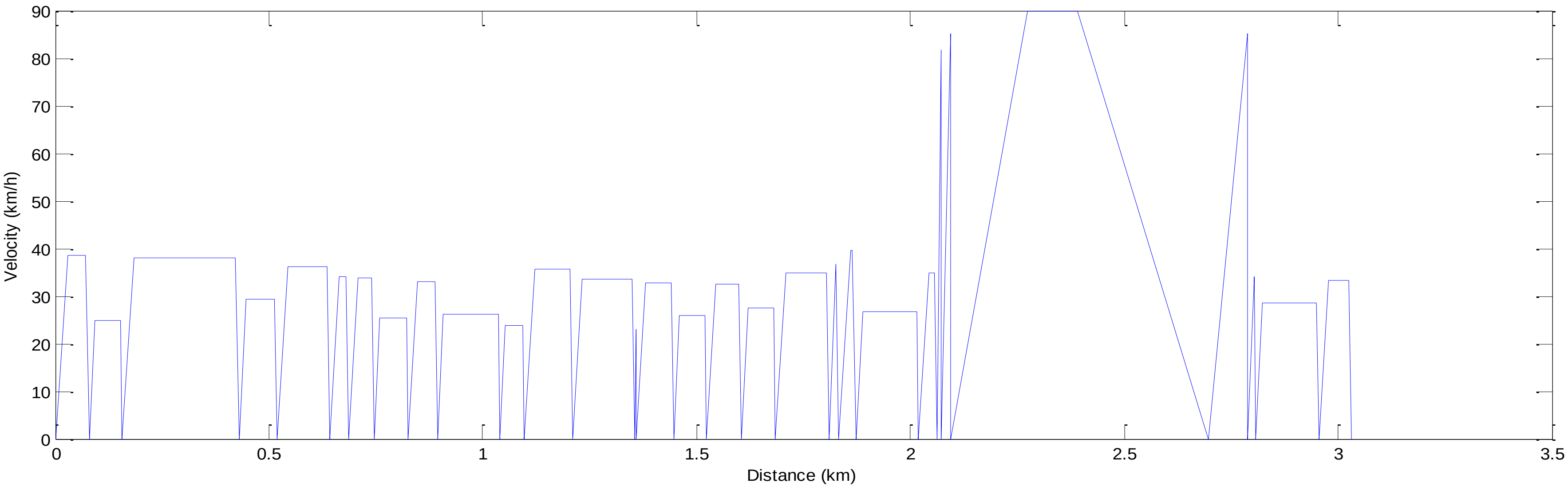
- We would like to get to the destination with minimum fuel consumption through:
  - Selecting the best route.
  - Optimizing the driving style.

- In addition, we want to satisfy the timing constraints.
  - Drive to the Logan airport from the Stata Center in 15 minutes.

Results: hybrid car

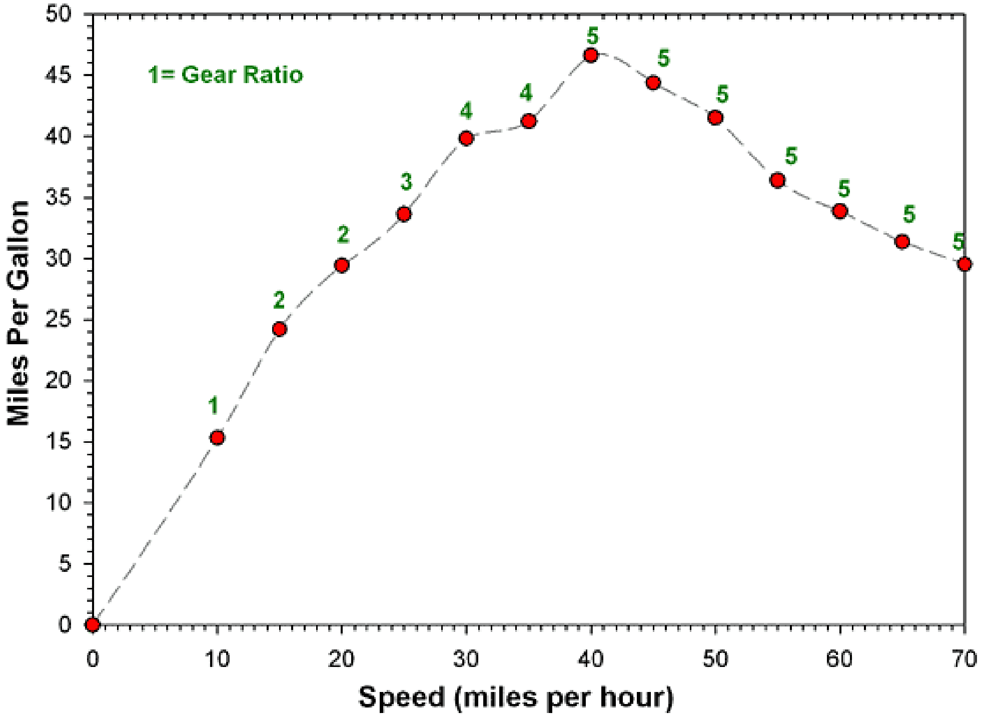
- Path selected by the algorithm.

- Driving profile.



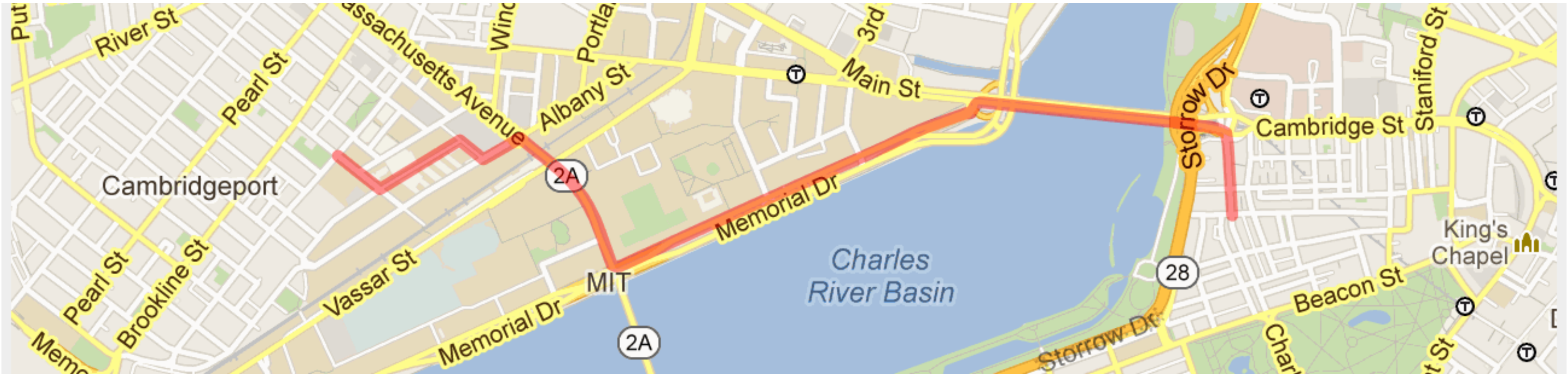
Non-hybrid cars

- Volkswagen Golf

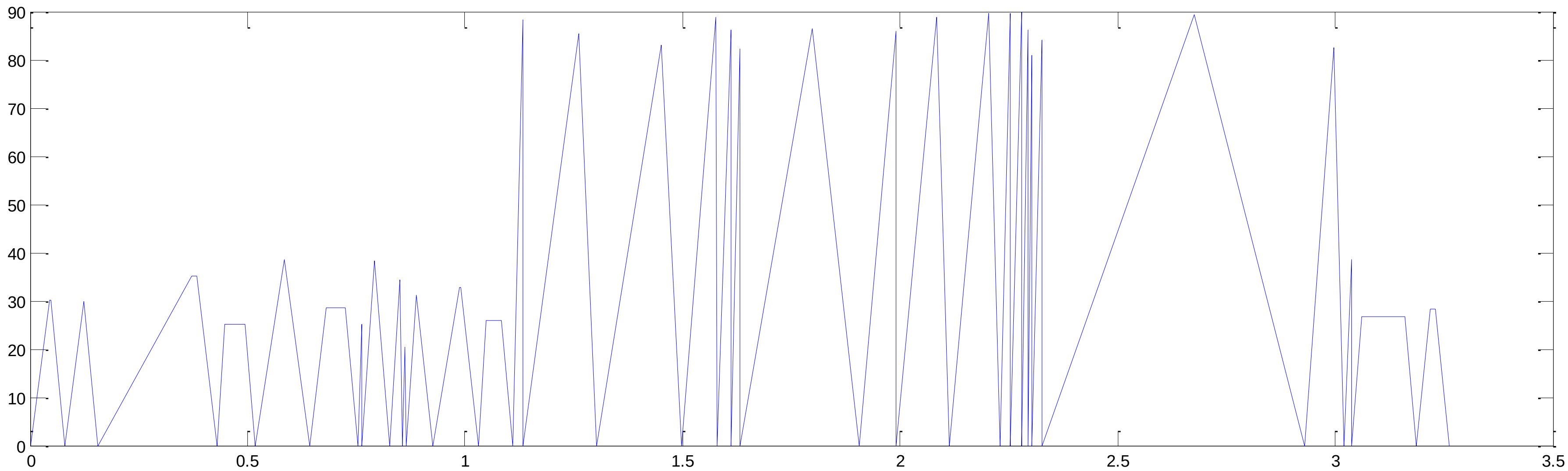


Non-hybrid cars

- Path selected by the algorithm.



- Driving profile.



Summary

- Problems solved:
  - How to select the path that minimize fuel cost, while satisfying timing constraints.
  - The best driving style/modes that minimize the fuel consumption.
- Remarks
  - The modeling and optimization of energy systems can be very hard.
  - Suboptimal approaches may save a lot of time.
- Hybrid cars (and ECO driving modes) do help save fuel!

Byproduct

- Fuel efficient path generator:
  - Compute a path for your trip based on OpenStreetMap Database, then project on Google Map.
  - Satisfies your timing constraint while minimizing fuel consumption.
  - Available for Prius and Golf, in Cambridge and Boston area.
  - http://people.csail.mit.edu/yupeng/files/16S949.zip