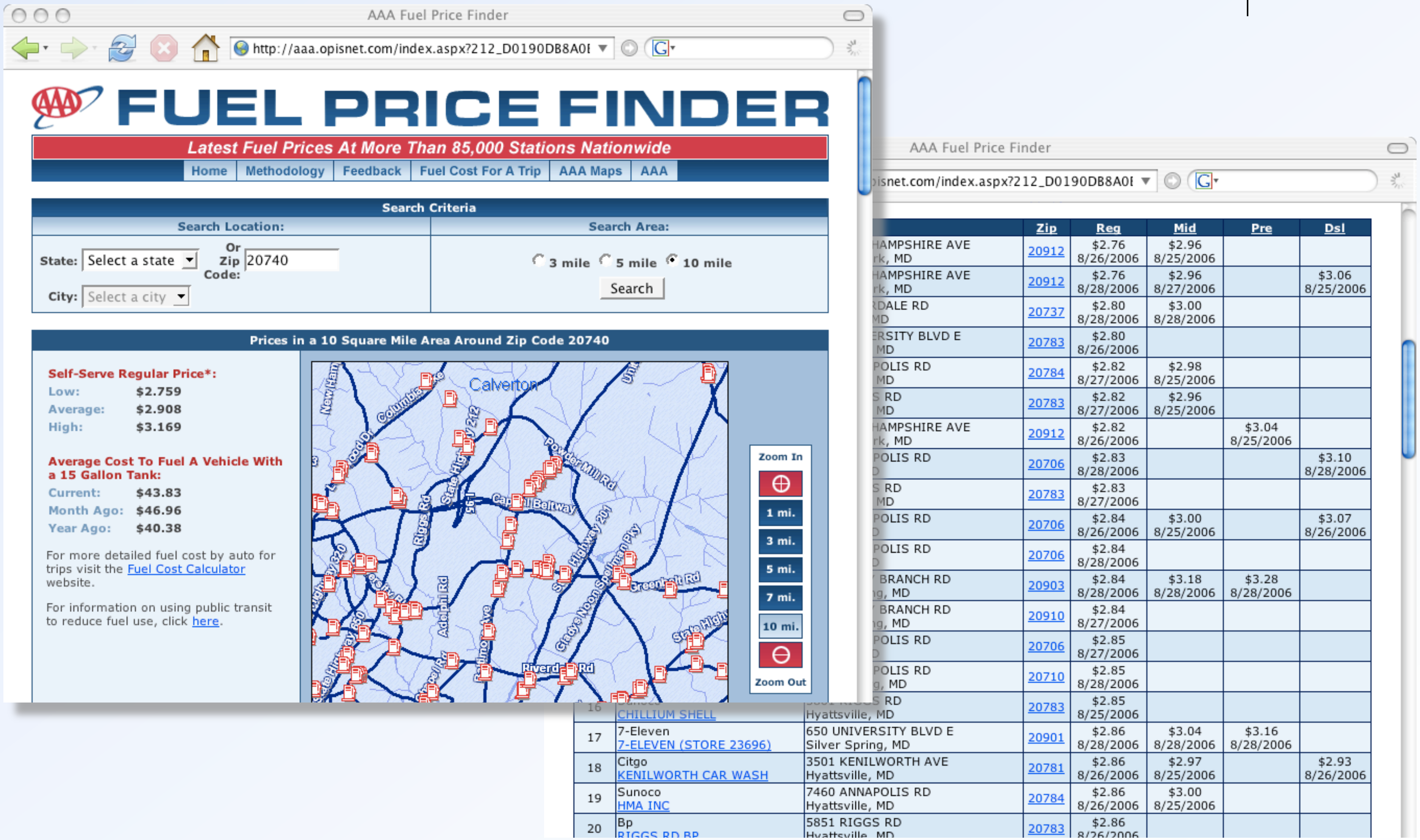
The Gas Station Problem

Samir Khuller
Azarakhsh Malekian
UMD
Julian Mestre UMD → MPI

General Description

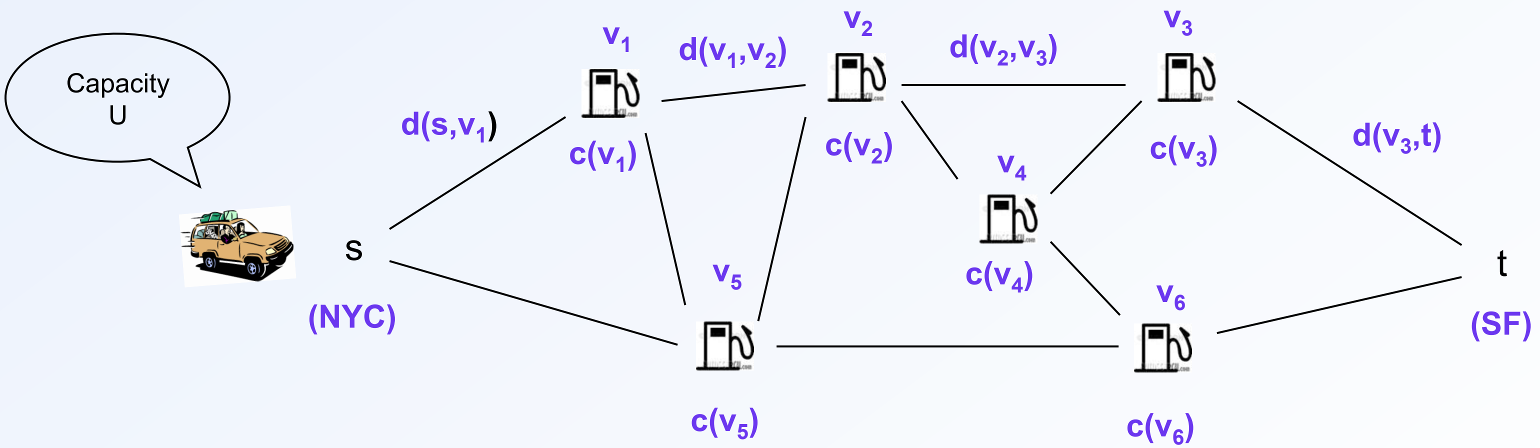
- Suppose you want to go on a road trip across the US. You start from New York City and would like to drive to San Francisco.
- You have :
  - roadmap
  - gas station locations and their gas prices
- Want to:
  - minimize travel cost (gas expenses)

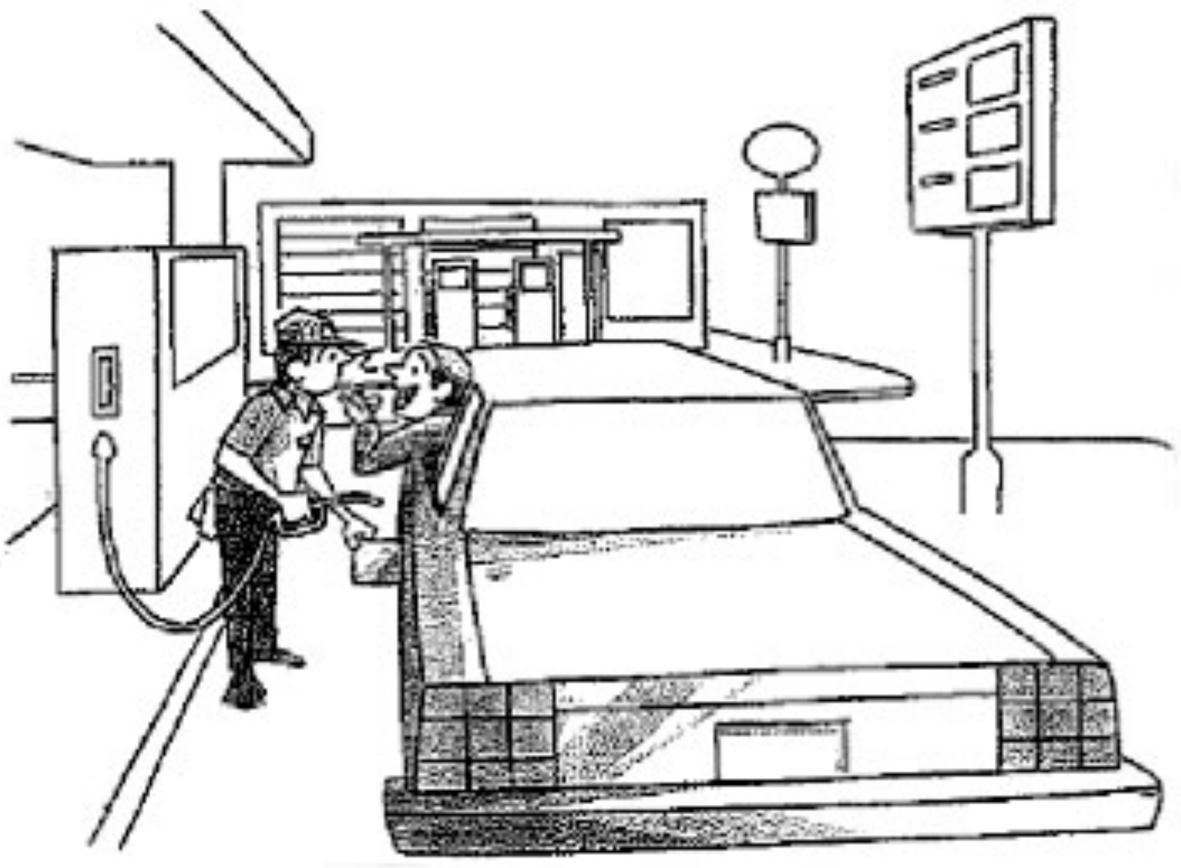
Finding gas prices online



Information and Constraints

- Information:
  - map of the road: $G=(V, E)$
  - length of the roads: $d: E \rightarrow R^+$
  - gas prices: $c: V \rightarrow R^+$
- Constraint:
  - tank capacity: U
- Goal:
  - find a min. cost solution to go from s to t

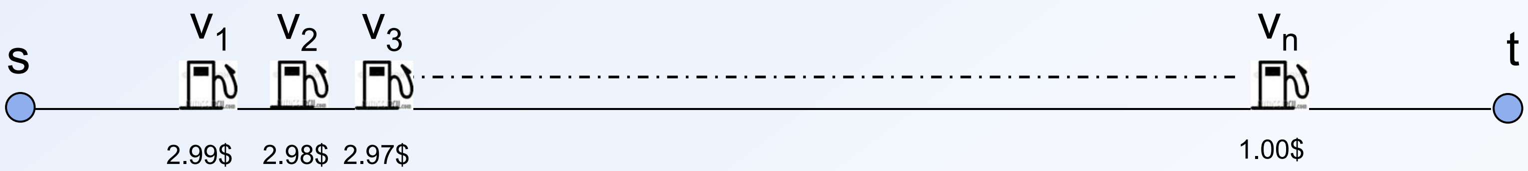




"Just enough to get me across the street
to the cheaper station."

Structure of the Optimal Solution

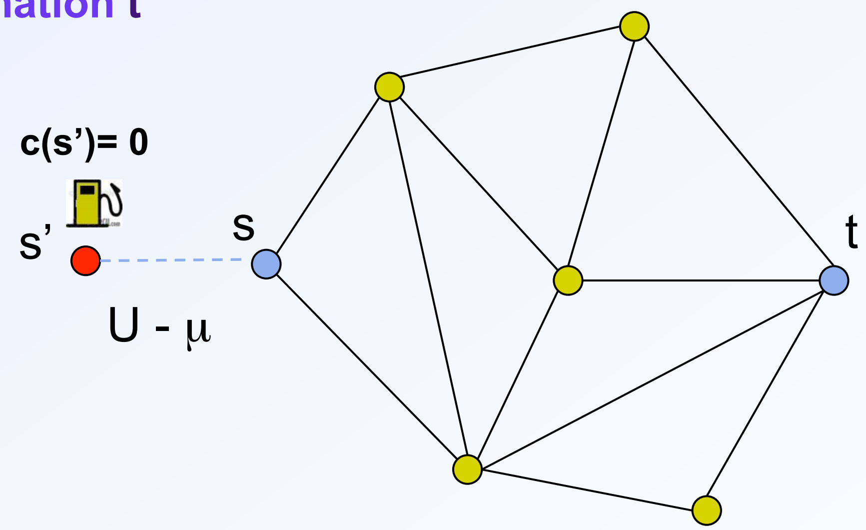
- Two vertices s & t
- A fixed path



- Optimal solution involves stops at every station!
- Thus we permit at most Δ stops.

The Problem we want to solve

- Input:
  - Road map **G=(V,E)**, source **s**, destination **t**
  - U: tank capacity
  - **d**: $E \rightarrow R^+$
  - **c**: $V \rightarrow R^+$
  - Δ: No. of stops allowed
  - μ: The initial amount of gas at s
- Goal:
  - Minimize the cost to go from **s** to **t**.
- Output:
  - The chosen path
  - The stops
  - The amount of gas filled at each stop
- We can assume we start with 0 gas at s.



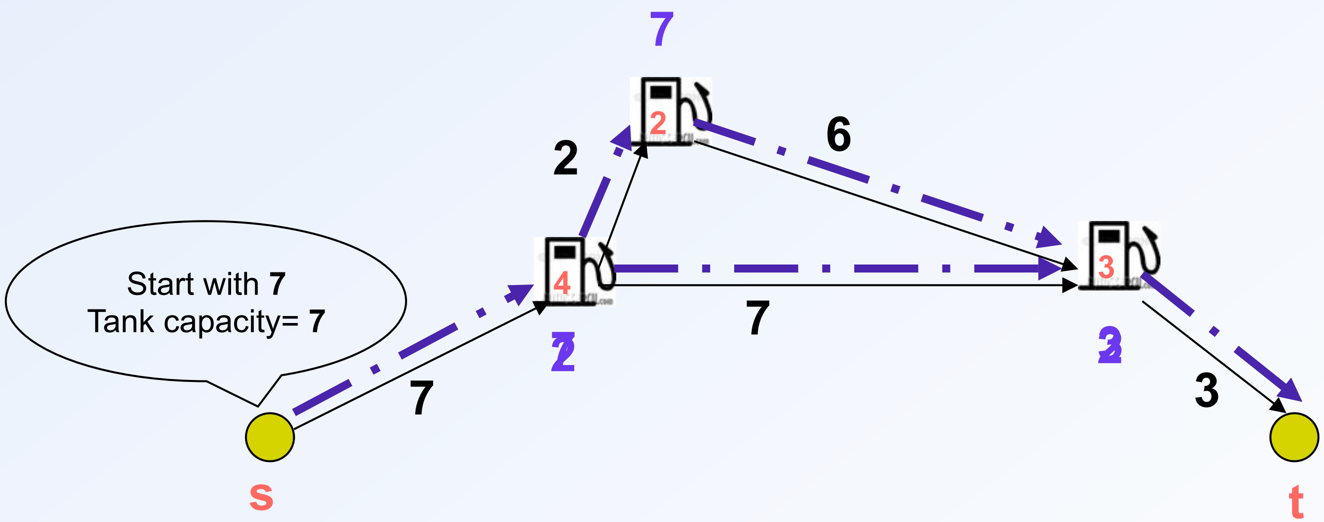
Dynamic Programming

OPT[**x**,**q**,**g**] = Minimum cost to go from **x** to t in **q** stops, starting with **g** units of gas.

- Assuming all values are integral, we can find an optimal solution in $O(n^2 \Delta U^2)$ time.
- Not a strongly polynomial time algorithm.

Why not Shortest Path?

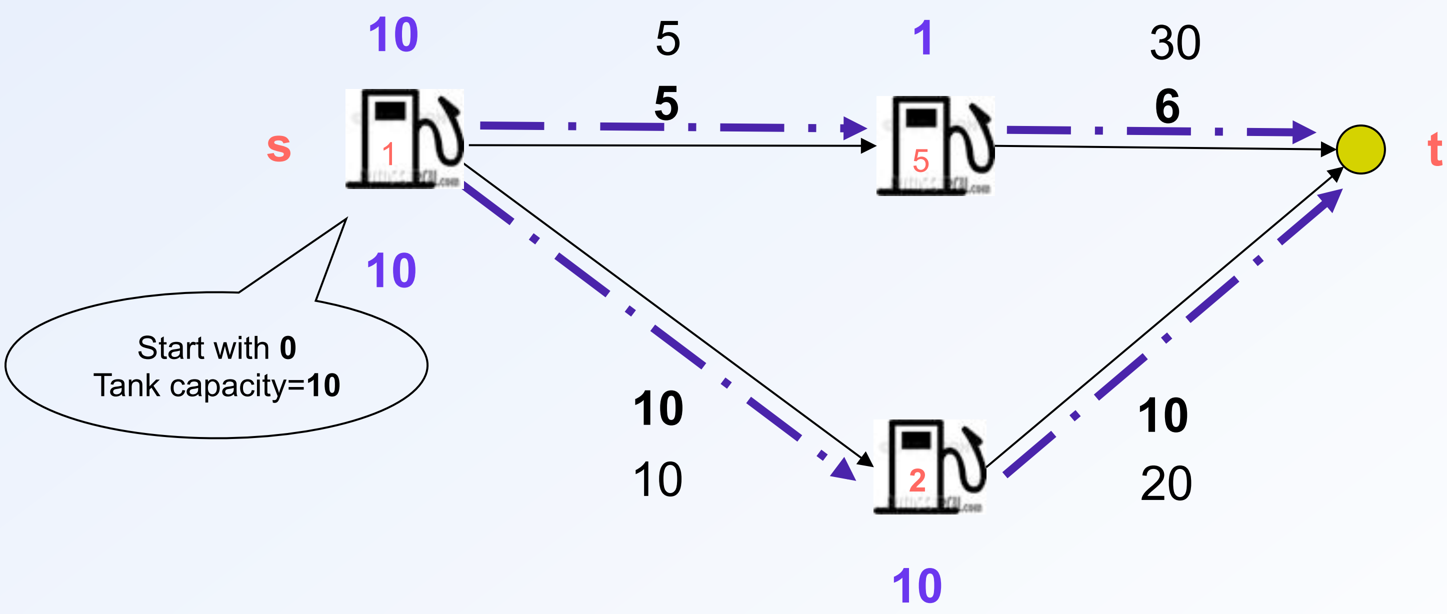
- **Shortest** path is of length **17.** Cost = 37 = 4×7 + 3×3
- **Cheapest** path is of length **18.** Cost = 28 = 4×2 + 2×7 + 2×3



One more try at Shortest Path

Let the length of (u,v) be d(u,v)×c(u), if d(u,v) ≤ U

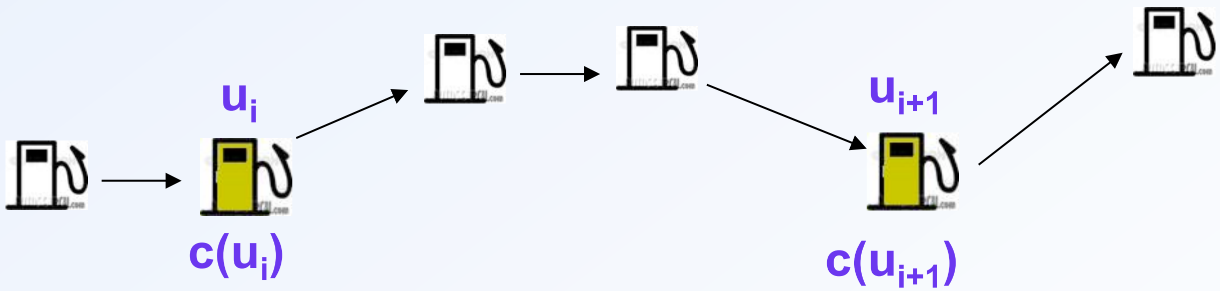
- **Shortest** path has length 30. Cost = 1x10 + 2×10 = 30
- **Cheapest** path has length 35. Cost = 1x10 + 5×1 = 15



Key Property

Suppose the optimal sequence of stops is $u_1, u_2, \ldots, u_{\Delta}$.

- If $c(u_i) < c(u_{i+1})$ ⇨ Fill up the whole tank
- If $c(u_i) > c(u_{i+1})$ ⇨ Just fill enough to reach u_{i+1}.

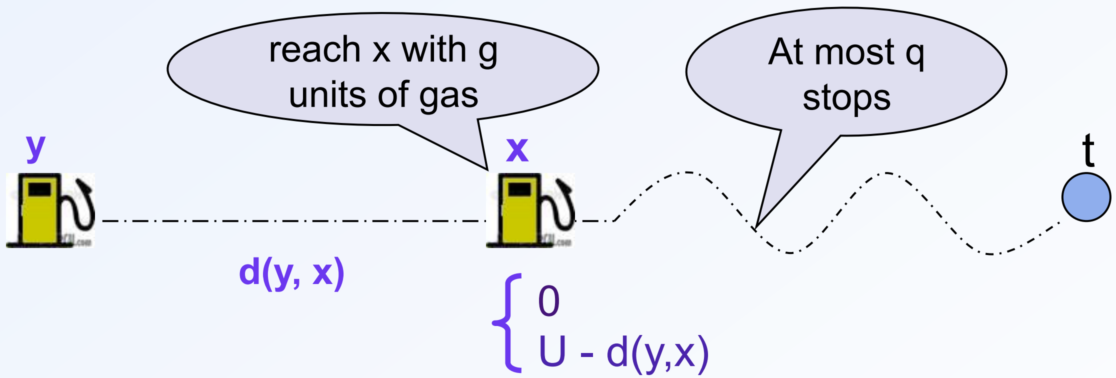


Solution

$\mathrm{OPT}[\mathbf{x},\mathbf{q},\mathbf{g}]$ = Minimum cost to go from **x** to t in **q** stops, starting with **g** units of gas.

- Suppose the stop before x was y.
  The amount of gas we reach x with is $\begin{cases} 0 & \text{if } c(x) < c(y) \\ U - d(y,x) & \text{if } c(x) \geq c(y) \end{cases}$

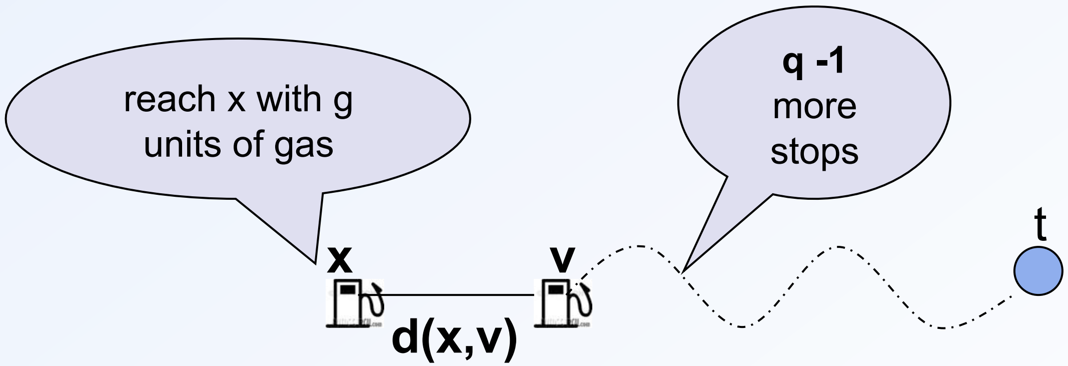
- For each x we need to keep track of at most n different values of g.



Dynamic Program

OPT[**x**,**q**,**g**] = Minimum cost to go from **x** to t in **q** stops, starting with **g** units of gas.

$$\min_v \begin{cases} OPT[v,q-1,0] + (d(x,v)-g)\ c(x) & \text{if } c(v) \le c(x)\ \&\ d(x,v) > g \\ OPT[v,q-1,U-d(x,v)] + (U-g)\ c(x) & \text{if } c(v)>c(x) \end{cases}$$



Problem Wrap Up

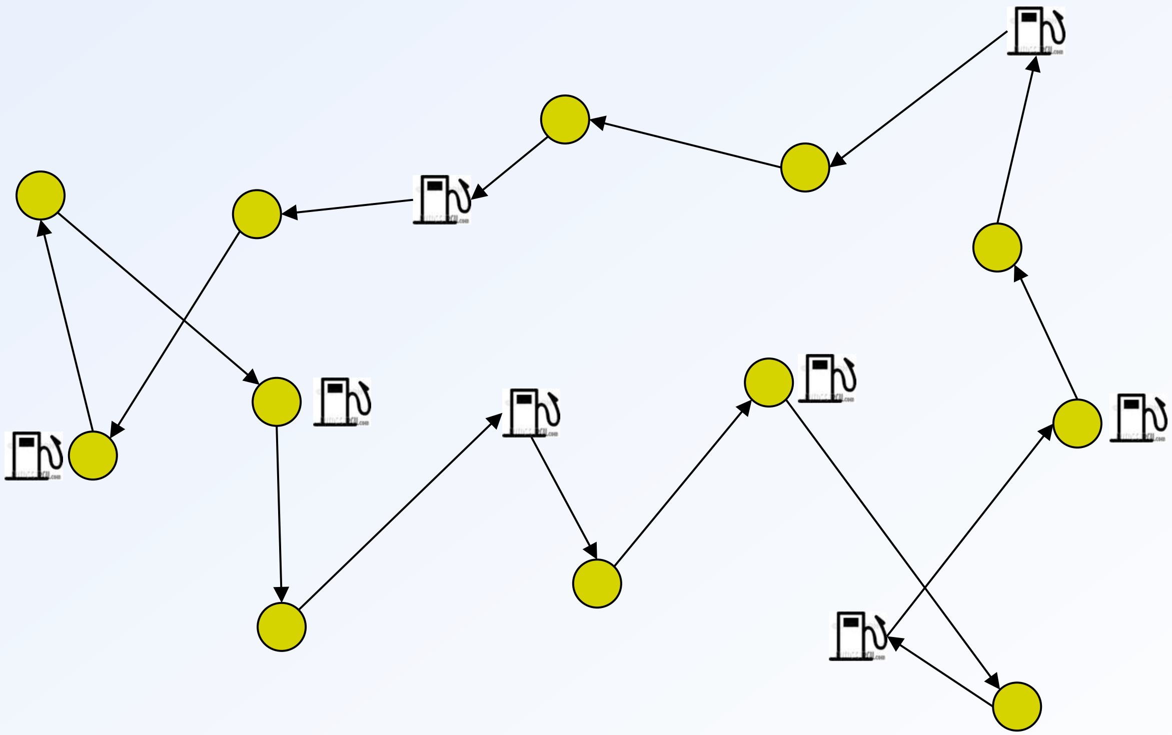
- The given DP table can be filled in:
  - $O(\Delta\, n^3)$ time by the direct and naïve solution
  - $O(\Delta\, n^2 \log n)$ time by a more intricate solution
- Faster algorithm using a different approach for the "all-pairs" version
- Faster algorithm for the fixed path with $\Delta=n$ with running time $O(n \log n)$

Tour Gas Station problem

- Would like to visit a set of cities **T**.
- We have access to set of gas stations **S**.
- Assume gas prices are uniform.
  - The problem is NP-hard even with this restriction.
  - Guess the range of prices the optimal solution uses, pay extra factor in approximation ratio.
  - Deals with gas companies.

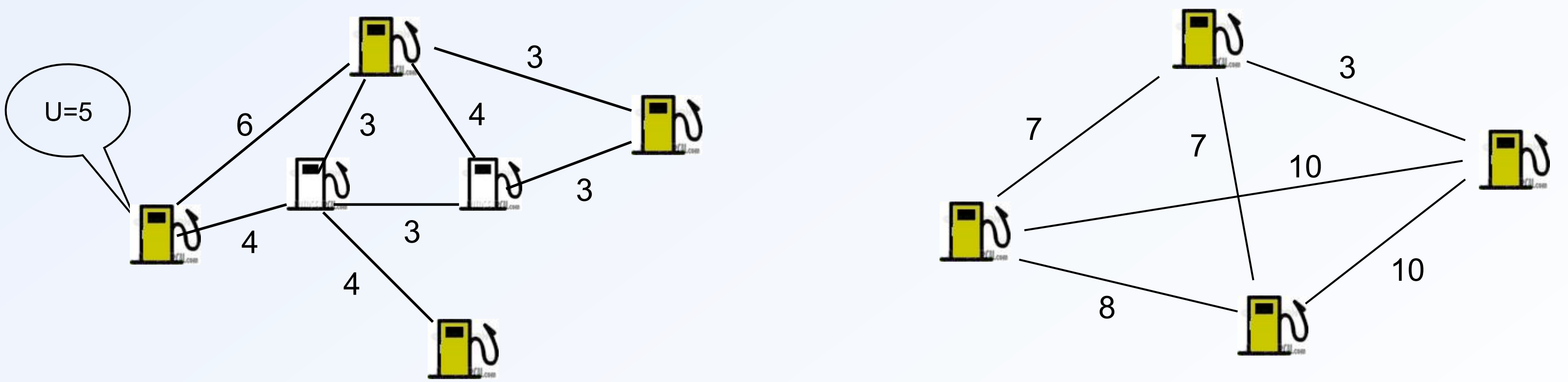
Uniform Cost Tour Gas Station

- There is a set **S** of **gas stations** and a set **T** of **cities.**
- Want to visit the cities with **min cost.**
- Gas prices are the **same.**



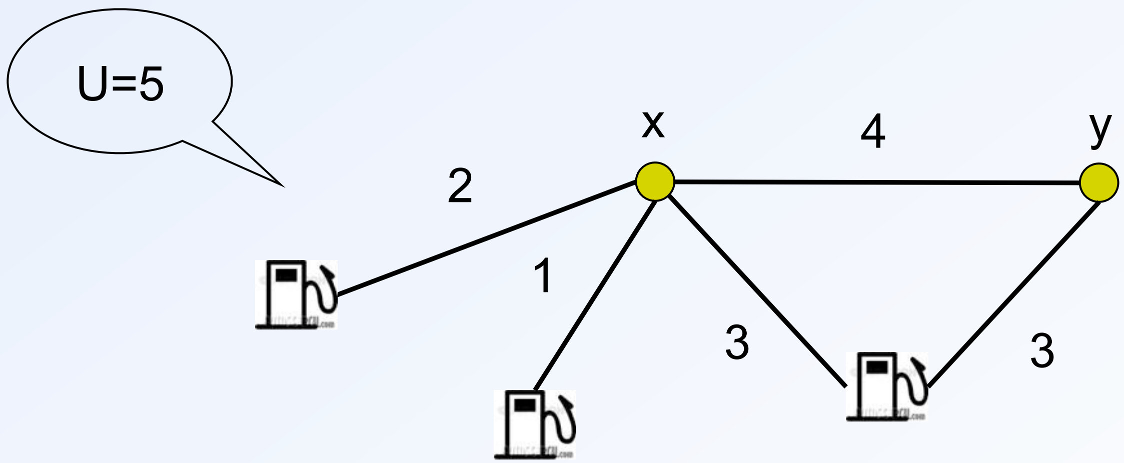
Uniform Cost Tour Gas Station

- Problem is APX-hard since it generalizes TSP.
- If each city has a gas station (T⊆S) the two problems are equivalent:
  - Let **c(x,y)** be **shortest feasible path** from x to y.
  - Triangle inequality holds in **c**



Uniform Cost Tour Gas Station

- The method works only when T⊆ S
  - **c(x, y)** depends on the **last stop before x**.
- Assumption:
  - Each city has a gas station within distance at most **αU/2.**

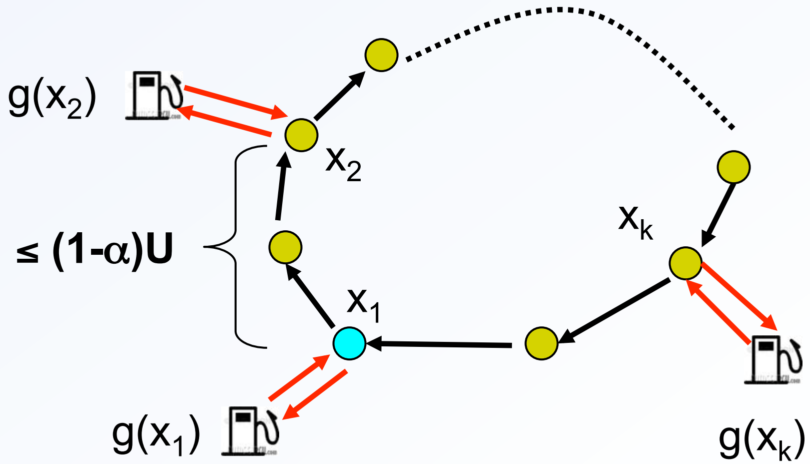


A simple case

For all edges (u,v) in the tour $d(u,v) \le (1-\alpha)U$

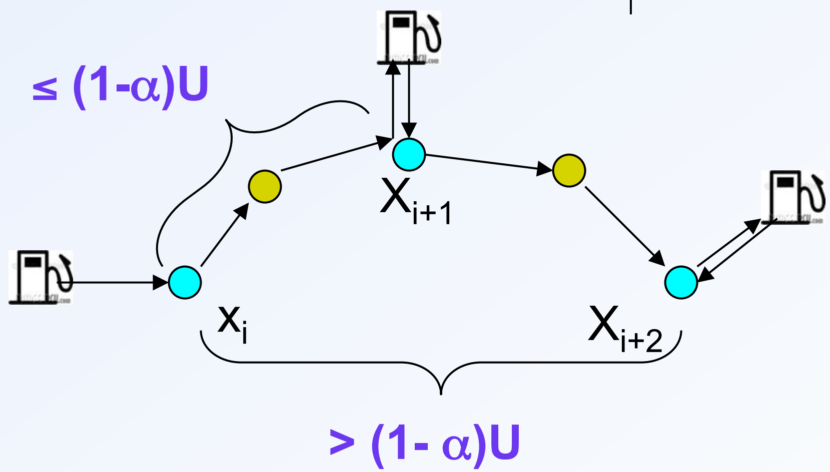
1. Find the TSP on the cities.
2. Start from $g(x_1)$, go to x_1
3. Continue along the tour until x_2, farthest city at distance at most $(1-\alpha)U$
4. Go to $g(x_2)$, repeat the procedure from $g(x_2)$
5. Continue until you reach x_1.

Let g(v) be nearest gas station to v



Uniform Cost Tour Gas Station

- In this solution
  - $|T(x_i,x_{i+1})| \leq (1-\alpha)U$
  - $|T(x_i,x_{i+2})| > (1-\alpha)U$

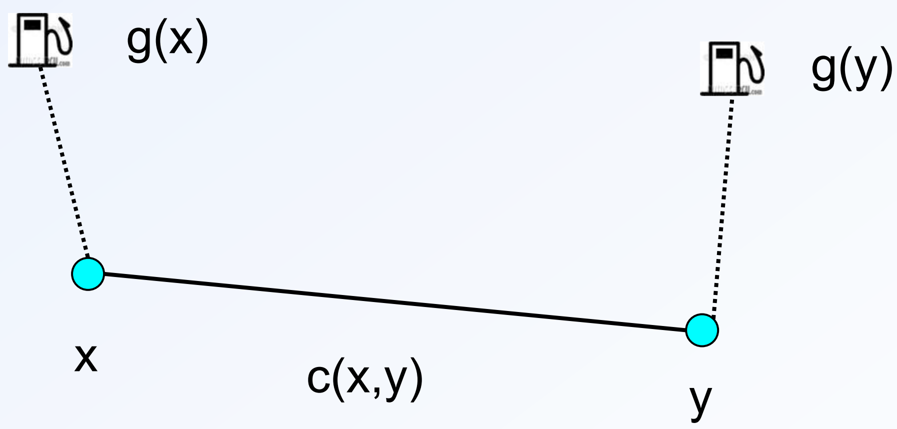


- Charge cost of trips to Gas Stations to the tour:

$$|T| + \underbrace{\alpha U k}_{\text{Round trips to } g(x_i)} \leq (1 + 2\alpha/(1-\alpha))\,|T|$$

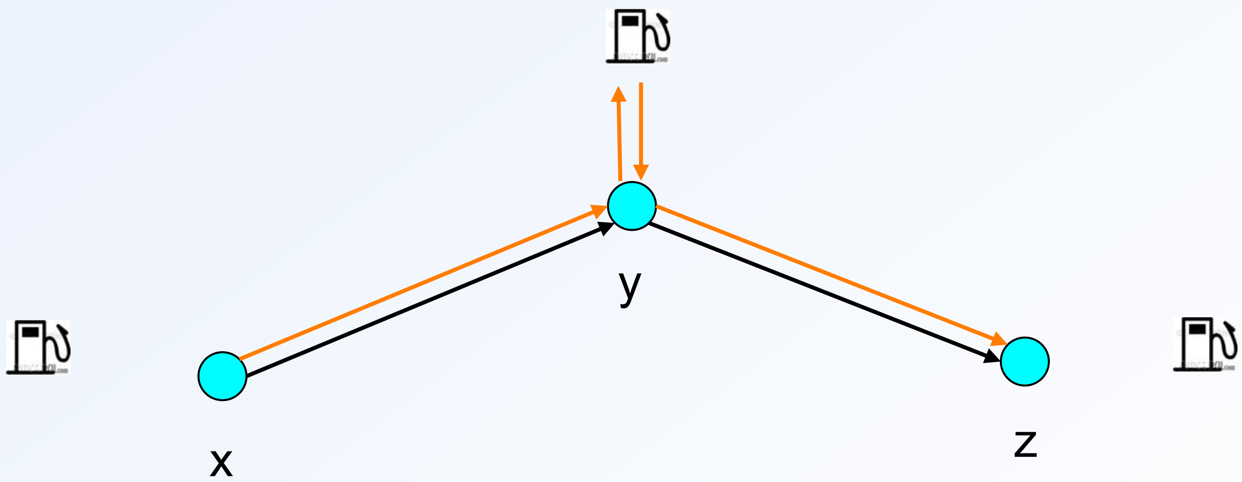
Uniform Cost Tour Gas Station

- Obtain a bound of $(1+\alpha)/(1-\alpha)$ 1.5 c(OPT).
- Note that when $\alpha=0$, then we get 1.5 c(OPT).
- Let c(x,y) be cheapest traversal to go from x to y, such that we start at g(x) and end at g(y).



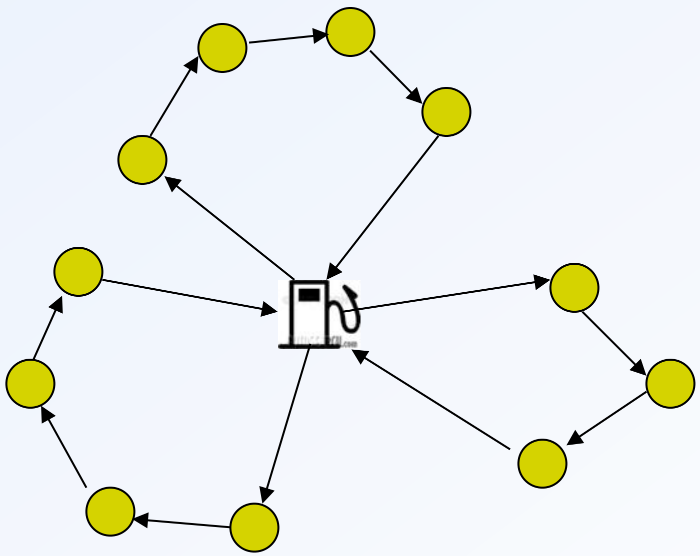
Main problem

- Lack of triangle inequality.
- To get around it, use Christofides's method combine with previous approach to get a feasible solution.



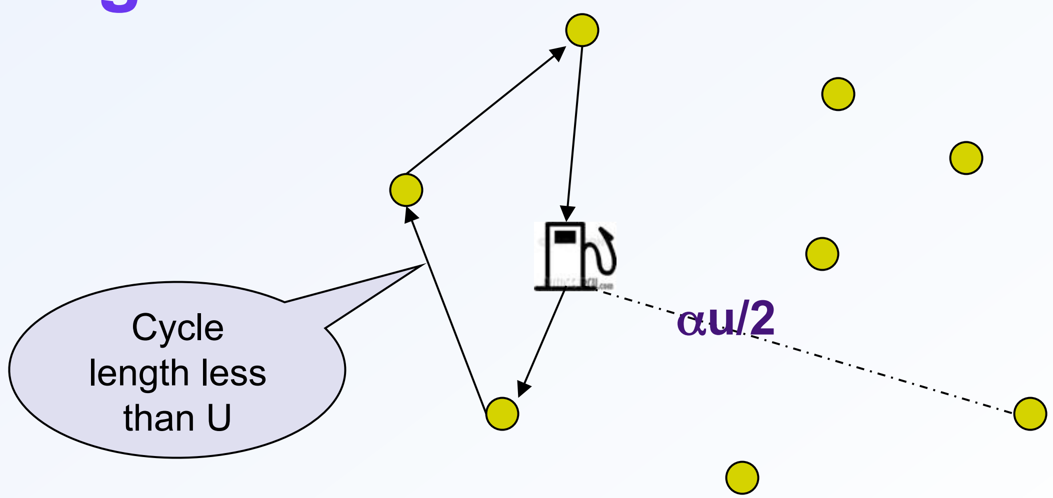
Single Gas Station Problem

- Suppose the salesman lives near a gas station.
- He wants to go to a **set of cities.**
- In each trip we can travel a distance of **U**.



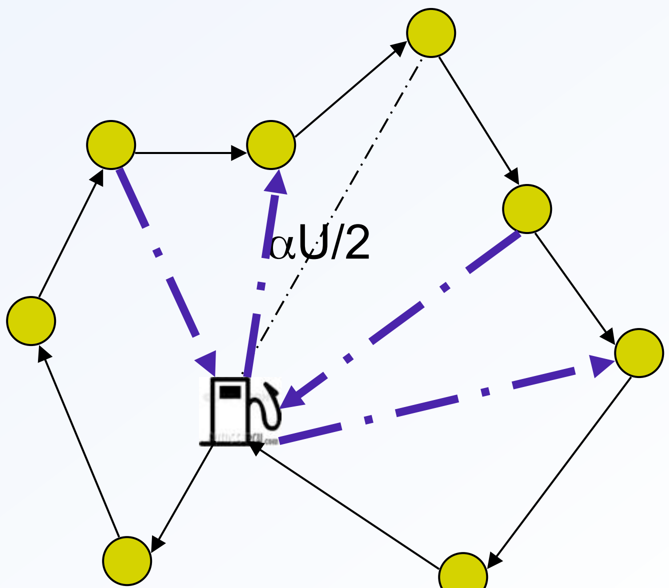
Single Gas Station

- We are given **G=(V,E)**
- We want to cover the vertices in **V**
- We have only **one gas station**
- Dist. of the **farthest city** to the gas station is **αU/2**.
- Each cycle has **length ≤ U**



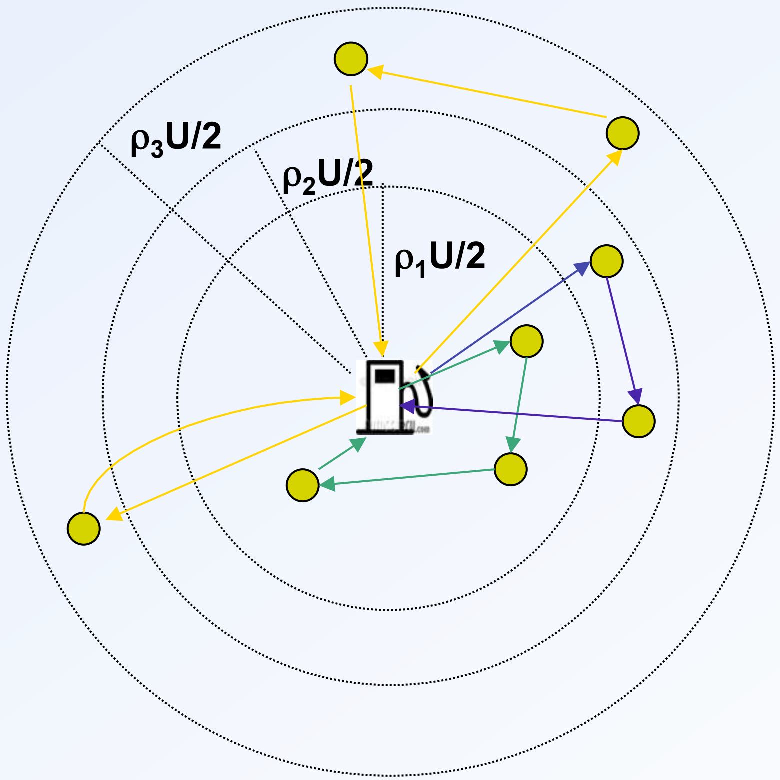
Naïve Solution

- Find the **TSP** on **cities** & **gas station**
- **Chop** the tour into parts of **length (1-α)U**
- **Connect** each segment **to the root**
- This is a **1.5/(1-α)** approximation
- Given by Li et al. [1991]



Improved Method

- **Group cities** based on their **distance** to the gas station
- Solve the problem for **each group separately**



O(log 1/(1-α))

Summary of The Results

Problem	Complexity	Approx. Ratio
2 Cities **Graph Case**	Single sink: $O(\Delta n^2 \log n)$ All pairs: $O(n^3 \Delta^2)$	
Fixed Path (Δ=n)	$O(n \log n)$	
Single gas station	APX-hard	$O(\log 1/(1-\alpha))$
Uniform Tour	APX-hard	$O(1/(1-\alpha))$

Conclusion

- Incorporate the algorithms as part of a “tool” for path planning.
- Solve the tour gas station problem with arbitrary gas prices.
- Remove the assumption that every city has a gas station at distance $\alpha U/2$.
- Planar instances

Thanks for your attention