Notes 8

- Consider a shortest path problem where we would like to find the shortest path, but we also have a fuel constraint.
  - c_{ij} : length of edge (i,j)
  - f_{ij} : fuel required to traverse edge i,j

  $f_{ij} \not\propto c_{ij}$ (not proportional)

- We can incorporate this constraint into our LP
  - let B be the fuel budget

$$\min_y \sum c_{ij} y_{ij}$$

$$\text{s.t.} \quad \sum y_{ia} - \sum y_{ai} = \begin{cases} 0 & \text{at } a \neq s, a \neq t, a \in V \\ -1 & \text{at } a = s \\ 1 & \text{at } a = t \end{cases}$$

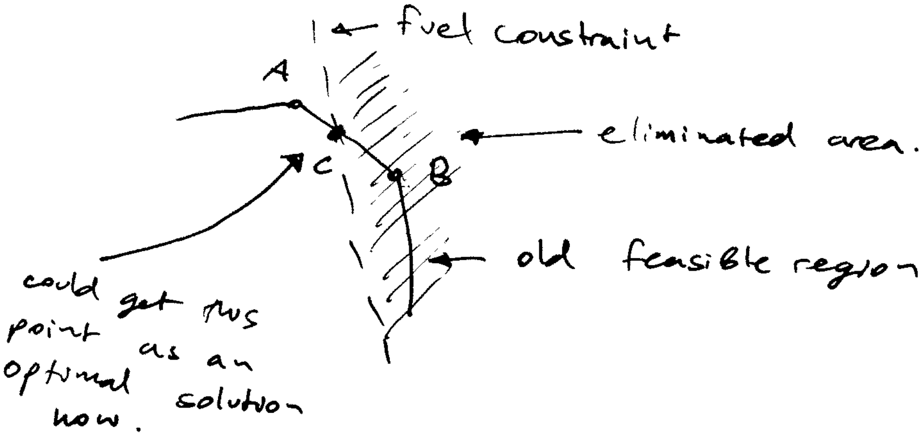
$$\sum f_{ij} y_{ij} \leq B \quad \longleftarrow \text{ fuel constraint.}$$

$$y_{ij} \geq 0.$$

As soon as we add the fuel constraint, the constraint matrix loses its unimodularity property.

- solving the LP now can give fractional solutions for the y_{ij}.
- computationally unpleasant to switch to $y_{ij} \in \{0,1\}$.
- luckily, the fractional values have a nice interpretation.

Adding the new constraint cuts off a portion of the feasible region.



- Point C is a convex combination of two of the original extreme points: A and B.
  - one of those points corresponds to a path that uses too much fuel (B)
  - the other point corresponds to a path that doesn't use up all the fuel, but is too long (A).
  - we can think of point C as giving us an optimal randomized solution.
    - e.g. $C = .12\,A + .88\,B$

      means: take the expensive path (that uses a bit too much fuel) 88% of the time.

      take the cheap path (that doesn't use all the fuel) 12% of the time.
  - in expectation, we'll use exactly the fel we are given.
  - we've found the best randomized strategy that uses no more than the fuel budget.

- Unfortunately, the new constraint also completely messes up our nice, fast algorithms... Dijkstra, for example.

- Let's look at this problem from another angle...

- Basically, we have two objective functions

  1) Length 2) Fuel

  - we want to have both of these be as small as possible, but moving one of them down may increase the other.

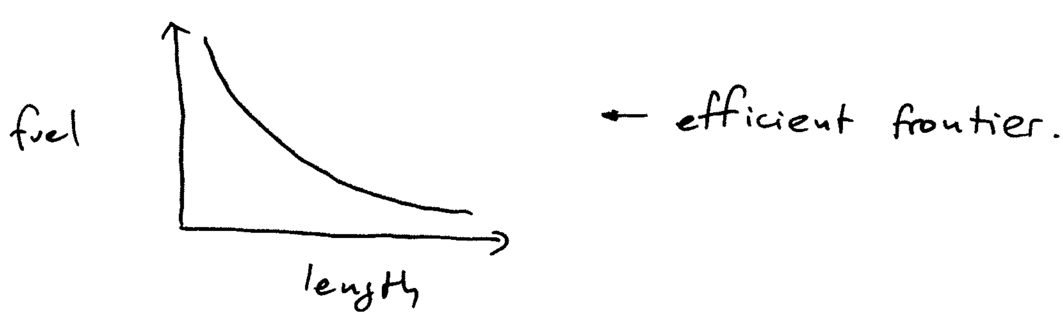
- There are three approaches to dealing with both objectives in conjunction.

  1) min. Length
     s.t. $\text{Fuel} \leq B$

  2) min Fuel
     s.t. $\text{Length} \leq B'$

  3) min $\alpha \, \text{Length} + \beta \cdot \text{Fuel}$.

- All three approaches produce the same answer:



Which approach do you like the most? Why?

$$\min \quad \sum c_{ij} y_{ij} + w \cdot \sum f_{ij} y_{ij}$$

$$\text{s.t.} \quad \sum y_{ia} - \sum y_{ai} = \begin{matrix} 0 \\ -1 \\ 1 \end{matrix}$$

$$y_{ij} \geq 0$$

– got Dijkstra back! unimodularity also.

- But, how are say approach 1) and 3) related?
- What happened to the fuel budget B? what do we set the weight w to?

1)
$$\min \quad \sum c_{ij} y_{ij}$$

$$\text{s.t.} \quad \sum y_{ia} - \sum y_{ai} = \begin{matrix} 0 \\ +1 \\ -1 \end{matrix}$$

$$\sum f_{ij} y_{ij} \leq B$$

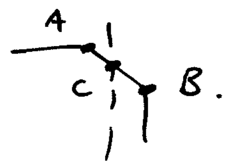
$$y_{ij} \geq 0$$

3)
$$\min \quad \sum c_{ij} y_{ij} + w \cdot \Big(\underbrace{\sum f_{ij} y_{ij} - B}_{\text{positive if we violate the fuel budget}} \Big)$$

$$\text{s.t.} \quad \text{Netflow}(y)$$

if we set w to be equal to exactly the optimal dual variable of 1), the answers (values) of 1) and 3) are exactly equal.

- This is very cool because we can use Dijkstra on 3), and if we knew the right value of w, we would get the right optimal objective value. of 1).

  - What about finding the optimal solution to 1)?
    - that is generally more difficult... involves finding the right convex combination of solutions ... remember 

    - can be done.

So, how do we find the right value for w?

- binary search:
  - if we put in too big of a value, the path we get out will underuse its fuel.
  - if we put in too small of a value, the path we get out will overuse its fuel.

Seems like a wasteful thing to do?

Dijkstra is so much faster than LP, that even doing a binary search for w is faster than solving 1).