Outline:

- motivational examples
- dynamic programming: the big picture
- basic optimization techniques

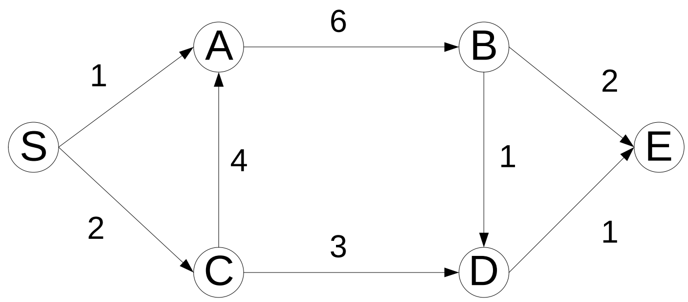
1 Motivational examples

- System identification: when we try to know someone, we ask him/her a series of questions and analyze the responses. Well designed questions give us rich information about the person we interview. Consider a similar idea in a mechanical system. System identification in ME233 teaches how to design these questions, order them, and analyze the responses, to reveal the properties of the system.
- Optimal path for you to come from your house to campus: this can conceptually be solved by the topic we are going to discuss today.

2 Big picture of dynamic programming

- history: developed in the 1950's by Richard Bellman
- "programming": has nothing to do with computers at the time of its introduction (computers were not popular at all at that time); it actually means "planning"
- it is a very useful idea/concept rather than a set of equations
- lots of applications: control, computer science, economics
- key idea: solve a (usually large and difficult) problem via solving a collection of sub problems

Example: (path planning) Consider the following graph. We are interested to go from node S to node E. The numerical values are the costs (e.g., fuel consumption, time, etc) of the paths.



- example application: air traffic design for optimal fuel consumption
- solution: $S \to C \to D \to E$. (For this simple case, we can solve the problem by observation. But the complexity increases greatly as the number of nodes grow.)
- key points to analyze the problem:
  - if node C is on the optimal path, the then path from node C to node E must be optimal as well ($C \to D \to E$)
  - the optimal cost to go is only a function of the node index (recall that in LQ, $J_{k-1}^{o} = f\left(x\left(k-1\right)\right)$, which is a function of the state at time $k-1$)
- the idea of dynamic programming: let $dist\left(E\right)$ denote the minimum distance from node S to node E.

– Backward analysis:

$$\begin{aligned} dist\,(E) &= \min\{dist\,(B)+2, dist\,(D)+1\} \\ dist\,(B) &= dist\,(A)+6 \\ dist\,(D) &= min\{dist\,(B)+1, dist\,(C)+3\} \\ dist\,(C) &= 2 \\ dist\,(A) &= \min\{1, dist\,(C)+4\} \end{aligned}$$

– Understanding the above procedure: demonstrated during discussion

– Forward computation

$$\begin{aligned} dist\,(C) &= 2 \\ dist\,(A) &= 1 \\ dist\,(B) &= 1+6=7 \\ dist\,(D) &= 5 \\ dist\,(E) &= 6 \end{aligned}$$

– Hence the optimal cost is 6 and the path is $S \to C \to D \to E$.

3 Basic optimization techniques

- brief history:
  - 100BC: birth of linear algebra in China–"Nine Chapters of Mathematical Methods"; 300BC in China–linear equations
  - 18th century: least squares for linear systems by Gauss in Europe
- basic concept: for a certain class of problems,[1] e.g., $f(x) = x^2 - 2x + 1$, $f(x) = ||Ax + b||_2$, $\min f(x)$ is achieved by x^* that satisfies $\nabla f(x^*) = 0$, i.e., the gradient at x^* must be zero
- gradient computation:

$f(x)$	$\nabla f(x)$
$\|x\|_2$	$\frac{x}{\|x\|_2}$
$\|x\|_2^2$	$2x$
$y^T\mathrm{x}$	y
$x^T y$	y
$\frac{1}{2}\|Ax-b\|_2^2$	$A^T(Ax-b)$

- example computation: $||x||_2^2 = x^T x = x_1^2 + x_2^2 + \ldots$, $\nabla f(x) = 2x$; $||Ax||_2^2 = x^T A^T A x$, $\nabla f(x) = 2A^T A x$

[1] unconstrained convex problems