Monte Carlo Methods

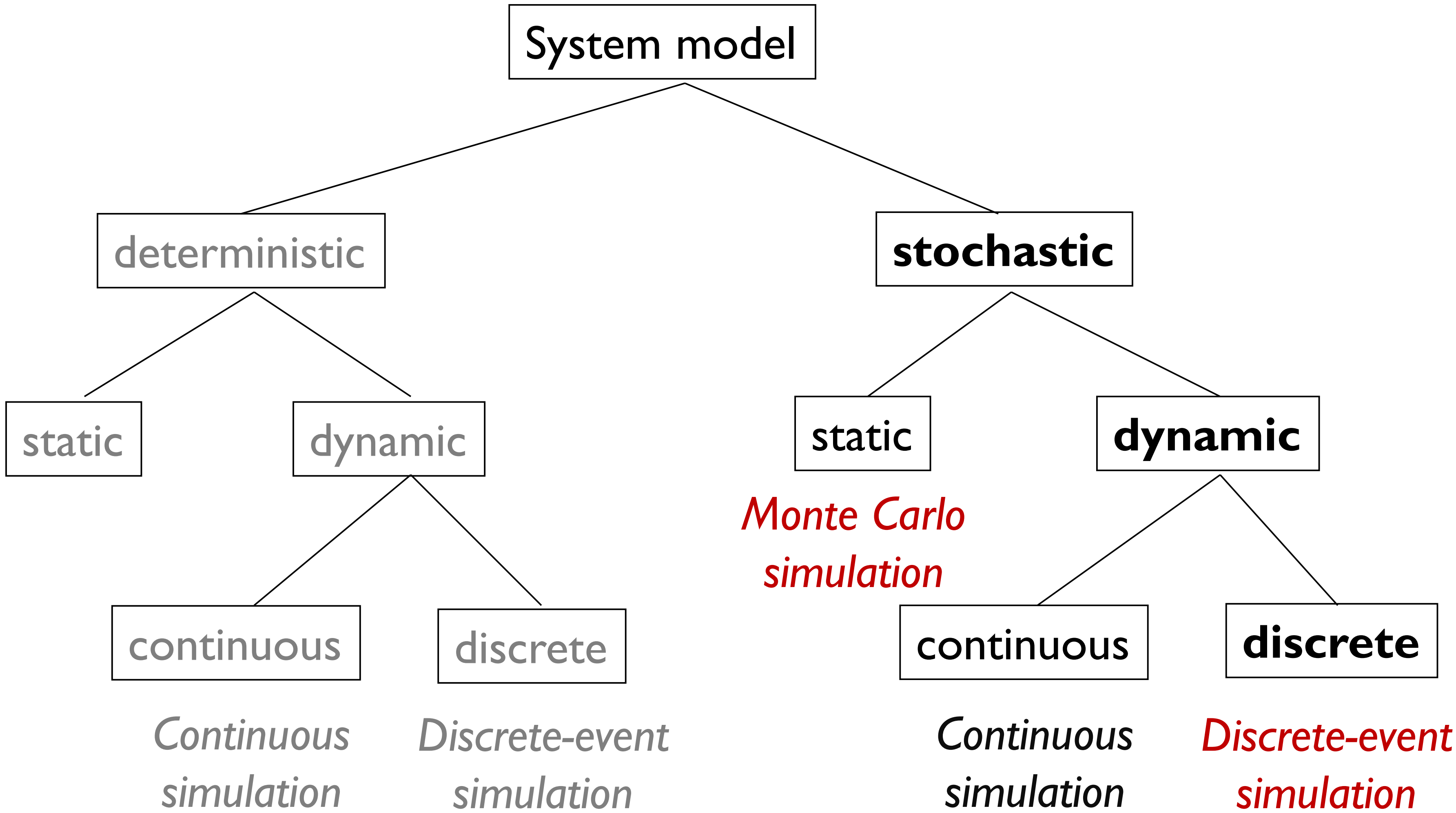
CS1538: Introduction to Simulations

Monte Carlo Simulation

- We've primarily focused on simulation models that are stochastic and dynamic

- Today, we start talking about static models, often called Monte Carlo Simulations
  - May be useful for determining quantities difficult to compute by other means
    - Idea is to determine some quantity / value using random numbers
    - Ex: Evaluating an integral that has no closed analytical form

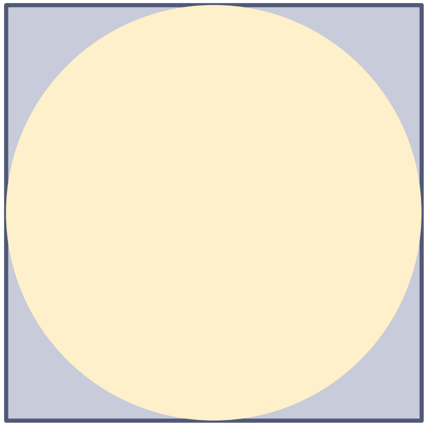
Types of Simulation Models



Slide adapted from: R. Fujimoto

A First Example

- Suppose:
  - We don't know the formula for the area of a circle
  - We do know how to compute the area of a square in closed form.
- We can figure out the area of the circle by bounding it inside a square (such that the square's sides are tangent to 4 points of the circle)
- We then generate a large number of random points known to be within the square



A First Example

- Calculate the % of points in the circle
- This approximates the ratio between the area of circle and the square
- $A_{circle} \sim ratio * A_{square}$

Empirical vs. Axiomatic Probability

- Consider a random experiment with possible outcome C
- Run the experiment N times, counting the number of C outcomes, N_C
- The empirical probability is the relative frequency of occurrence of C is the ratio N_C/N
- As $N \rightarrow \infty$, N_C/N converges to the "true" axiomatic probability of C, or

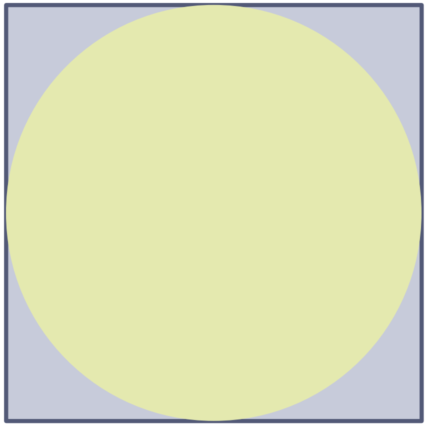
$$p(C) = \lim_{N \to \infty} \frac{N_C}{N}$$

Square/Circle Example

- Suppose we have a circle with diameter 1 and want to know its area. How would we use Monte Carlo simulation to perform the calculation?

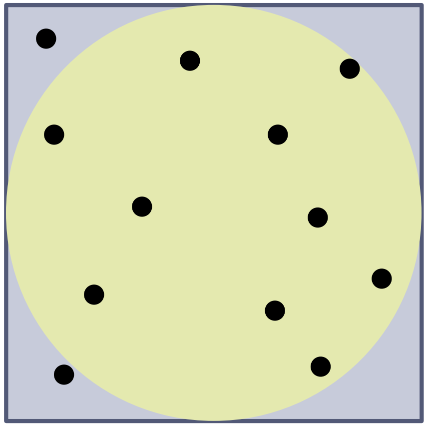
Square/Circle Example

- Suppose we have a circle with diameter 1 and want to know its area. How would we use Monte Carlo simulation to perform the calculation?



Square/Circle Example

- Suppose we have a circle with diameter 1 and want to know its area. How would we use Monte Carlo simulation to perform the calculation?



Monte Carlo Integration

- Suppose we have a function f(x) that is defined and continuous on the range [a,b]
- Let F(x) be a function that defines the area under f(x)
  - So $F(x) = \int f(x)dx$ or equivalently, $F'(x) = f(x)$
- The mean value theorem for integral calculus states that there exists some number c, with a < c < b such that:

$$F'(c) = \frac{F(b) - F(a)}{b - a} \quad or$$

$$\frac{1}{b-a}\int_a^b f(x)dx = f(c) \quad or$$

$$\int_a^b f(x)dx = (b-a)f(c)$$

There is some point c between a and b such that the change in F(x) is the average

If we think of F(x) as the area under f(x), f(c) gives the average height, and we get the full area with (b-a)*f(c)

Monte Carlo Integration

- So if $\int_a^b f(x)dx$ is difficult to compute directly, but we know how to calculate f(x), we can evaluate the integral by figuring out f(c)
- Using Monte Carlo method:
  - Choose N random values $x_1, \ldots, x_N$ in [a,b]
  - Calculate the average (or expected) value, $\dot{f}(x)$ in that range:

$$\overline{f(x)} = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \approx f(c)$$

  - Now we can estimate the integral value as

$$\int_a^b f(x)dx = (b-a)\overline{f(x)}$$

Example: Monte Carlo Integration

- f(x) = 3x
- What is $\int_0^2 f(x)dx = F(x)|_0^2$?

- Assume a PRG returns these 10 values:
- 0.76, 0.60, 0.38, 0.65, 0.05, 0.96, 0.71, 0.57, 0.06, 0.90

Application of Monte Carlo Integration

- Probabilistic Reasoning
  - Parameter Estimation
  - Approximate inferences

- Markov Chain Monte Carlo
  - e.g. Simulated Annealing

Parameter estimation

- Let's consider a simple probability model: that of a biased coin with unknown bias *p*. Suppose we toss it a few times and got "HTTHT." What's the chance of seeing H on the next toss?
  - Option 1: Use Maximum Likelihood Estimate
    - How would we use it?
  - Option 2: But suppose we actually know something about the coin, like we strongly believe it to be fair for some reason. Then we may want to take the maximum a posteriori approach
    - $p = \text{argmax}_{p^*} \Pr(\text{"HTTHT"}|p^*)\Pr(p^*)$
    - This lets us directly put in a prior $\Pr(p^*)$, like $\Pr(p=0.5) = 0.8$
  - Option 3:
    - Figure out the expected outcome of the next toss across all possible p
    - $\Pr(\text{next H} \mid \text{"HTTHT"}) = \int \Pr(\text{next H} \mid p) \Pr(p \mid \text{"HTTHT"})\, dp$

Brief tour of Bayesian Reasoning

- A full joint probability distribution can answer any question about the domain, but it can become intractably large as the number of variables grow

Example

- You have a new burglar alarm installed at home. It's fairly reliable at detecting burglary, but also responds occasionally to minor earthquakes.
- You have two neighbors, John and Mary. They've both promised to call you when they hear the alarm.
  - John promised to call you when he hears the alarm, but sometimes confuses the phone ringing with the alarm and calls you then too.
  - Mary likes loud music and sometimes misses the alarm.
- Given who has called or not called, what is the probability of a burglary?

Example: Joint Probability Table

			Burglary=True		Burglary=False	
			Earthquake=T	Earthquake=F	Earthquake=T	Earthquake=F
Alarm=T	J=T	M=T	5.985E-07	0.295212292	0.000182517	0.000314056
		M=F	2.565E-07	0.126519554	7.82217E-05	0.000134595
	J=F	M=T	6.65E-08	0.032801366	2.02797E-05	3.48951E-05
		M=F	2.85E-08	0.014057728	8.6913E-06	1.4955E-05
Alarm=F	J=T	M=T	2.5E-11	1.4955E-05	3.54645E-07	0.000249001
		M=F	2.475E-09	0.001480548	3.51099E-05	0.024651124
	J=F	M=T	4.75E-10	0.000284146	6.73826E-06	0.004731024
		M=F	4.7025E-08	0.028130411	0.000667087	0.46837135

- How many observations are needed to determine this table?

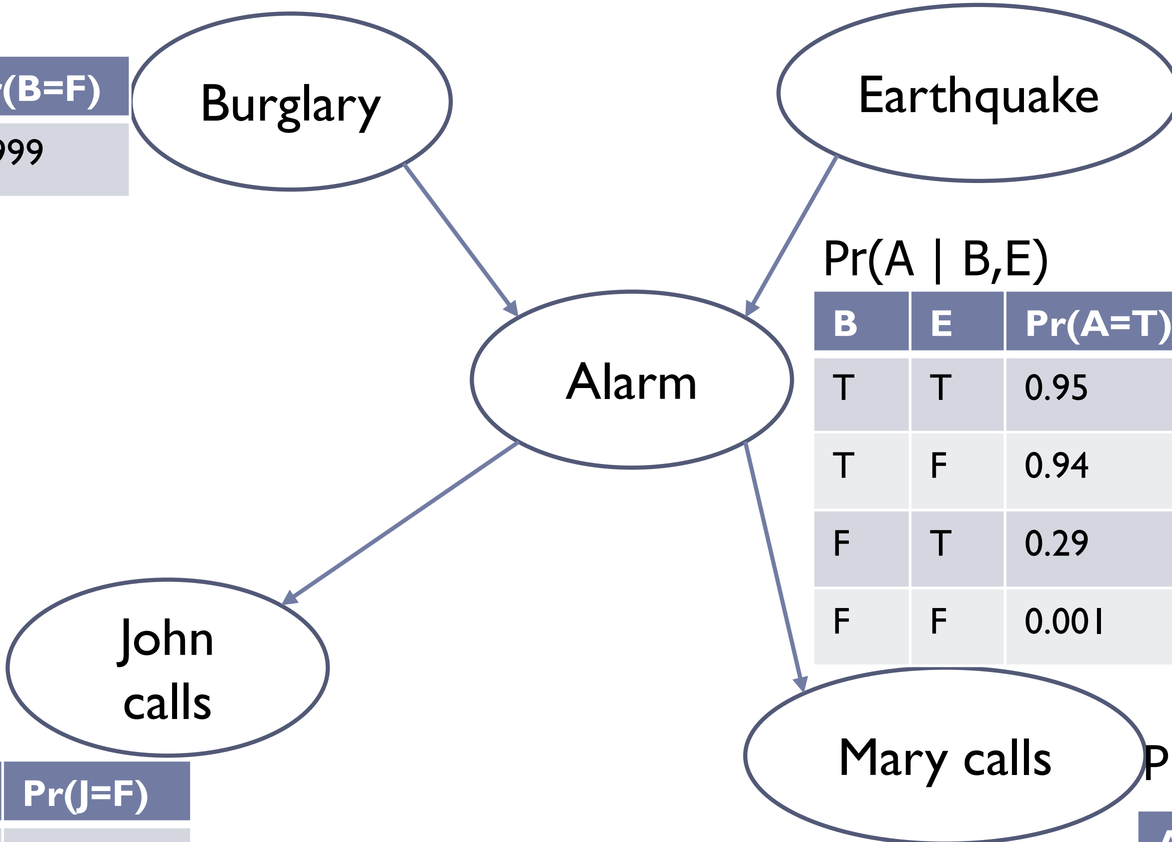
Bayesian network

- Don't want to specify the probabilities for all possible worlds one by one
- Independence and conditional independence relationship between variables reduce the number of probabilities we need to specify
- Bayesian network is used to represent dependencies among variables
- A Bayesian network is a directed graph
  - Each node represents a random variable
  - A set of directed links or arrows connects pairs of nodes. If there is an arrow from node X to node Y, X is said to be a parent of Y
  - Each node X_i has a conditional probability distribution $P(X_i \mid Parents(X_i))$ that quantifies the effect of the parents on the node

Example

- Burglary and Earthquake are somewhat independent
  - Why not separate them to be independent?
- Alarm goes off if Burglary or Earthquake happens
  - Alarm depends on Burglary and Earthquake
- John calling and Mary calling are independent
  - i.e. John doesn't call Mary to tell you
  - But, John and Mary calling aren't independent – they both might call if the alarm is going off
    - John and Mary are conditionally independent, given alarm

Example: Bayesian Network



Pr(B=T)	Pr(B=F)
0.001	0.999

Pr(E=T)	Pr(E=F)
0.002	0.998

Pr(A | B,E)

B	E	Pr(A=T)	Pr(A=F)
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

Pr(J | A)

A	Pr(J=T)	Pr(J=F)
T	0.90	0.10
F	0.05	0.95

Pr(M | A)

A	Pr(M=T)	Pr(M=F)
T	0.70	0.30
F	0.01	0.99

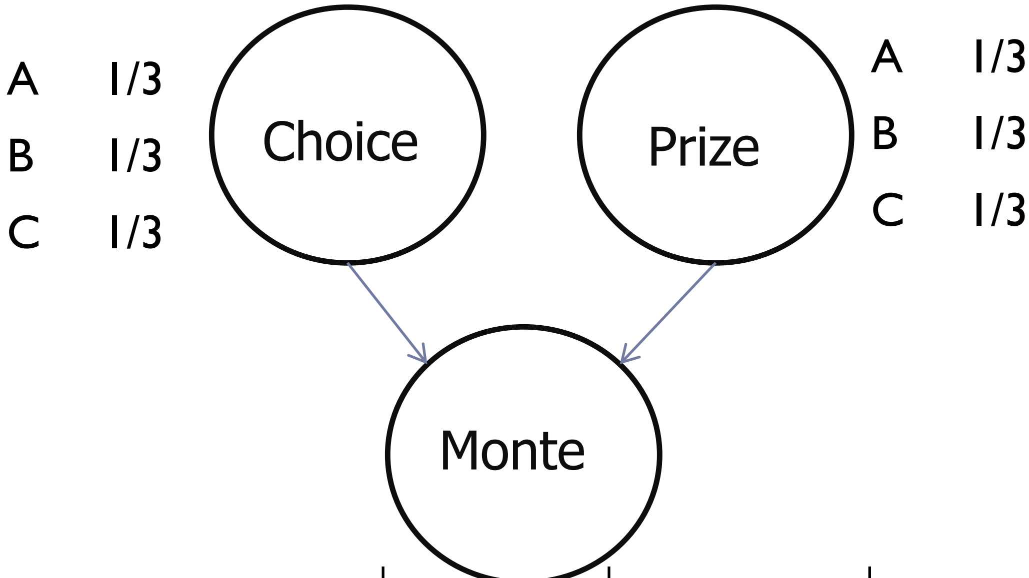
Example: the Monty Hall problem

- A somewhat counter-intuitive probability problem
  - Player is given choice of 3 doors
    - Behind one is a grand prize
    - Behind the other two are duds
  - After player chooses a door, Monty opens *one of the other two doors*, and he always shows one that has a dud.
  - The player has option to keep the original choice or to switch to the remaining closed door
  - What should player do?

Monty Hall – Let's Make a Deal

- We can analyze the problem axiomatically with the use of conditional probabilities
- These types of problems are applications of probabilistic reasoning.
  - If you're in AI or if you've taken AI, you can refer to Chapter 13-14 of AI: A Modern Approach (Russell & Norvig) for more information.
- It turns out that for many interesting probabilistic reasoning problems, we cannot figure out the answers efficiently in closed form. Instead, we have to approximate it with sampling
  - Good application for Monte Carlo Simulation

Bayesian Analysis of the Monte Hall Problem



Prize=	A			B			C		
Choice=	A	B	C	A	B	C	A	B	C
Monte=A	0	0	0	0	½	1	0	1	½
Monte=B	½	0	1	0	0	0	1	0	½
Monte=C	½	1	0	1	½	0	0	0	0

- Should I switch?
- Given what Monte revealed, is the chance that Choice≠Prize greater than the chance that Choice=Prize?
  - Pr(Choice=Prize | Monte)
    =Pr(Choice=Prize,Monte)/Pr(Monte)
  - From network, we know:
    - Pr(Choice,Prize,Monte) = Pr(Monte|Choice,Prize) * Pr(Choice) * Pr(Prize)

Pr(Monte=B) =

$\Sigma_{Choice,Prize}$ Pr(Choice, Prize, Monte)

Solve with Direct Sampling

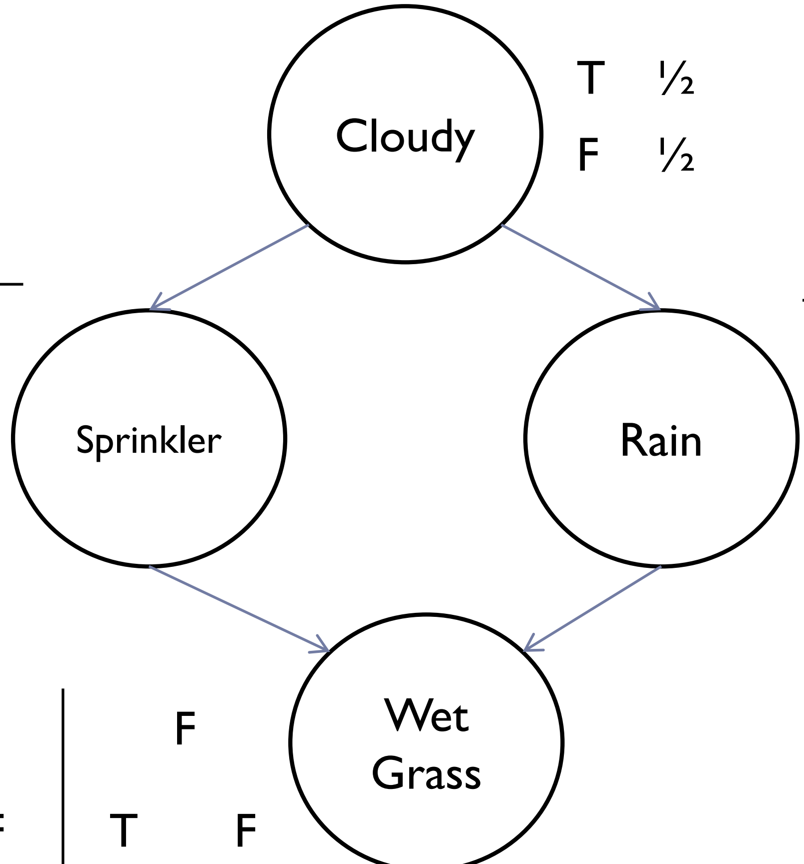
- One sample trial:
  - Set "Choice" by randomly picking a door at 1/3 chance
  - Set "Prize" by randomly picking a door at 1/3 chance
  - If our sampled Choice = Prize, randomly pick Monte's reveal from the remaining two doors at ½ chance
  - Otherwise, Monte's choice is fixed
  - Suppose we always switch after Monte's reveal
- Run many trials
  - Tally % time we win the prize
- We can then try never switch after Monte's reveal; run many trials and see how often we win the prize.

Running Example: Wet Grass

- You observe that the grass is wet.
- The grass can be wet because it rained or because the sprinkler went off.
- You observe that the sky is cloudy.

- Why is the grass wet?

Example: Wet Grass



T	½
F	½

Cloudy=	T	F
T	0.1	0.5
F	0.9	0.5

Cloudy=	T	F
T	0.7	0.2
F	0.3	0.8

Rain=	T		F	
Sprinkler=	T	F	T	F
T	0.99	0.9	0.9	0.01
F	0.01	0.1	0.1	0.99

Example: Wet Grass

- Pr(WetGrass=True | Rain=True,Sprinker=True) = ?
- Pr(Rain=False | Cloudy=True) = ?
- Pr(WetGrass=True | Rain=True) = ?
- Pr(Rain=True | Sprinkler=True) = ?

Example: Wet Grass

- Suppose we want to create a simulation that follows the distribution. How do we generate random situations?
  - Use Direct Sampling
    - Sample whether Cloudy
    - Sample whether Sprinkler based on the outcome of Cloudy
      - If Cloudy is true sample from the <0.1, 0.9> distribution
      - Otherwise from the <0.5, 0.5> distribution
    - Sample Rain given Cloudy
    - Sample WetGrass given Rain and Sprinkler's outcomes

Example: Wet Grass

- What if we want to compute Pr(Rain | Sprinkler=True)?

Example: Wet Grass

- What if we want to compute Pr(Rain | Sprinkler=True)?
  - Use Rejection Sampling
    - Like Direct Sampling, but immediately reject all samples that generate Sprinkler = False

Importance Sampling

- Also called Likelihood Weighting
- Suppose we want to compute

  P(Rain | Cloudy=True, WetGrass=False)
- Here, rejection sampling doesn't save us that much work (since we have to get to WetGrass=True)
- Want to be able to fix the random variables in the conditioning part (Cloudy, WetGrass) and just sample the rest of the network
  - This requires us to figure out how much to adjust the weight of the complete sampled point to deal with the fact that we didn't sample every variable

Importance Sampling – Likelihood Weighting

Estimate: P(Rain | Cloudy=True, WetGrass=False)

1. We want Cloudy to be true. The chance of that is 0.5
   - Set weight w to 0.5
2. Sample Sprinkler given Cloudy=True as usual
3. Sample Rain given Cloudy=True as usual
4. We want WetGrass to be False. Using sampled outcomes of steps 2 and 3, look up the chance of Pr(WetGrass=False | Sprinkler=Step2, Rain=Step3)
   - Set weight w to w * Pr(WetGrass=False | Step2, Step3)
5. If Rain=True from Step 3, we add w to the Rain=True column; otherwise we add w to the Rain=False column

Markov Chain Monte Carlo

- The previous 3 sampling techniques for large networks and many evidence variables
- Don't generate each sample from scratch
- Make a random change to the previous sample
  - Simulated Annealing belongs to this family of simulations
  - Another one is Gibbs Sampling
  - A more general version (we won't cover in this class) is called the Metropolis-Hastings Sampling

Gibbs Sampling (Mechanism)

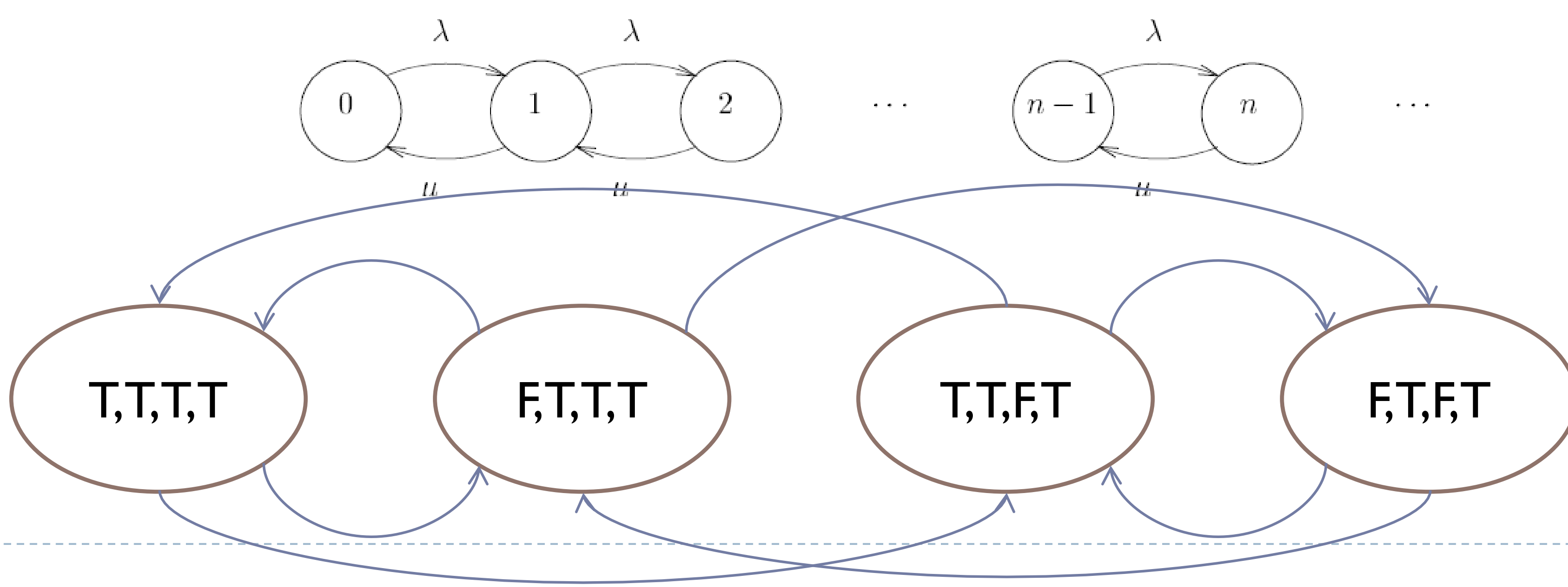
- Illustrative example: estimate

  P(Rain|Sprinkler=True, WetGrass=True)

  - Get a first sampled point: [Cloudy=True, Rain=False, Sprinkler=True, WetGrass=True]
  - Resample the non-evidence random variables (Cloudy, Rain):
    - Cloudy: sample from a distribution based on the previous sample: P(Cloudy|Rain=False,Sprinkler=True,WetGrass=True)
    - We now have a new sample point where Cloudy's value is replaced by the new sample
    - Rain: sample from P(Rain | Cloudy, Sprinkler, WetGrass) where the conditioning random variable's values are all determined from the previous sample point.
  - Repeat resampling until the distribution of unique samples is stable

Gibbs Sampling (Idea)

- Each unique sample point can be thought of as a Markov state, and we're just taking a random walk on the chain of Markov states.
  - Recall from our discussion during queueing models that a Markov Chain is like a probabilistic finite state, where you transition from state to state probabilistically



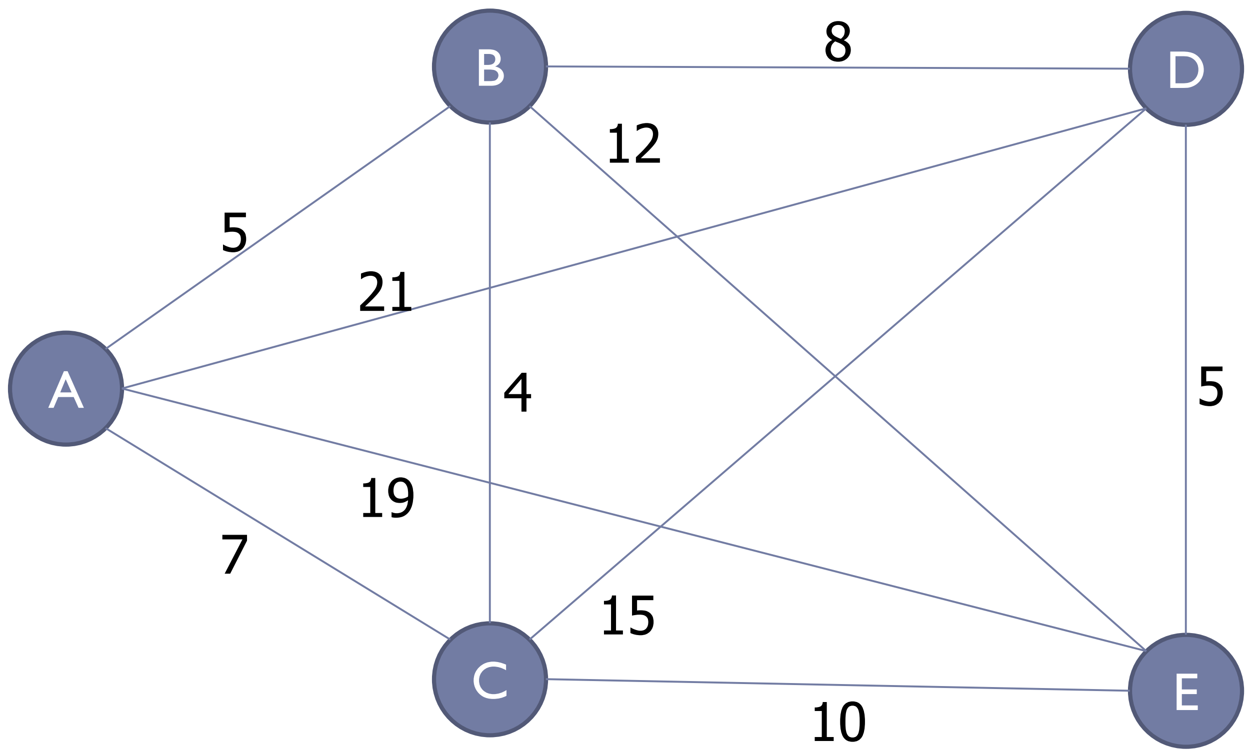
Another application: Simulated Annealing

- Inspired by physical annealing processes used in materials science
  - Obtain an improved solution by randomly changing candidate solutions to "neighbor" solutions
  - Early on, high temperature → lots of randomness
  - Progression: lower the temperature to decrease randomness

  - Example – Traveling Salesman Problem (TSP)
    - Given a completely connected graph with weighted edges, what is the shortest cycle that visits each vertex exactly one time?
    - Find a tour path; then randomly "perturb" it
      - If the new solution is "better", keep it; even if it's worse, we might keep it with some probability (that depends on the "temperature")

TSP Example

- What is the shortest tour?
  - NP-Hard problem to solve exactly

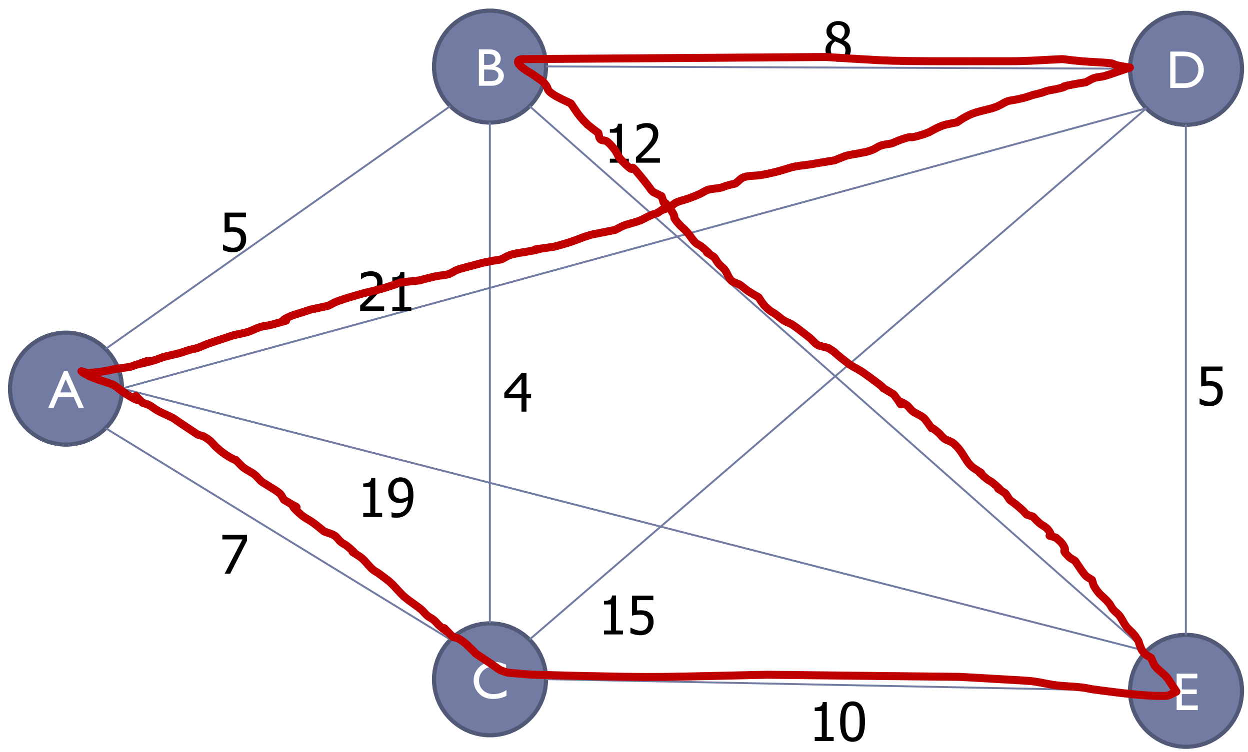


TSP Example

T = 1.0

- Simulated Annealing
  - Randomly create cycle: A, D, B, E, C, A

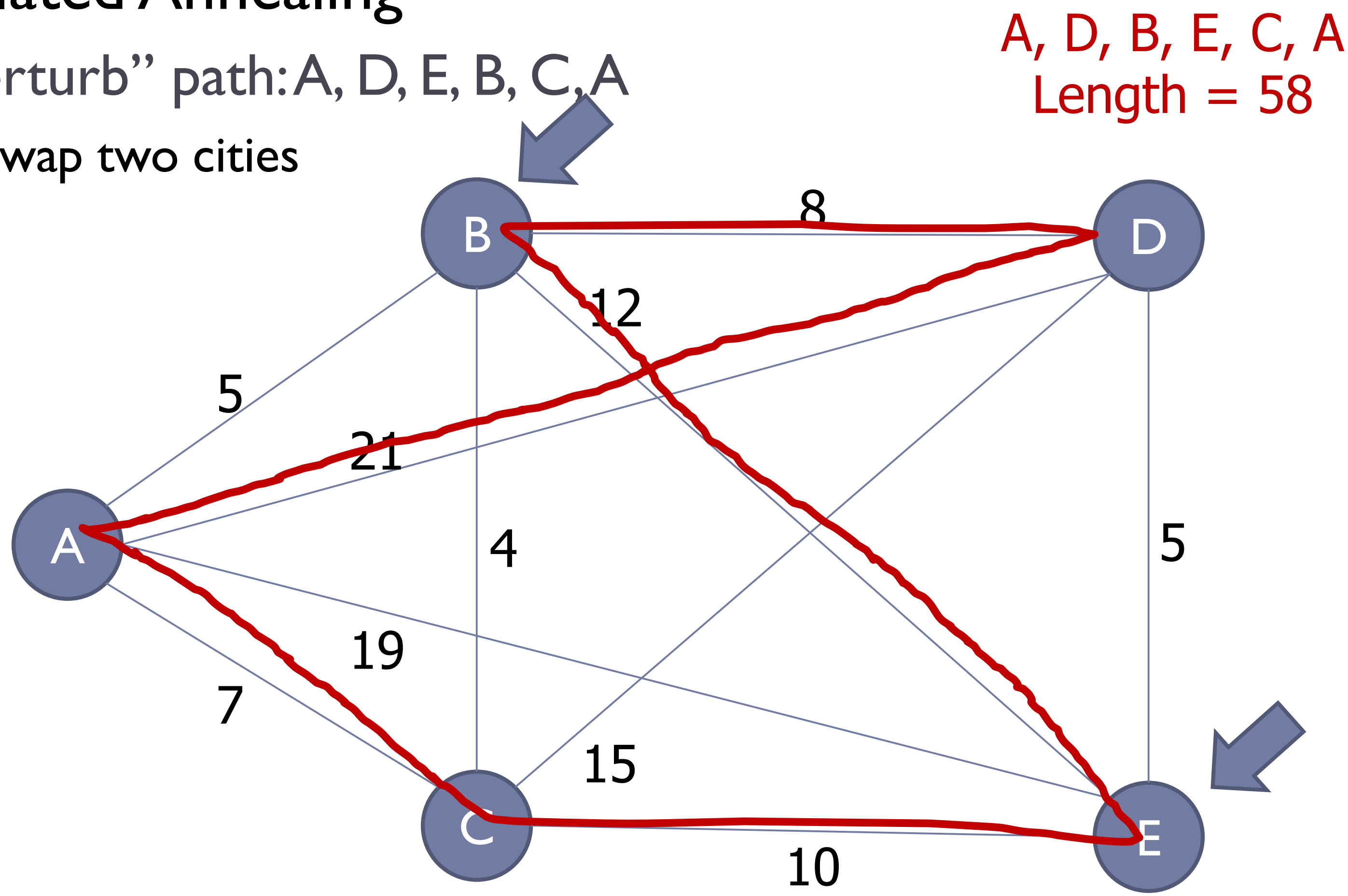
Length = 58



TSP Example

T = 0.8

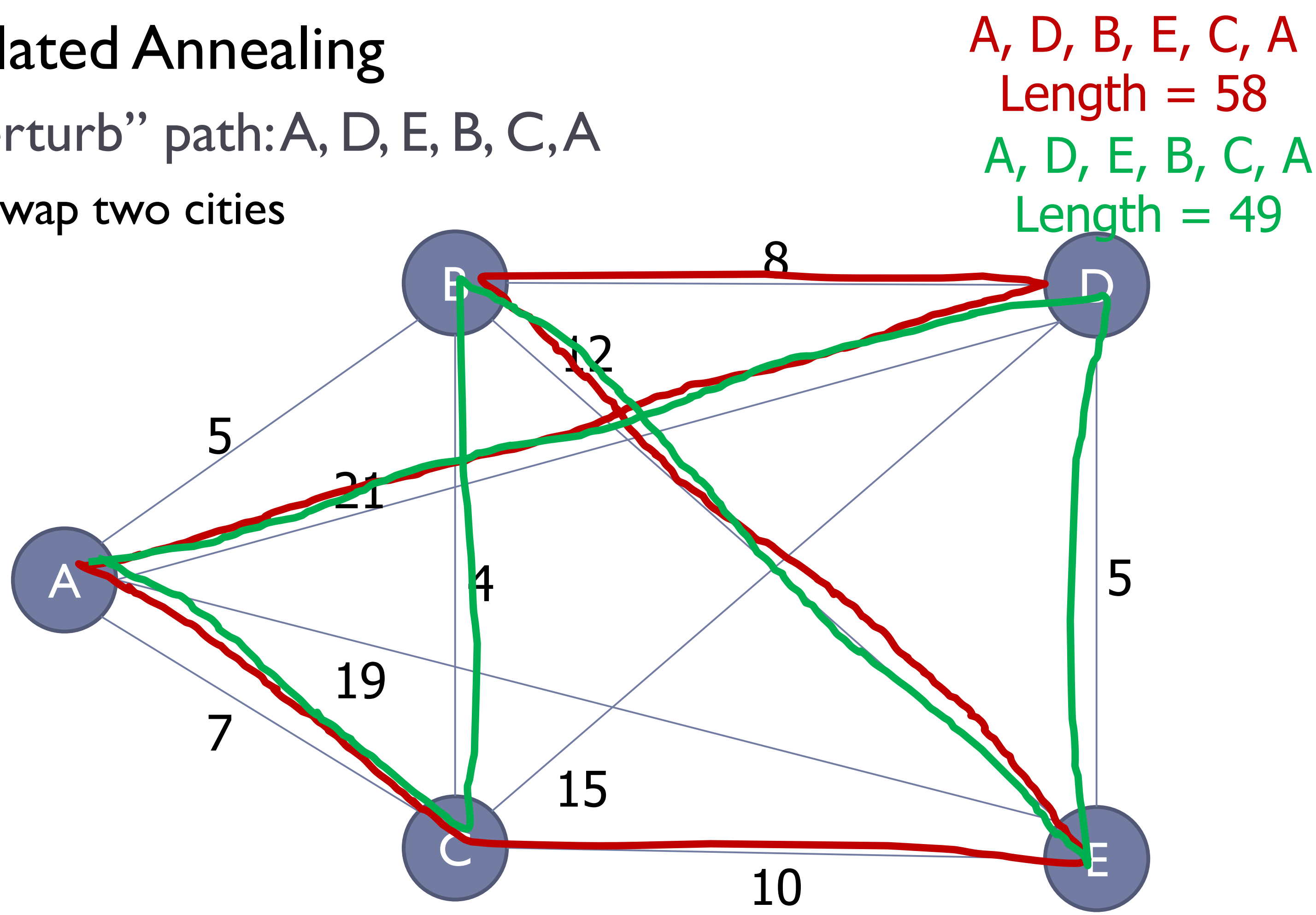
- Simulated Annealing
  - "Perturb" path: A, D, E, B, C, A
    - Swap two cities



TSP Example

T = 0.8

- Simulated Annealing
  - "Perturb" path: A, D, E, B, C, A
    - Swap two cities

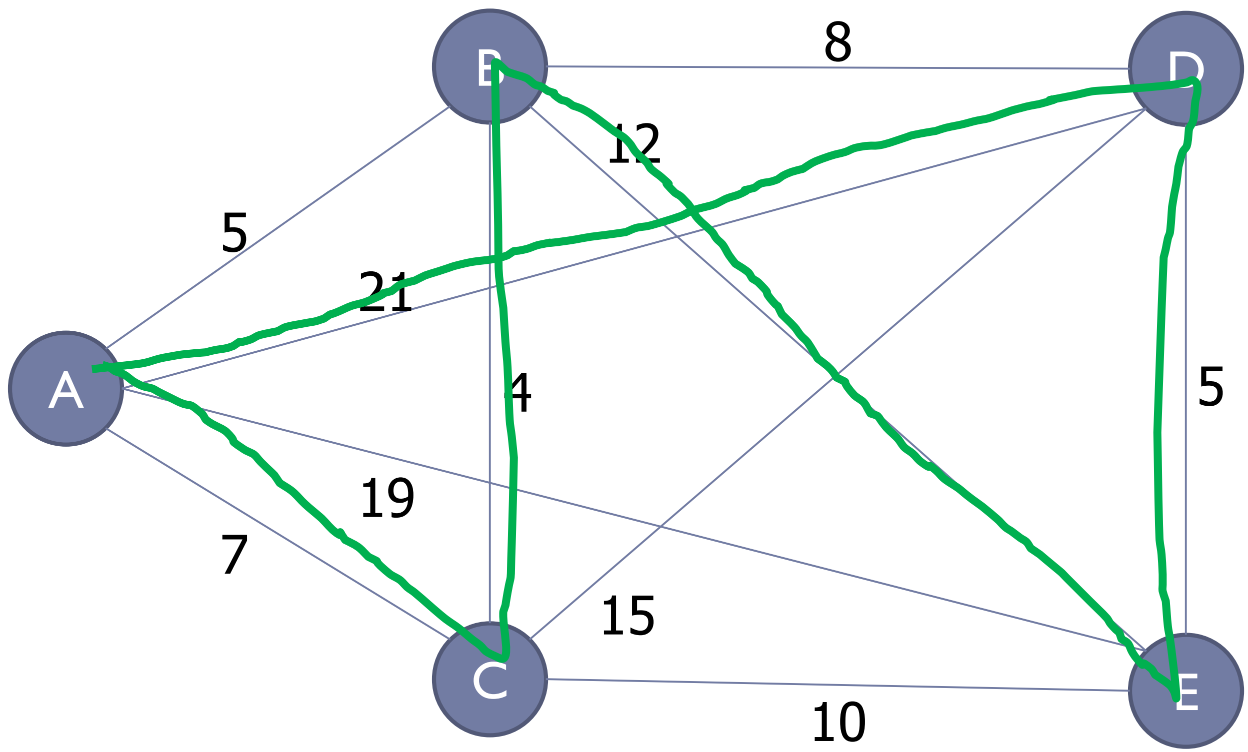


TSP Example

T = 0.8

A, D, E, B, C, A
Length = 49

- Simulated Annealing
  - Since new length is less, keep new path



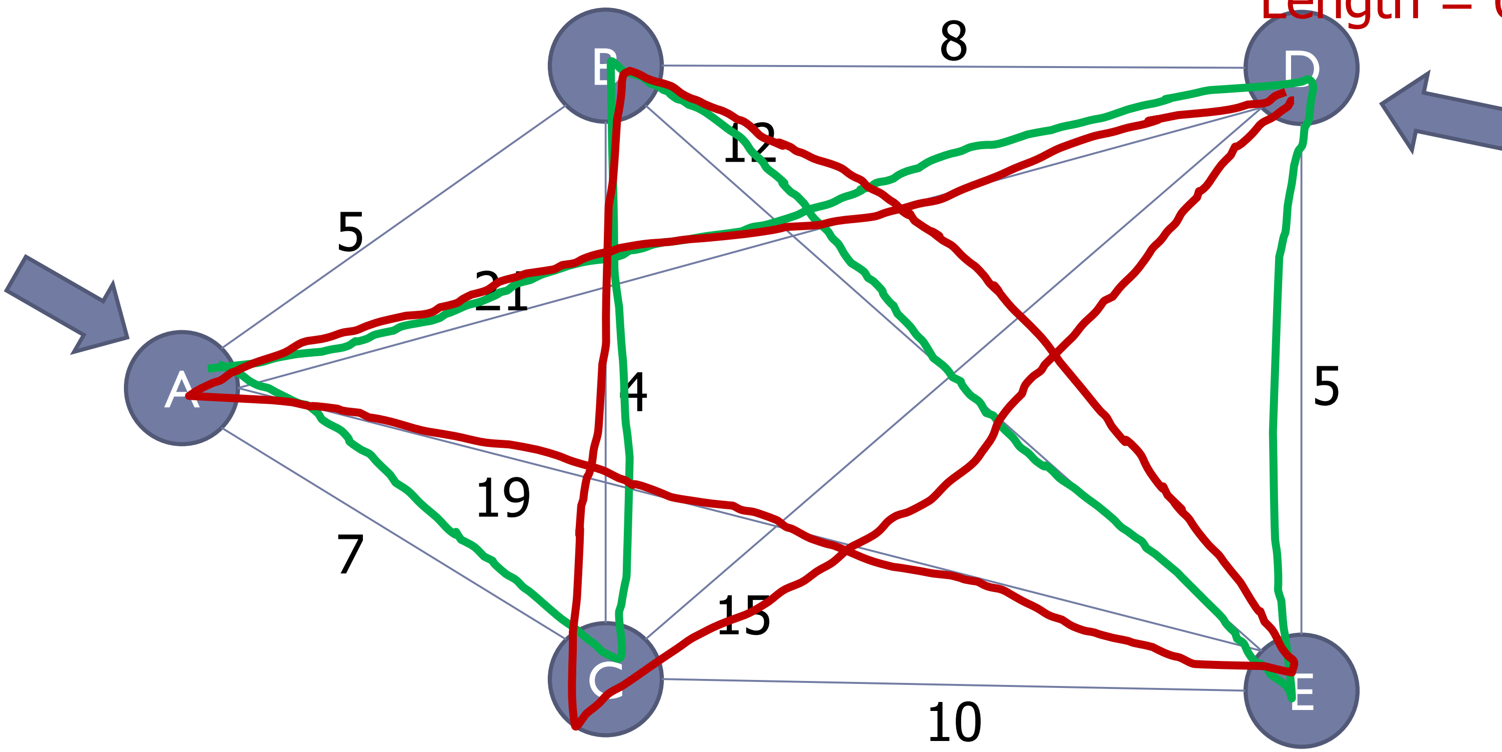
TSP Example

T = 0.6

- Simulated Annealing
  - "Perturb" path: D, A, E, B, C, D

A, D, E, B, C, A
Length = 49

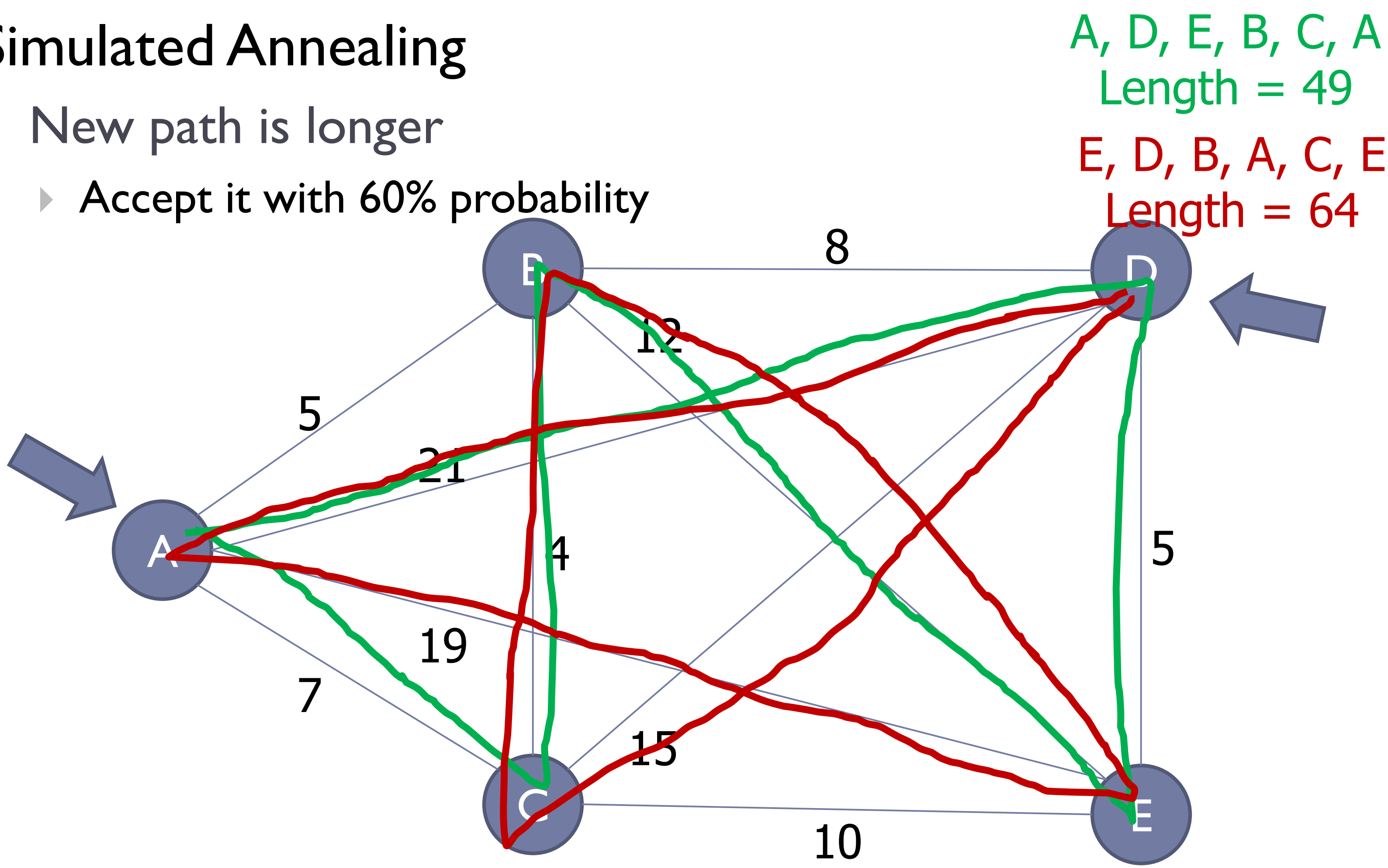
E, D, B, A, C, E
Length = 64



TSP Example

T = 0.6

- Simulated Annealing
  - New path is longer
    - Accept it with 60% probability



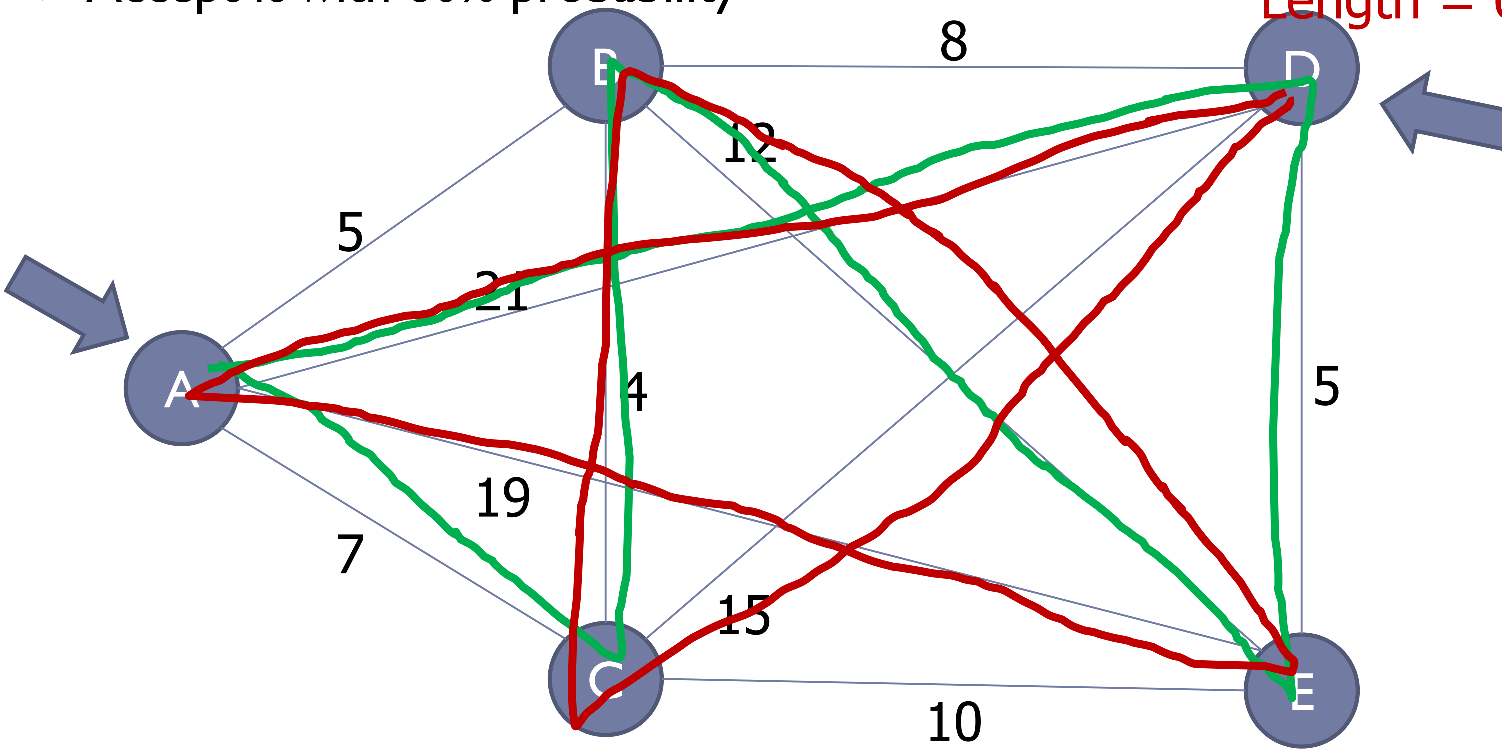
TSP Example

T = 0.6

- Simulated Annealing
  - New path is longer
    - Accept it with 60% probability

A, D, E, B, C, A
Length = 49

E, D, B, A, C, E
Length = 64



TSP Example

T = 0.4

- Simulated Annealing
  - Reject new path
  - "Perturb" path and repeat until T = 0 and no new better paths