ORIE 4580/5580: Simulation Modeling and Analysis

ORIE 5581: Monte Carlo Simulation

Unit 11: The Physics of Stochastic Processes

Sid Banerjee

School of ORIE, Cornell University

understanding complex stochastic systems

- most systems are designed based on solving static optimization problems
- stochastic models account for this uncertainty
- our intuition is not so good when there are uncertainties, especially when uncertainties interact and accumulate
- stochastic process physics: helps us understand and debug simulations

physics of stochastic systems

three major concepts:

1. phase transitions: small changes in parameters at *critical points* lead to large changes in outputs
2. conservation laws: in well-behaved (*stable*) systems, macroscopic quantities are conserved
3. fluctuations: uncertainty leads to perturbations of the system about its 'steady-state'

example 1: DTMC with absorption

the gambler's ruin chain

- two gamblers A and B start with $\$a$ and $\$b$ respectively
- each game is won by A with probability p_A and B with probability $p_B = 1 - p_A$
- whoever wins gets \$1 from the other player
- game ends when one player runs out of money

example: CTMC on infinite space

single-server M/M/1 queue

- number of servers: 1, queue capacity: infinite
- service discipline: first-come-first-served (FCFS or FIFO)
- interarrival times: exponential(λ)
- service times: exponential(μ)

phase transitions

small local imbalances lead to large global changes

phase transition in the gambler's ruin

- example: in the gambler's ruin, one interesting question is to understand the probability that A or B win, as a function of p_A

phase transition in the gambler's ruin

phase transition in queues

in a queuing system, we want to understand when the system is stable, i.e., if over a (infinitely) long time period, the average queue length stays bounded

capacity for work

for queue to be stable, average 'work' coming to the system must be less than the average work capacity of system

- example: we can view server to be working at a constant rate 1, and each agent can be thought of as bringing exponential(μ) amount of work

phase transition in queues

clicker question: minimum servers

a bank runs a call center for both day-to-day transactions and for financial advising. callers with day-to-day transactions call at rate 30 per hour and average call times are 5 minutes. callers asking about financial advising call at rate 5 per hour and average call times are 30 minutes. a single pool of agents handle all calls. what is the minimal number of agents needed?

(a) 3

(b) 5

(c) 6

(d) 7

(e) 10

conservation laws: mass conservation

flow conservation

in a stable queueing system, unless agents are spontaneously born/destroyed, the departure rate from a queue must equal the arrival rate into it over the long run.

- example: no matter how hard the baristas work, a coffee shop cannot serve customers faster than they arrive.
- exceptions:

flow conservation in networks

conservation laws: work conservation

Little's law

for stable queues, in the long run

average 'congestion' = arrival rate × average 'delay'

formally $\mathbb{E}[l] = \lambda \mathbb{E}[w]$, where:

- l = number of people in system
- w = total time in system for each person

- note 1: needs stability (i.e., flow conservation)
- note 2: you can use this for the system or the queue

Little's law: examples

Little's law

average congestion $=$ arrival rate $\times$ average delay

Little's law: intuition

Little's law

average congestion $=$ arrival rate $\times$ average delay

conservation laws: mean conservation

balanced gambler's ruin

in a balanced gambler's ruin setting (i.e., $p_A = 0.5$), the average per-round earning by A (or B) is zero

we can use this to compute ruin probabilities

more complex conservation laws

the 'law of delay'

if agents do not abandon, then delay explodes as arriving work approaches system capacity:

$$\text{delay} = \text{constant}/(\text{capacity - arrival rate})$$

- example: customers arrive (randomly) to a ticket window at rate 4.5 per minute, and service takes 12 seconds
  – queueing predicts average waiting time 2 minutes
  if the arrival rate increases just 10

clicker question: delay with abandonment

Lyft gets ride requests according to a PP at rate 130 per hour, and the average ride time is 10 minutes
passengers are willing to wait for an exponentially distributed time with mean 5 minutes before abandoning
the mean time spent by passengers on the app waiting over all passengers (including those that abandon) is

(a) at most 5 minutes

(b) somewhere between 5 and 10 minutes

(c) more than 10 minutes

(d) indeterminate

fluctuations in queues

the 'law of variability'

more variability in arrival or service time $\implies$ more congestion in the system

- example: computing jobs arrive (uniformly) to a server window at rate 4.5 per minute, and each job takes exactly 12 seconds
  - average waiting time = 0 minutes!
- in a manufacturing system, if we replace a manual process with a machine: queueing congestion can drop significantly even if the average times for both are the same!

mathematical queueing models

useful for rough-cut analysis when there is limited information
(typically arrival rates, service rates, # of servers, system size)

- most models approximate processes as exponential rvs
- exponential rvs are highly variable with a single parameter, the rate.
- exponential inter-arrival times make sense (poisson process)
  exponential service times have much less justification

queueing models are very useful for basic capacity and feasibility decisions, but do not give detailed results, transient behavior

standard queueing models

kendall's notation:

arrival dist|service dist|num servers(|capacity|num agents|discipline)

Kendall Notation	Characteristics
M/M/c	Single line, c parallel servers, unlimited waiting space, exponential distributions
M/M/c+M	Same as M/M/c but customers will only wait for an exponential amount of time before abandoning
M/M/c/N	Same as M/M/c, except waiting space is limited (N-c waiting customers)
M/G/1	Same as M/M/c, but only 1 server and service time distribution does not have to be exponential
M/G/∞	Same as M/M/c but an infinite number of servers (e.g., self-service system)