Course Description for CS522

Instructor: Dimitris Achlioptas

Hi all,

The title/topic of CS522 for this coming term is

Algorithms on Random Structures.

While "Algorithms" is clear, by "Random Structures" I mean Random Graphs (70%), Random Satisfiability Formulas (20%), and Random Other Stuff (10%) (e.g. n points chosen uniformly at random on the unit square – what is the expected cost of their TSP tour?). Like most things about the course's content these percentages are flexible: if random 3-SAT is simply "too much fun" :-) we can spend more time on that and less on, say, the Hamilton cycle problem.

What is this course?

Let me start by stating what the course is not.

It is not a "Randomized Algorithms" course: the input for the algorithms will not be arbitrary; instead, it will come from a known probability distribution. Also, the main focus will not be on running time per se, here "average-case", but instead on the special qualities of random structures and how these qualities relate to algorithmic analysis.

It is not a "Probabilistic Method/Random Graphs" course: the viewpoint is not existential but algorithmic; so, rather than asking "What is the chromatic number of a random graph?" we will ask "How well can a random graph be colored in polynomial time?"

We will introduce and use a number of Probabilistic (Method) tools. Some of these will be "classics" (e.g. the Lovász Local Lemma, the second moment method) while some will be more "cutting edge" (e.g. Talagrand's inequality).

OK, but what is the course about?

I think that the best way of answering this question is by examples of problems/topics that we'll cover in the course.

- Consider a set of m identical jobs and n identical machines, where any job can be served by any machine. Assume that each job chooses just two machines uniformly at random as its potential servers (ignoring the other $n-2$). Given these choices, how well can a centralized algorithm assign jobs to machines so as to minimize the maximum load?

  If $m = (k - o(k))n$ then with probability $1 - o(1)$ an assignment with maximum load k exists and can be found in linear-time.

  Exercise 1: formulate this as a graph theoretic problem. Hint: the graph has n vertices.

- Consider a set of n jobs and n machines where for each i, j the cost of assigning job i to machine j is a random real number c_{ij}, chosen uniformly and independently from $[0, 1]$. Let C_n denote the expected optimal (minimum) cost of assigning jobs to machines, i.e. the expected weight of a minimum weight perfect matching in the corresponding bipartite graph. Do you believe $C_n = O(1)$? How about

  $$\lim_{n\to\infty} C_n = \frac{\pi^2}{6} \ ?$$

  Anyhow, physicists knew all this since the mid-70s...

- Form a random graph $G(n, m)$ on n vertices by starting with n isolated vertices and adding m edges one-by-one, each edge connecting two randomly chosen vertices.

  - Can you find a maximum matching in such a graph? Yes.
  - Can you decide whether it has a Hamilton cycle? Yes.
  - Can you find a maximum clique? No.
  - How about an algorithm that returns a clique of size c times the maximum one? For $c \le 1/2$ it's trivial. For $c > 1/2$ we have no idea (and probably not a chance). Even fancy algorithms, like simulated annealing, provably fail.

- Consider a random graph on n vertices in which we prescribe the degree distribution. That is, we say how many vertices will have degree 1, how many will have degree 2, and so on, and then chose the edges between vertices randomly, while respecting the degree conditions.

  If you find this a more realistic model for random graphs, then you'll be excited to know that there are many nice things we can say about it. For example, we can use it to build error-correcting codes that are bandwidth-efficient and extremely fast to encode and decode (and led to a Silicon Valley startup).

- Did I mention that the analysis of these codes can be reduced to studying a system of elementary differential equations? Oh, and the same is true for the analysis of the load-balancing algorithm, the matching algorithm and the clique-finding algorithm in $G(n, m)$.

- OK, but can random graphs be used to explain the "six degrees of separation" phenomenon?

  For sure. They further suggest geometry as its crucial underpinning.

Hope to see you in class,
dimitris