CSC2420: Algorithm Design, Analysis and Theory **Fall 2017**

Lecture 1: Introductions

Lecturer: Allan Borodin, Nisarg Shah *Scribe: Lily Li*

1.1 Introductions

Allan is doing the first half. He is doing conceptually simple algorithms. The interest is to take a paradigm and put in a theoretical framework (outlining its capabilities and limitations and giving these things *definitions*). Nisarg is doing the second half of the course.

1.2 Graham's Online Algorithm and the Makespan Problem

These papers are old, we are talking 1966-ish. They precede definition of NP vs P but the ideas described are similar.

The makespan problem is similar to the load assignment/ bin packing problem (the simplified form is partition of course). n jobs $J_1, ..., J_n$ each job has a some execution time. There are m machines assigned to do the jobs, we wish to minimize the finish time.

Graham did some some competitive analysis (before the term was even defined!) for the greedy algorithm for the online version of this problem. Several different ways to do the analysis: worst case (adversarial), stochastic (assuming, usually, i.i.d), random order model (ROM) — adversary does some planning to form a distribution but nature gets to make random choices from distribution.

Possible extensions to the one pass paradigm allows for look-ahead, revoking rights, multiple passes, advice etc. Instead of extension, you could think of different measures of performance. Some of these measures are not at all trivial.

Back to the makespan problem. First greedy: where you put down the jobs into the currently most free machine. This results in a $2 - 1/m$ approximation (please formalize this). Possible improvement would be to sort in decreasing order. This results in a $4/3 - 1/(3m)$ approximation (please analyze this — hint: do this by contradiction then take a look at the smallest job, this job has to pretty big... something, something contradiction).

There are possibly better approximation schemes. There is even a PTAS (polynomial time approximation scheme) that is an $1 + \epsilon$ approximation, but of course, more accuracy means more time namely $\mathsf{Poly}(e^{1/\epsilon}, n)$ where n is the length of the description.

When you start sorting, the problem is no longer online. Approximation bound depends on m — the number of machines.

Further extensions of the problem include: different machines take different times (speed), or jobs can only be run on some machines (restricted machines), or precedence constraints (Graham actually showed that the ratio is still $2 - 1/m$, but there are some modifications to the problem, add more machines, reducing the time of jobs, removing restricts, that changes the calculations).

1.3 Knapsack Problem

1.3.1 Priority Algorithm Model

This is a way of discussing greedy algorithms. The intuition of these algorithm considers one input item in each iteration and at each iteration we make a myopic choice (using previously seen info).