CSC373S Lecture 2

- Last time we ended by claiming that a greedy algorithm (lets call it EFT for earliest finishing time) that sorts intervals by their finishing times (ties can be broken arbitrarily) and then accepts "greedily" is an optimal algorithm for the interval selection problem (= one machine interval scheduling).

  The text and the notes from CSC373 Fall 2010 suggest the following proof for the optimality. Let S_i be the set of intervals accepted by the end of the i^{th} iteration. We say S_i is *promising* if it can be extended to an optimal solution. Formally this means that for all i, there exists an optimal solution OPT_i such that $S_i \subseteq OPT_i \subseteq S_i \cup \{J(i+1), \ldots, J(n)\}$ and by induction (as in those notes) we prove that all S_i are promising which implies that S_n must be optimal.

- An alternative proof is by what is called a "charging argument". In this case, the charging argument informally wants to charge each interval of an optimal (or arbitrary solution) to a unique interval in the greedy solution. That is, let OPT be any (feasible) solution (we can think of it as an optimal solution) and S the solution of the EFT greedy algorithm. We want to define a 1-1 function $h : OPT \rightarrow S$. This would imply that $|OPT| \leq |S|$. Here then is how we can define h: Let $h(J)$ be that interval J' in S that intersects J and has the earliest finishing time amongst intervals in S intersecting J.
  First we claim tha h is a function (i.e. J' must exist and is unique). Why?
  Second we claim h is 1-1. Suppose both J_1 and J_2 in OPT are mapped to the same J' in S and and without loss of generality assume that $f_1 < f_2$ where f_1 (resp f_2) is the finishing time of $J(1)$ (resp. $J(2)$). Let f' be the finishing time of J'. By the definition of the mapping h, $f' \leq f_1$ or else the greedy EFT algorithm would have taken J_1 (and not J'). So we have $f' \leq f_1 < f_2$ and since J_1 and J_2 cannot intersect, J_2 cannot intersect J'.

- Both kinds of proofs can be utilized to prove optimality for other greedy algorithms (for other problems). The "promising argument" seems (to me) tailored for greedy style algorithms. The charging argument is generally applicable. Both arguments can be used to prove approxmation bounds (for non optimal algorithms) but I think a charging argument is a little easier to present for approximation results.

- A few comments about the other non optimal greedy algorithms for interval selection.

  Earliest start time (EST) is arbitrarily bad (ie approx ratio $n-1$ where $n =$ the number of intervals).
  Shortest processing time SPT is a 2-approximation algorithm; i.e. no solution can have more than twice the number of intervals found by SPT.

  Fewest conflicts is not opimal (example shows no better than 4/3 approx); tight approx is 3/2. I also note that the structure of this algorithm is somewhat different in that the order in which intervals are chosen is not fixed initially but rather changes adaptively depending on what intervals can no longer be taken.

- Lets consider a generalization of the interval selection problem, namely what is called the JISP problem (Job Interval Selection/Scheduling problem). An interval $J(i)$ now is repesented by a triple (s_i, f_i, c_i) where s_i and f_i are as before the starting time and finishing times for the interval. Now c_i is the job (or 'class') to which $J(i)$ belongs. Compatible (non conflicting) intervals must not intersect (as before) and furthermore cannot belong the the same job; that is, if $c_i = c_k$ then $J(i)$ and $J(k)$ are not compatible.
  Consider the same algorithm EFT but now with this more general notion of compatibility. In the problem set you are asked to show that EFT is a 2-approximation algorithm for the JISP problem. That is, for any solution OPT, and S the solution given by EFT (with the new def of compatible) we have $|OPT| \leq 2 \cdot |S|$.
  NOTE: JISP is an NP hard optimization problem and hence we cannot expect an optimal algorithm. In fact, it is know that it is NP-hard (in the worst case) to approximate JISP within approximation factor $(1 + \epsilon)$ for some small $\epsilon > 0$. (This will be in contrast to some problems which have PTAS and FPTAS algorithms as will be explained later.) Currently, the greedy EFT provides (as far as I know) the best approximation guarantee of any deterministic polynomial time algorithm.
  Motivation: Variable length courses are given once a week and are represented by intervals denoting their start and finishing times and the instructor (i.e. c_i). The goal is to choose a maximum number of non-intersecting courses without taking two courses from the same instructor.

- The next greedy algorithm we wish to consider is for what is (usually) called the interval coloring problem (and called interval partitioning in the Fall notes) . Namely, we are again given n intervals $J(1), \ldots, J(n)$ and now we wish to color all the intervals with as few colors as possible so that intervals having the same color do not intersect. (As a machine scheduling problem we want to schedule the intervals on as few machines as possible.)

  Surprisingly (perhaps) the ordering $s_1 \leq s_2 \ldots \leq s_n$ (which was arbitrarily bad for interval selection) now leads to an optimal greedy algorithm (lets call it EST for earliest starting time) for interval coloring. See slide 40 of the notes. The style of argument used to prove optimality is different here. Essentially, the argument is to show some intrinsic lower bound (in terms of some parameter of the input instance) for any allowable solution and then show that the greedy solution achieves this bound. In this case, the bound is the maximum number of intervals that can intersect at any point of time (called the "depth" d of a set of intervals in the notes). That is, $|G| = d \leq |OPT|$ where G is the set of colors used by the greedy algorithm EST and OPT is any solution. So an optimal algorithm must then use exactly d colors (as does the greedy algorithm).