CSC373S Lecture 3

- I was asked in office hours to do another charging argument proof. I will show that SPT is a 2-approximation algorithm by giving a 2-1 mapping $h : ARB \rightarrow SPT$ where ARB is any feasible solution (and in particular can be OPT) and SPT is the set of intervals selected by the SPT algorithm.

  Let $J \in ARB$. Define $h(J)$ to be the J' in SPT that intersects J and has the earliest finishing time. As before h is well defined. We now want to show that h is 2-1 (ie at most 2 intervals in ARB can be mapped to any $J' \in SPT$. Clearly if $J' \in ARB$ then no other $J \in ARB$ can intersect J'. Suppose J intersects J'; we cannot have J properly included in J' or else SPT would have take J and not taken J'. Hence, any interval J that intersects J' must intersect at an endpoint of J' and since there are only two endpoints (i.e. the start and finish times of J'), there can be at most two intervals in ARB that intersect J'.

- The interval selection, JISP and interval coloring problems can all be formulated in graph theoretic terms which will suggest more general problems for which we can apply these greedy algorithms.
  Namely, suppose we are given a set of intervals $J(1), \ldots, J(n)$. Let every interval $J(i)$ be represented by a node v_i, and (for the interval selection and coloring problems) let the edge set be $\{(v_i, v_j) | J(i)$ and $J(k)$ intersect $\}$. The resulting graph is called an interval graph. (That is, interval graphs are graphs that are induced by the intersection of a set of intervals.) The interval selection (resp. interval coloring) problem becomes an instance of what is known as the maximum independent set MIS problem (resp. graph coloring problem) for interval graphs. For arbitrary graphs, (subject to our standard religious complexity beliefs) MIS and graph coloring are hard to approximate within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$; that is, for arbitrary graphs and in the worst case, this is almost as bad an approximation as one can get since a factor of n is trivial in both cases.

- What makes these problem easy to solve for interval graphs? Is there a purely graph theoretic way to understand the optimality of the EFT (for interval selection) and EST (for interval coloring)?

  Consider the ordering $f_1 \leq f_2 \ldots \leq f_n$ for the interval selection problem. Graph theoretically consider the neighbourhood of the node v_1 representing interval $J(1)$. We note that all intersecting vertices represent intervals that must intersect the finish time f_1 of $J(1)$. Hence they all must intersect each other and hence are (graph theoretically speaking) adjacent. That is, they form a clique (in the graph induced by $v_2, \ldots v_n$). Once we remove v_1, all the neighbours of v_2 (intersecting intervals of $J(2)$ having removed $J(1)$) are also a clique (in the graph induced by $v_3, \ldots, v_n$) and more generally the neighbours of v_i are a clique (in the graph induced by $vi + 1, \ldots, v_n$).

  This property is abstracted to provide one of the definitions of a chordal graph: namely, $G = (V, E)$ is a chordal graph if there exists a "perfect elimination order" PEO $v_1, \ldots, v_n$ of the vertices such that $N(v_i)$ is a clique in the graph induced by $v_{i+1}, \ldots v_n$.

Claim: The greedy algorithm that orders vertices by a PEO and then greedily accepts vertices is an optimal algorithm for the MIS problem on chordal graphs. Similarly, the greedy algorithm that orders vertices by the reverse of a PEO and then colors vertices greedily (ie give out the lowest color possible at each iteration) is an optimal algorithm for coloring chordal graphs. Note that the ordering $s_1 \leq s_2 \ldots \leq s_n$ can be alternatively viewed (by reversing time) as the ordering $f_n \geq f_{n-1} \ldots \geq f_1$.)

- This is ,of course, only interesting if there are other classes of graphs (besides interval graphs) which are chordal.

  Claim: Every tree is an interval graph.

  An alternative (and standard way) to define chordal graphs is that they are graphs that do not have any induced cycles of length greater than 3; that is, every cycle (bigger than a triangle) has chords (and hence is triangulated). So we can start with any graph and add chords to any induced big cycles and arrive at a chordal graph. There are other alternative definitions for chordal graphs.

- For me, there is an additional reason why chordal graphs are of interest. Namely, they can be generalized (using the PEO definition) to include many other graphs, especially other geometric intersection graphs.

  We will say that a graph $G = (V, E)$ is *inductively k-independent* if there exists an ordering (call it a k-PEO) of the vertices $v_1, \ldots, v_n$ such that the vertices in $N(v_i)$ are covered by at most k cliques in the graph induced by $v_{i+1}, \ldots v_n$. This implies that (but is not equivalent to) $N(v_i)$ has at most k independent vertices in the graph induced by $v_{i+1}, \ldots v_n$.

  Claim: The JISP problem can be formulated as an MIS problem restricted to inductively 2-independent graphs; that is, similar to the interval selection problem, we form a graph representing conflicts by letting every interval $J(i)$ be represented by a node v_i, and let the edge set be $\{(v_i, v_j) | J(i)$ and $J(k)$ intersect or $c_i = c_k$ $\}$.

  The intersection graph of unit disks (in 2-space) is inductively 3-independent. The intersection graph of unit squares is inductively 2-independent.

  We can then generalize the 2-approximation result for JISP by stating: given an appropriate k-PEO ordering (eg the ordering induced by the earliest finishing time in the JISP problem, the ordering induced by the leftmost center of a unit disk) of the vertices, there is a greedy k-approximation algorithm for solving the MIS problem when restricted to inductively k-independent graphs.

- NOTE: We have just observed that we do not need the geometric representation for (one machine) interval selection once we have ordered the vertices by their finishing times (i.e. by a PEO). However, in the m-machine interval scheduling problem, question 3 on the assignment claims there is a difference between the first fit EFT and best fit EFT algorithms for m-machine interval scheduling. Here the input is a set of intervals and a given m; that is, m is part of the input and not fixed.

  1. First fit EFT
     Sort intervals so that $f_1 \leq f_2 \ldots \leq f_n$
     For $i : 1..n$
         Let $k = min_\ell : J(i)$ does not intersect intervals on machine $\ell; 0$ if no such ℓ
         $\sigma(i) := k$ % $\sigma(i)$ specifies if and on which machine interval $J(i)$ is scheduled
     End For
  2. Best fit EFT
     Sort intervals so that $f_1 \leq f_2 \ldots \leq f_n$
     For $k : 1..m$
         $e_k := -0$ % e_k specifies the latest completion for intervals on machine k
     End For
     For $i : 1..n$
         Let $k = argmin_\ell : s_i - e_\ell > 0$ or $k = 0$ if no such ℓ
         $\sigma(i) := k$ % $\sigma(i)$ specifies if and on which machine interval $J(i)$ is scheduled
     End For
  3. Fact: The m machine interval scheduling Best fit EFT algorithm does not generalize to the analogous problem (sometimes called maximum m colorable subgraph problem) for all chordal graphs as this problem is known to be NP-hard.
     Moral: The geometry can be important for some problems.