

CSC373: Lecture 4

Greedy Interval Colouring Algorithm

Interval Graphs

Graph MIS and graph colouring

Chordal graphs

The problem set

Fixed order vs adaptive order greedy

Interval colouring

- We will now consider a minimization problem; namely given a set of intervals, we want to colour all intervals so that intervals given the same colour do not intersect and the goal is to try to minimize the number of colours used.
- We could simply apply the optimal m -machine ISP for increasing m until we found the smallest m that is sufficient. (Note: This is a simple example of a ***polynomial time reduction*** which is an essential concept when we study NP-completeness.)

Greedy interval colouring

- Consider the *EST* (earliest starting time) for interval colouring. Namely, having sorted the intervals by non decreasing starting times, we assign each interval the smallest numbered colour that is feasible given the intervals already coloured. (Recall that *EST* is a terrible algorithm for ISP.) Note that this algorithm is “equivalent” to *LFT* (latest finishing time first).
- Theorem: *EST* is optimal for interval colouring

Greedy interval colouring

Greedy interval colouring

Sort intervals so that $s_1 \leq s_2 \dots \leq s_n$

For $i : 1..n$

Let $k = \min_{\ell} : \ell \neq \chi(j)$ for all $j < i$ such that the j^{th} interval intersects the i^{th} interval

$\chi(i) := k \% \text{ the } i^{\text{th}}$ interval is greedily coloured by the smallest non conflicting colour

End For

Proof of optimality (sketch)

- The proof technique we will use here is also one often used for proving approximations.
- The idea is to find some bound (or bounds) that *any solution* must satisfy and then relate that to the algorithm's solution.
- In this case, consider the maximum number of intervals in the input set that intersect at any given point. The number of colours must be at least this large.
- It remains to show that the greedy algorithm will never use more than this number of colours.

Why doesn't the greedy colouring algorithm exceed this intrinsic bound?

- Recall that we have sorted the intervals by non-decreasing starting time (i.e. earliest start time first).
- Consider the first time (say on some interval I) that the greedy algorithm would have used its maximum number k of colours. Then it must be that there are $k-1$ intervals intersecting I . But then these intersecting intervals must all include the start time s of interval I and hence there are k intervals intersecting at s .

Interval graphs

- There is a natural way to view the interval scheduling and colouring problems as graph problems.
- Let \mathcal{I} be a set of intervals. We can construct the **intersection graph** $G(\mathcal{I}) = (V, E)$ where $V = \mathcal{I}$ and (u, v) is an edge in E iff the intervals corresponding to u and v intersect. Any graph that is the intersection graph of a set of intervals is called an **interval graph**.

Graph MIS and colouring

- Let $G = (V,E)$ be a graph. The following two problems are known to be “NP- hard to approximate” for arbitrary graphs:
- Graph MIS: A subset V' of V is an independent set (aka stable set) in G if for all u,v in V' , (u,v) is not an edge in E . The maximum independent set problem is to find a maximum size independent set V' .
- Graph colouring: A function c mapping vertices to $\{1,2, \dots,k\}$ is a valid colouring of G if $c(u)$ is not equal to $c(v)$ for all (u,v) in E . The graph colouring problem is to find a valid colouring so as to minimize the number of colours k .

Efficient algorithms for interval graphs

- Given a set I of intervals, it is easy to construct its intersection graph $G(I)$. It is not obvious but given any graph G , there is a (linear time) algorithm to decide if G is an interval graph and if so to construct an interval representation.
- The MIS problem (resp. colouring problem) for interval graphs is the MIS (resp. colouring) problem for its intersection graph and hence these problems are efficiently solved for interval graphs. Is there a graph theoretic explanation? YES : interval graphs are chordal graphs.
- The minimum colouring number (chromatic number) of a graph is always at least the size of a maximum clique. The greedy interval colouring proof shows that for interval graphs the chromatic number = max clique size.

What are chordal graphs?

- Many equivalent ways to define chordal graphs. For our purposes, let's define chordal graphs $G = (V, E)$ as those having a ***perfect elimination ordering*** PEO of the vertices; that is an ordering $v(1), v(2), \dots, v(n)$ such that for all i , $\text{Nbhd}(v(i)) \cap \{v(i+1), \dots, v(n)\}$ is a clique.
- Note that ordering intervals by earliest finishing times will provide a PEO for the intersection graph of intervals and hence interval graphs are chordal.

More on chordal graphs

- We can abstract the arguments used for interval selection (resp. colouring) to show the optimality of greedy algorithms for any chordal graph using a PEO (reverse PEO) ordering.
- An equivalent (and initial) definition of chordal graphs are those which do not have any k -cycles (for $k > 3$) as induced subgraphs.
- What are and are not chordal graphs? For example a 4-cycle cannot be an interval graph.

Comments on the problem set

- Question 1 of the problem set discusses a greedy algorithm for graph colouring. It uses a breadth first search to determine the order of vertices being coloured. For arbitrary graphs, determining if a graph can be 3 coloured is NP hard.
- Question 2 discusses a scheduling problem related to the JISP problem as will be explained. We will first explain the notation.

Fixed order vs adaptive order greedy

- The algorithms for interval scheduling and colouring choose a (fixed) ordering of the input items (i.e. intervals) and then consider them in that order. The algorithm stated in question 2 of the problem set has a somewhat different structure than the algorithms for interval scheduling/colouring in that the order in which inputs (e.g. jobs) are considered is decided adaptively based on previous decisions.