

CSC2420: Lecture 5

- Today's agenda:
 1. Greedy interval colouring
 2. Abstracting when greedy algorithms work well

Optimal greedy interval colouring

- Whereas the greedy algorithm utilizing the ordering $s_1 \leq s_2 \dots \leq s_n$ has an arbitrarily bad way approximation ratio, the problem of interval colouring is optimally solved by greedily using this ordering; that is, if the colours are 1,2, ... the algorithm colours I_j using (say) the smallest non conflicting colour (introducing a new colour only whenever necessary).

Proof of greedy colouring optimality

- The proof follows the style used for the greedy makespan algorithm. Namely, when viewed as a graph colouring problem (in this case we have an interval graph),
 - it is clear that the chromatic number (i.e. the minimum number of colours needed) is at least as large as the graphs clique number (i.e. the size of a maximum clique).
 - The greedy colouring (with the EST ordering) will only introduce colour k if there is a clique of size k
- As a byproduct of this algorithm/proof, we obtain the result that interval graphs are “perfect graphs”.

Abstracting when greedy algorithms work well

- The previous comments are a starting point for considering results about intervals (and hence interval graphs) in more generality. In graph theoretic terms, the (weighted) interval selection problem is the (weighted) maximum independent set MIS problem for interval graphs. For arbitrary graphs, the MIS and graph colouring problems are hard to approximate to within a factor $n^{1-\epsilon}$ for any $\epsilon > 0$. How can we abstract the properties that make the previous greedy algorithms (and the extensions to revocable acceptances and stack algorithms) so successful?
- What other types of problems lend themselves to greedy algorithms?

K+1 Claw free graphs, chordal graphs and their extensions

- We introduce $k+1$ claw free graphs by some motivating examples.
- Weighted k -set packing problem: Given a collection of sets $\mathbf{C} = \{S_1, S_2, \dots, S_n\}$ over a universe $\mathbf{U} = \{e_1, e_2, \dots, e_m\}$ and weights $w_i = w(S_i)$. The goal is to find a disjoint sub-collection \mathbf{C}' so as to maximize $\sum_{S_i \in \mathbf{C}'} w_i$. When the size of all sets S_i is at most k this is the k -set packing problem. Weighted set packing is the underlying problem in single minded and general combinatorial auctions (CAs).
- Weighted interval selection with equal (=unit) length intervals.
- Weighted selection of axis oriented unit squares (unit disks) in the plane. Note: disk graphs are often considered as a wireless communication model.

Motivating problems as graphs

- All these problems can be naturally viewed as intersection graphs.
 - In set packing, the nodes are the sets and there is an edge between sets when the sets intersect.
 - In (unit) interval(resp. squares, disks) selection, the nodes are the intervals and an edge exists whenever the intervals (resp. squares,disks) geometrically intersect.
- The stated problems become the weighted MIS problem for these graphs. Similarly, we can define the natural colouring problems for these graphs.

$K+1$ claw free graphs

- A $k+1$ claw is a root node connected to $k+1$ (talon) independent nodes. A graph is $k+1$ claw free if it does not have a node induced $k+1$ claw. Equivalently, for any vertex, the induced subgraph of its neighbourhood has at most k independent vertices. Examples:
 - Line graphs are 3-claw free. Note 3 claw free graphs are simply called claw free graphs and they are of special importance.
 - In k -set packing graphs, the neighbourhood of *every* vertex has at most k independent vertices (i.e. the induced graph of the neighbourhood has independence number at most k).
 - In unit interval graphs, the neighbourhood of *every* vertex has at most 2 independent vertices.
 - In axis oriented unit square graphs, the neighbourhood of *every* vertex has at most 4 independent vertices since every intersection hits a corner of the square.
 - In unit disk graphs, the neighbourhood of *every* vertex has at most 5 independent vertices

Weighted MIS on $k+1$ claw free graphs

- The “natural” greedy algorithm for WMIS is to sort the vertices so that $w(v_1) \geq w(v_2) \dots \geq w(v_n)$
- It is easy to see that this algorithm provides a k -approximation for $k+1$ claw free graphs. The proof is a simple charging argument. We can define a mapping h : OPT into Greedy as follows: $h(v) = v$ if v in Greedy, else $\arg \max\{w(v') \text{ in Greedy} : (v, v') \in E\}$. We break ties by assigning to v' earliest in ordering. Claw free property implies that h is k to 1 . Greedy ordering implies that every v in OPT mapped to v' in Greedy has weight at most $w(v')$ weight in OPT.
- A similar proof shows that a simple local search algorithm also achieves a k -approx; namely, swap in v' if its conflicting set has weight less than $w(v')$.

Chordal graphs

- It is easy to see that (non unit) interval graphs are not $k+1$ claw free for any $k < n-1$. So how do we account for the optimality of the greedy algorithm (in the unweighted case) and the priority stack algorithm (in the weighted case)?
- There are many equivalent definitions of chordal graphs, one definition being that there are no node induced cycles other than triangles (i.e. every big cycle has a chord). The definition we use is that G is a chordal graph if it has a *perfect elimination ordering* (PEO) of vertices v_1, v_2, \dots, v_n satisfying the property that the induced subgraph $\{v_j \mid j > i \text{ and } v_j \text{ in Nbhd}(v_i)\}$ is a clique; i.e. has independence number 1.

Chordal graphs

- Note that the EFT ordering $f_1 \leq f_2 \dots \leq f_n$ is a PEO for interval graphs.
- Trees are chordal graphs where a PEO can be formed by peeling off leaves of the tree.
- A clique is trivially a chordal graph.
- A split graph consists of an the union of an independent set I and a clique K and some edges in $I \times K$

Interval graph results that extend to chordal graphs

- Using any PEO ordering and then greedily taking vertices that are not adjacent to previously taken vertices is an optimal algorithm for unweighted MIS.
- Using the reverse order of a PEO and then colouring greedily is an optimal colouring algorithm.
- Using a PEO ordering, the priority stack algorithm is an optimal algorithm for weighted MIS.
- BUT the unweighted m -machine unweighted interval scheduling algorithm does not extend to all chordal graphs. Problem is NP hard when m is part of input.

A common generalization of $k+1$ claw free graphs and chordal graphs.

- A graph is inductive k -independent if there is an ordering of the vertices v_1, \dots, v_n such that the induced subgraph $\{v_j \mid j > i \text{ and } v_j \text{ in } \text{Nbhd}(v_i)\}$ has independence number at most k . (Akcoglu et al, YeB)
- Clearly $k+1$ claw free graphs are inductive k -indep.
- The intersection graphs of unit squares (resp. unit disks) are inductive 2-independent (resp. 3-indep.) as seen by ordering according to (say) the x-axis.
- The intersection graphs of arbitrary size squares (resp. disks) are inductive 4-indep. (resp. 5-indep.)

More inductive k-independent classes

- It is not obvious but every planar graph is inductive 3-independent.
- The (W)JISP leads to another example of inductive 2-independent graphs. In this problem, an interval I_j is represented by (s_j, f_j, w_j, c_j) where c_j is the job class to which this interval belongs. Now a feasible set of intervals means that the intervals do not intersect and there is at most one interval in any job class. The (W)JISP problem is to choose a feasible set of intervals so as to maximize the weight (cardinality).

Generalizing the chordal graph algorithms

- The priority stack algorithm for weighted chordal graphs becomes a k -approximation for inductive k -independent graphs (using a k -EO).
- The greedy colouring algorithm for chordal graphs becomes a k -approximation for inductive k -independent graphs (using the reverse order of a k -EO).
- Note: I don't know if local search can yield same approx
- In many cases, there are better algorithms for specific classes of graphs. For example, by further exploiting geometric properties, there are PTAS algorithms for WMIS on planar graphs (Baker) and disk graphs (Nieberg et al and Erlebach et al).

Status of (W)JISP

- It follows that the simple greedy algorithm for unweighted JISP is a 2-approximation and the priority stack algorithm is 2-approximation for the weighted JISP. (Note that in the unweighted case, the stack algorithm becomes the greedy algorithm).
- There is a somewhat involved randomized algorithm that achieves approximation $e/(e-1) + \epsilon$ for any $\epsilon > 0$ but this algorithm while polynomial in the input description is exponential in $1/\epsilon$.
- For the weighted case, as far as I know there is no polynomial time c -approx algorithm for $c < 2$. And even for the unweighted case I do not know of a deterministic poly time alg with $c < 2$.

Independence systems

- We now consider another abstraction for problems where the natural greedy algorithm and simple local search give “good” approximations.
- Let $\mathbf{M} = (E, \mathbf{F})$, where E is a set of elements, \mathbf{F} is a collection of subsets over E . A set S in \mathbf{F} is called an independent set. A basis is a maximal size set in \mathbf{F} .
- An (hereditary) independence system satisfies the following properties:
 - 1) The empty set is in \mathbf{F}
 - 2) A subset of an independent set is independent.
- Admissability systems satisfy the weaker property
 - 2') If S is indep., then there exists x in S such that $S - \{x\}$ in \mathbf{F} .

Matroids and greedoids

- A matroid \mathbf{M} is an independence system if it also satisfies the following exchange property:
 - 3) If S, T are in \mathbf{F} and $|S| < |T|$, then there exists x in $T-S$ such that $S + \{x\}$ in \mathbf{F} .
- A greedoid is an admissability systems that also satisfies property 3)
- It follows that every basis has the same cardinality called the *rank* of the matroid (greedoid).

Many examples of matroids

- Uniform matroids: For any fixed k , all sets of cardinality at most k are independent.
- Partition matroids: Let E be partitioned into m sets E_1, \dots, E_m , then for fixed d_i , any set having at most d_i elements from E_i is independent.
- Transversal matroids: Let E be the union of possibly overlapping sets C_1, \dots, C_r , then S is independent if each e in S belongs to a unique C_i ; i.e. S is a system of representatives for the collection $\{C_i\}$.
- The independent vectors in a vector space.
- The acyclic subsets of edges in a graph are indep.

Greedy algorithm for maximizing a linear function subject to a matroid

- Suppose $\mathbf{M} = (E, \mathcal{F})$ is a matroid (greedoid) and let each element e in E have a weight. We want an indep. set (or basis) that maximizes the weight.
- The standard greedy algorithm greedily chooses an element of largest weight maintaining independence. The standard greedy algorithm is optimal for maximizing a linear function subject to a matroid (or greedoid). By negating all weights, the standard greedy algorithm that chooses the smallest weight elements first will optimally compute a basis of minimal weight.
- The proof of optimality is essentially the proof of optimality for Kruskal's MST algorithm.

Submodular functions

- Submodular set functions can be defined by the “marginal decreasing property”; that is, $f(S+x)-f(S) \geq f(T+x)-f(T)$ whenever S is a subset of T .
- f is monotone if $f(S) \leq f(T)$ when S is a subset of T . f is normalized $f(\text{emptyset}) = 0$. Unless otherwise stated we will assume f is normalized and monotone.
- Nemhauser et al show that the “standard greedy algorithm” (or simple local search) is a 2-approximation for maximizing a submodular function subject to a matroid constraint.

Matroid (greedoid) characterization

- The previous optimality result is due to Rado (for matroids). Edmonds proved a converse; namely, if the standard greedy algorithm maximizes every linear function over an independence system \mathcal{M} , then \mathcal{M} is a matroid. Later Korte and Lovasz proved a similar result for greedoids; namely, if the standard greedy algorithm maximizes every linear function over an admissability system \mathcal{M} , then \mathcal{M} is a greedoid.
- Another characterization of matroids is that of an independence system in which all bases have the same rank.

A parameterized independence system generalization

- A k -independence system is an independence system where for all bases B and B' , $|B| \leq k |B'|$.
- Jenkyns defined such systems and showed that the standard greedy algorithm for maximizing a linear (resp. submodular) function subject to a k -independence system yields a k (resp. $k+1$) approx.
- There is also a converse in that if the standard greedy algorithm is a k -approximation for all weight functions over an independence system \mathbf{M} , then \mathbf{M} is a k -independence system.