CSC2421: Online and Other Myopic Algorithms Fall 2025

Allan Borodin

October 22, 2025

Week 8

Annoucements

- Next week is reading week. We will not be meeting. Please use reading week to submit a proposal for your project.
- I will soon start posting questions for Assignment 2. Assignment 1 now available for submission on Markus and is due Friday (given the announced extension).

Todays agenda

- Finish up online colouring algorithms
- Priority algorithms for d-inductive and chordal graphs?
- Online fair allocation.

Colouring trees and Bipartite Graphs

For colouring trees we have the following (almost) optimal online result:

Theorem: Every tree can be online colored with $\lfloor \log n + 1 \rfloor$ colors by the *FirstFit* algorithm.

This then shows that the competitve ratio is $\frac{\lfloor \log n+1 \rfloor}{2}$ since trees are bipartite graphs and bipartite graphs are precisely the graphs that are 2-colourable. This is (almost exactly) a tight bound since it is shown that:

For every deterministic online colouring algorithm, there is a a graph that requires $\frac{\lfloor \log n \rfloor}{2}$ colours.
The lower bound follows from the following inductive statement that holds for any deterministic colouring algorithm ALG. :
For every deterministic algorithm and for all $k \geq 1$, the adversary can create disjoint trees $T_1, T_2, \ldots$ wuch that the akgorithm has used k colours to colour the roots of these trees T_i and the total number of nodes in all these trees is at most $2^k - 1$.

The lower bound proof for online tree colouring.

The base of the induction is $k = 1$ in which we have a single node tree. For the induction step, assume true for k and extend to $k + 1$. Let $T_1, T_2 \ldots, T_m$ be the trees whose roots are coloried with k different colors. Call thess colors S. Then after all the nodes corresponding to the trees $\{T_i\}$ have been colored, create a second list of trees $T'_1, T'_2, \ldots$. Then one of two cases can occur:

1. One of the T'_i trees is colored with a color not in S. Then we are done since the algorithm has used $k + 1$ colors and the combined size is at most i $2(2^k - 1) < 2^{k+1} - 1$.
2. The roots $r_1, r_2, \ldots$ of the T'_i trees are colored by the k colors in S. Then the adversary creates a new vertex connected to each of the r_i so that the algorithm must use a new colour. The total number of nodes in all the t_i and T'_i trees (including the new root) is at most $2(2^k - 1) + 1 = 2^{k+1} - 1$.

The lower bound on the competitive ratio follows since trees can be coloured with 2 colours and the algorithm has used $k = \log(n + 1)$ colours.

First Fit coloring on trees and bipartite graphs

Inspired and following the lower bound argument, in order for the adversary to Force First Fit to use k colors on trees it must create at least 2^{k-1} nodes. Hence First Fit is only using $k = \log n$ colours and 2 colours are required.

Trees are bipartite and bipartite graphs are clearly 2-colourable.

Perhaps suprisingly, in contrast to trees, there are bipartite graphs on which First Fit will use $\Omega(n)$ colors.

Lovász, Saks and Trotter provide a deterministic algorithm for colouring bipartite graphs with competitve ratio $\log n$ and this is the optimal ratio.

Online colouring of bipartite graphs

Consider the following algorithm, called CBIP, for online graph Coloring of BIpartite Graphs. When a vertex v arrives, CBIP computes the connected component C_v (so far) to which v belongs. Since the entire graph is bipartite, C_v is also bipartite. CBIP computes a partition of C_v into two blocks: S_v that contains v and $\widetilde{S}_v$ that does not contain v. Note: that the argument is symmetric for both sides of the graph.

Note that neighbors of v are only in .$\widetilde{S}_v$.
Let i denote the smallest colour that does not appear in $\widetilde{S}_v$. CBIP colors v with color i. We can show that CBIP colours the graph using a most $2 \log n$ colours and hence is $\log n$ competitve. The result follows from the following claim:
Theorem: Let $n(i)$ denote the minimum number of nodes that have to be presented to *CBIP* in order to force it to use color i for the first time. By induction on i, we can show that $n(i) \geq \lceil 2^{i/2} \rceil$.

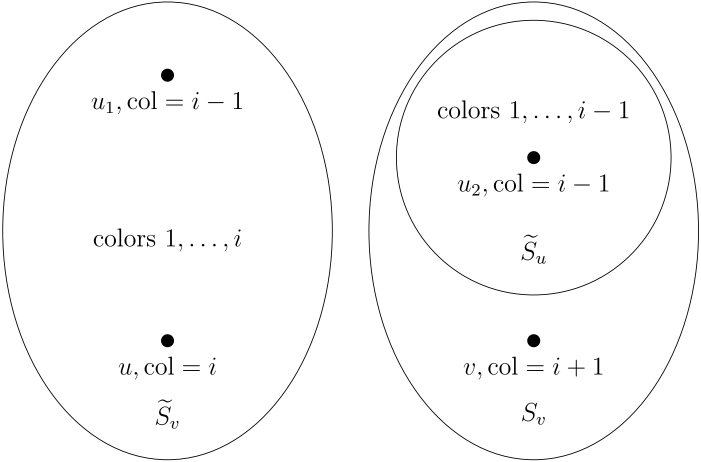
The induction for CBIP colouring of bipartite graphs

The base case is that $n(1) = 1$ and $n(2) = 2$. Assume claim is true for $i \geq 2$, show true for $i + 1$.
Let v be the first vertex that is colored with color $i + 1$ by *CBIP*. Consider C_v, S_v, and $\widetilde{S}_v$ as defined on the last slide. In particular, *all colors* $1, 2, \ldots, i$ appear among $\widetilde{S}_v$. Let u be a vertex in $\widetilde{S}_v$ that is colored i. Let $C_u, S_u, \widetilde{S}_u$ be defined as before, but now for the vertex u at the time that it appeared. Since u was assigned color i, then all colors $1, 2, \ldots, i - 1$ appeared in $\widetilde{S}_u$. Observe that $\widetilde{S}_u \subseteq S_v$.

Therefore, there exists vertex $u_1 \in \widetilde{S}_v$ colored $i - 1$ and there exists vertex $u_2 \in S_v$ colored $i - 1$, as well. Without loss of generality assume that u_1 arrived before u_2. At the time that u_2 was colored, the connected component C_{u_2} of u_2 had to be disjoint from the connected component C_{u_1} of u_1;, for otherwise u_2 would not have been colored with the same color as u_1.

Finishing the induction proof

Thus, we have $C_{u_1} \cap C_{u_2} = \varnothing$. Furthermore, we can apply the inductive assumption to each of C_{u_1} and C_{u_2} to get that $|C_{u_1}| \geq \lceil 2^{(i-1)/2} \rceil$ and $|C_{u_2}| \geq \lceil 2^{(i-1)/2} \rceil$. Thus, the number of vertices that have been presented prior to v is at least $|C_{u_1}| + |C_{u_2}| \geq 2\lceil 2^{(i-1)/2} \rceil \geq \lceil 2^{(i+1)/2} \rceil$.



Colouring d-inductive graphs

For now I am just going to state the theorem and state some immediate consequences. Then I will move to interval graphs. The following theorem is due to Irani [1994]

Theorem: *FirstFit* colours every d-inductive graph using at most $O(d \log n)$ colours.

As a consequence,

- *FirstFit* is $O(\log n)$ competitve on trees since trees are 1-inductive. Of course we know we have a tight competitive ratio of $\frac{\log n}{2}$ for trees. .
- *FirstFit* has competitive ratio $O(\log n)$ on planar graphs since planar graphs are 5-inductive.
- *FIrstFit* has competitive ratio $(\log n)$ on chordal graphs since chordal graphs are $\chi(G)$-inductive where $\chi(G)$ is the minimum number of colours to colour a graph.
- This will imply the same competitve ratio for interval graphs since interval graphs are a special case of chordal graphs. But as we shall see we can do much better for interval graphs.

Colouring interval graphs

We note that it is known that *FirstFit* colouring of interval graphs has competitive ratio between 5 and 8. This follows a series of results. We give the history in the Historical notes for chapter 7.

Kierstead and Trotter [1981] show there is deterministic online algorithm ALG that colours every interval graph G with at most $3\chi(G) - 2$ colours where $\chi(G)$ is the chromatic number of the graph. That is, ALG is 3-competitive.

The competitive ratio for ALG will follow fairly easily from the following result that will be proved by induction:

Theorem: Let $\omega(G)$ denote the clique number of a graph G and note that $\chi(G) \geq \omega(G)$. For all k, there is an algorithm $RECG_k$ (recursive greedy) that will color every interval graph with at most $3\omega(G) - 2$ colours.

The inductive construction of $RECG_k$

We will derive $RECG_k$ by induction on k. The base case $k = 1$ is trivial as G must be an independent set so that no two nodes are adjacent and hence $3 \cdot 1 - 2 = 1$ colour suffices.

Let G be an interval graph whose largest clique is k and consider the online input sequence of vertices $v_1 \prec v_2 \ldots \prec v_n$. The algorithm maintains a partition $V = A \cup B$ of vertices. When a vertex v_i arrives, the algorithm $RECG_k$ will place v_i in A if $\omega(A \cup \{v_i\}) < k$; otherwise, $RECG_k$ puts v_i in B. Any $v_i \in A$, it can be colored (by induction) by $RECG_{k-1}$ using $3k - 5$ colors. Any $v_i \in B$ is colored by $FirstFit$.

We will show that vertices in B can be colored using at most 3 colors larger that $3k - 5$ so that the total number of colors is $3k - 2$. In order to show that the 3 additional colours suffice, want to show that every $v \in B$ has at most two adjacent vertices in B.

It is convenient to think that we have an interval repreesentation of the graph but this will not be necessary. Using the interval repesentation we can see that there cannot be any triangles in B or else the clique number

Finishing the induction and using $RECG_k$ to colour any interval graph

It is convenient to think that we have an interval repreesentation of the graph but this will not be necessary. Using the interval repesentation we can see that there cannot be any triangles in B or else the clique number would be at least $k+1$. Similarly, no $v \in B$ can have three neighbours. This finishes the induction argument.

The algorithm starts using $RECG_1$. For every k, the algorithm continues to use $RECG_k$ until a new vertex causes a $k+1$ clique, and then the algorithm switches to $RECG_{k+1}$. Note that the algorithm only needs to be able to detect cliques and doesn't need the interval representation.

This implies that the competitive ratio is at most 3 since the colouring number $\chi(G) = \omega(G($, for interval (and chordal) graphs.

By induction, we can also show that for all k there exist interval graphs that require $3\omega(G) - 2$ colours so that 3 is the optimal competitive ratio.

Chordal graphs

As we have noted, interval graphs are a special case of chordal graphs. It is natural to ask if we can get a constant competitive ratio for all chordal graphs. In fact, Irani asked that question. It was solved by Albers and Schraink [2017,2021] who showed a randomized lower bound $\Omega(\log n)$ on the competitive ratio to colour some chordal graphs.

Albers and Schraink also showed an $\Omega(\log n)$ randomized lower bound on the competitive ratio for trees. Since trees are planar and also bipartite, the same $\Omega(\log n)$ randomized lower on the competitive ratio applies to planar graphs and bipartite graphs.

Finally, they show that for all d and n, there is a $\Omega(d \log n)$ randomized lower bound for colouring d-inductive graphs. Thus $\Theta(d \log n)$ is the optimal competitive ratio for arbitrary d-inductive graphs.

Priority colouring of graph classes?

It is interesting to note that both d-inductive and chordal graphs are defined in terms of the existence of a certain ordering on the vertices.

For d-inductive graphs, the definition is the ordering. For chordal graphs, the standard definition is that the graph does not have any induced cycles of length $\ell \geq 4$. They can also be characterized by having a *pefect elimination ordering (PEO)*.

If we consider reversing the ordering, we get that every d-inductive graph can be coloured with $d + 1$ colours. Of course, some graphs can be coloured with less colours. For example, planar graphs are 5-inductive while they can alwaye be 4-coloured.

And by reversing the perfect elimination ordering, every chordal graph can be coloured optimally with $\chi(G)$ colours. In particular, for interval graphs, the reverse of the PEO results in the same ordering as intervals sorted by non-decreasing starting times.

Priority colouring and MIS for d-inductive and chordal graphs?

Here is where I missopke in todays class. If I didn't have to worry about how to obtain the reverse of a PEO, the previous comment would show that there is an optimal priority algorithm for colouring any chordal graph.

It is also the case that the PEO provides an optimal fixed order priority algorithm for computing a maximum independent set in a chordal graph (i.e., the MIS problem). For example (as we know), sorting by non decreasing finishing times, we obtain an optimal fixed order priority algoritihm for interval selection.

I was thinking in terms of the interval repesentation of an interval graph. BUT, it is not at all clear how one can can do this from a graph representation (e.g., the VAM-PH input model). One might still say we have a greedy algorithm for optimally coloring any chordal graph since we can create the PEO and the reverse of the PEO within polynomial time but this doesn't satisfy our definition of a priority algorithm

Inductively independent graphs

An equivalent way to state a PEO ordering is to say that the induced graph of $Nbhd(v_i) \cap \{v_{i+1}, \ldots, v_n\}$ has at most one independent node (i.e., has independence number 1).

We can define a larger (than chordal) class of graphs as follows: A graph G is d-inductively independent if it has a d inductive indepedendence order $v_1, v_2, \ldots, v_n$ such that the induced graph of $Nbhd(v_i) \cap \{v_1, v_2, \ldots, v_n\}$ has independence degree d. Clearly, d-inductive graphs are a subclass of d-inductively-independent graphs.

There are many examples of d inductively independent graphs.

- Planar graphs are 3-inductively independent.
- The intersection of disk graphs (resp, unit disk graphs) are 5 (respectively 3) inductively independent. See Ye and Borodin [2012] for other examples.

Extending the results for chordal graphs, we obtain polynomial time (but noti necessarily priroity) d approximation algorithms for the MIS and colouring problem for d-inductively independent graphs.

Priority algorithms for trees and bipartite graphs

When considering offline priority algorithms for graphs, the VAM-FI is the appropriate input model in contrast to online algorithms where VAM-PH is the more approrate model.

In the VAM-FI model, it is not difficult to see that we can optimally 2-colour bipartite graphs (and therefore trees) by an adaptive priority algorithm. We can simply start coluring any node r with color 1.

Then the next nodes in the ordering are the neighbours N_r of r in a breadth first search. The nodes in N_r receive color 2. The (not yet coloured) neighbours of each node v in N_r becone the nodes next in the ordering, and they receieve color 1. We continue in this way to colour any bipartite graph with 2 colours.

If we ever try to give conflicting colours to any node previously coloured than we know that the graph is not bipartitie.

Moving on to Chapters 8 and 9

We will move from Chapter 7 (graph problems) to discuss a few of the results in Chapters 8 anbd 9. Chapter 8 considers extensions to three classic problems, namely ski rental, bin packing, and the k server problem.

After reading week, I intend to discuss a few ski rental extensions, namely the two closely related problems, capital investment and multi-slope ski rental, and then file migration.

I also intend to discuss renting servers in the cloud, an extension of bin packing.

But this week, I want to start discussing fair division and mainly allocating indivisible goods to agents. which is part of Chapter 9 where we discuss algorithmic mechansim design, and social choice theory.

Fair Division

Fair allocation of goods (or chores) is a relative new area for online algorithms. Fair division is part of the more general topic of social choice theory which itself is part of algorithmic social choice and mechanism design. In these areas, we have self interested agents. Mechansims design tries to balance these self interests with some more global objective(s).

We will focus on the area of online fair division and mainly on fair division of indivisible items.

In some cases, the mechanism will be making deccisions (e.g., what items to give to an agent); in other cases, agents may be making decisions (e,g, should I take an item) in which case the mechanism has to incentivize agents.

Suppose there are n agents. We let $\mathbf{A} = (A_1, A_2, \ldots, A_n)$ be an allocation of goods (or chores) to the agents. We will focus on goods being allocated. For divisible goods, A_i will be the fractions of the different items that agent i receives. For indivisible goods, A_i will be the subset of items that the agent receives.

Fairness criteria

There are a number prominent fairness concepts including:

- Envy-freenes (EF): Agent i envies agent j if $v_i(A_j) > v_i(A_i)$. An allocation is envy-free if no agent envies another agent. One might say that "fairness is in the eye of the beholder". Some view EF as the "gold standard" for fairness. Unfortunately, as we will see, it is rarely obtainable. So we will have to settle for some weakening of EF.
- Proportionality: If agent i has value $v_i(S)$ when allocated the entire set S of items, then the allocation A_i that agnet i receives has value $v_i(A_i) \geq \frac{v_i(S)}{n}$.
- Max-Min fairness: The objective is to maximize the minimum allocation to any agent. The max-min ojective has also been studied in combinatorial optimzation. In social choice theory it is often called *egalitarian social welfare*. In scheduling we consider the max-min allocation (rather than min-max scheduling in the makespan problem) to machines. Max-min scheduling has been called the Sanat Claus problem in scheduling it is not normally thought of as a fairness measure.

More fairness criteria

- Max-Min-Share (MMS). Agent i partitions S into $(A_1, A_2, \ldots, A_n)$. We define $MMS_i = \max_{A_1,\ldots,A_n}\{\min_j v_i(A_j)\}$. A partition $(A_1, \ldots, A_n)$ is an MMS partition (and hence considered fair) if $v_i(X_i) \geq MMS_i$ for all agents i. Why is this considered fair? Every agent reasons that there is a way to partition the goods for me to obtain my max-min share so I am entitled to at least this much value.
- The Nash Social Welfare NSW (or simply Nash Walfare) of an allocation $(A_1, A_2, \ldots, A_n)$ of items to n agents is defined as $\Pi_{i=1}^{n} v_i(A_i)$. A Max Nash Welfare (MNW) solution is an allocation that maximizes the Nash Welfare over all possible allocations. Why is this a fairness condition?

More fairness criteria

- Max-Min-Share (MMS). Agent i partitions S into $(A_1, A_2, \ldots, A_n)$. We define $MMS_i = \max_{A_1,\ldots,A_n}\{\min_j v_i(A_j)\}$. A partition $(A_1, \ldots, A_n)$ is an MMS partition (and hence considered fair) if $v_i(X_i) \geq MMS_i$ for all agents i. Why is this considered fair? Every agent reasons that there is a way to partition the goods for me to obtain my max-min share so I am entitled to at least this much value.
- The Nash Social Welfare NSW (or simply Nash Walfare) of an allocation $(A_1, A_2, \ldots, A_n)$ of items to n agents is defined as $\Pi_{i=1}^{n} v_i(A_i)$. A Max Nash Welfare (MNW) solution is an allocation that maximizes the Nash Welfare over all possible allocations. Why is this a fairness condition? A MNS solution implies some desireable fairness outcomes for indivisible goods including approximate MMS. Although, NP hard to compute the MNS, it can often be obtained in practice. See the publicly available site Spliddit (http //spliddit.org/) and the Caragiannis et al paper [2016,2019] where they state that the MNW solution is "arguably, the ultimate solution-for the division of indivisible goods".

EF implies Proportional and Proportional implies MMS

- EF implies Proportional. Suppose an allocation $(A_1, \ldots, A_n)$ was not proportional. Then some agent i, $v_i(A_i) < v_i(S)/n$. This implies that some agent j has reeived more than $v_i(S)/n$ so that i envies j and hence the allocation was not EF.
- Proportional implies MMS. For any agent i, their least share is $\leq v_i(S)/n$

The previous inequalities apply to divisible or indivisible goods. We will mainly focus on indivisible objects and additive valuations. But we have to at least mention what arguably might be the most well known fairness setting, namely cake cutting. As we probably all know, in cake cutting, we have n agents (e.g. birthday party attendees) that want their "fair" share of the cake. We will want to consriuder cake curtting in the online setting.

Indivisible goods, EF and EF1

We will assume n agents who are being allocated m indivisible items (goods). When we consider online allocations with the goods arriving online, it is more standard to say that we are allocating a sequence $t = 1, 2, \ldots T$ of online goods using T ws the number of items. T may or may not be known to the mechanism or the agents.

It is easy to see that it is impossible to have an EF solution offline or online. Consider two agent who both want a single item. That item goes to one agent leaving the other agent envious.

As we previously indicated, we will have to settle for some weakening of EF to hope to obtain some degree of fairness.

Some weakenings of EF

The following provide some degree of envy freeness

- EF1 (*Envy freeness up to one good*): An allocation is EF1 if for all agents i and j, the exists an item $x \in A_j$ such that $v_i(A_i) \geq v_i(A_j \setminus \{x\})$. That is, taking away some item from j's alloxcation removes the envy.
- EFX (*Envy freeness up to any good*): An allocation is EFX if for all agents i and j, and for all $x \in A_j$, $v_i(A_i) \geq v_i(A_j \setminus \{x\})$.
- Approximating EF: We can relax EF (and other fairness criteria) by allowing an approximation. That is, we may be quite satisfied if for all agents i and j, $v_i(A_i) \geq c \cdot v_i(A_j)$ for some approximation factor c that need not be a constant. For example, c might be a function of n or m. In many applications we expect that either n or m might be small so that such approximations caan be useful.
- Bounded EF: There exists a constant b such that for all i and j, $v_i(A_i) \geq v_i(A_j) - b$.

Comments about EF1 and MMS

- EF1 is obtainable offline for all monotone valuations (i.e. $v_i(A_i) \leq v_i(A_i \cup \{x\})$; i.e., "free disposal". Lipton et al [2004]. For additive valuations, the *round robin mechanism* obtains EF1.
- MMS cannot be obtained offline. Procaccia and Wang [2-4] show that even for additive valuations, for $n \geq 3$ agents there cannot be a perfect MMS solution. Feige et al [2021] provide the current best $\frac{39}{40}$ inpproximation for an example with 3 agents and 9 items. The best offline approximation is $\frac{10}{13} \approx .7692$ by Heidari et al [2025] following the previous approximation of $\frac{3}{4} + \frac{3}{3836}$ by Akrami and Garg [2024].
- The situation gets much worse for online allocations even for additive valuations. Benade et al [2025] show that regret must grow with the number T of online items which as a consequence shows that no bounded EF is possible online. He at al [2019] show that, even for two agents, in order to obtain EF1 in an online sequence of T items, there must be $\Omega(T)$ re-allocations of items. We will discuss these negative results after presenting some positive results.

End of Week 8 slides

We will return to EF, EF1 and other fairness results in our next class after reading week on Wednesday, November 5.