Lecture Notes CS:5360 Randomized Algorithms
Lecture 1: Aug 21, 2018
Scribe: Sriram V. Pemmaraju

1 Introduction

Max-Coloring. The similarity between the interval coloring problem and the max-coloring problem can be best understood by casting these problems into a geometric setting as rectangle packing problems. Start with an interval representation $\{I_v \mid v \in V\}$ of the given interval graph $G = (V, E)$[1]. Interpret each weight $w(v)$ as the height of interval I_v. In other words, the instance of the problem consists of axis-parallel rectangles $\{R_v \mid v \in V\}$, such that the projection of R_v on the x-axis is I_v and the height of R_v is $w(v)$. Each rectangle can be slid up or down but not sideways; all rectangles have to occupy the positive quadrant; and the regions of the plane they occupy have to be pairwise disjoint. Given these constraints, the interval coloring problem is equivalent to the problem of packing these rectangles so as to minimize the y-coordinate of the highest point contained in any rectangle. The max-coloring problem seeks a packing of the rectangles into disjoint horizontal strips $S_i = \{(x, y) \mid x \geq 0, \ell_i \leq y \leq u_i\}$, denoted by (ℓ_i, u_i). The constraints are that every rectangle is completely contained in some strip and for any two rectangles R_u and R_v in a strip, their projections on the x-axis I_u and I_v are disjoint. Given these constraints, the max-coloring problem seeks a packing of the rectangles into strips so that the total height $\sum(u_i - \ell_i)$ of the strips is minimized. Figure 1 shows two rectangle packings of a set of rectangles; the packing on the left is optimal for the interval coloring problem and the packing on the right is optimal for the max-coloring problem.

2 Approximation algorithms for max-coloring

In an instance of the *on-line graph coloring problem*, vertices of a graph are presented one at a time and when a vertex is presented, all edges connecting that vertex to previously presented vertices are also revealed. Each vertex must be assigned a color immediately after it has been presented (and before the next vertex is presented) and a color assigned to a vertex cannot be changed later. An algorithm for the on-line graph coloring problem assigns colors to vertices in the manner described above, so as to construct a proper vertex coloring of the graph. We say that an algorithm A for the on-line graph coloring problem k-colors a graph G, if no matter which order the vertices of G are presented in, A uses at most k colors to color G.

Let A be an algorithm for the on-line graph coloring problem. We use A as a "black-box" to devise a simple algorithm for the max-coloring problem. The algorithm, called `MCA` (short for max-coloring algorithm) is given below.

[1]Without loss of generality, we assume that the input to our algorithms is a set of weighted intervals. This is because there are many linear-time algorithms for recognizing interval graphs and most of these return an interval representation of the given graph, if it is an interval graph. See [3] for a recent algorithm.

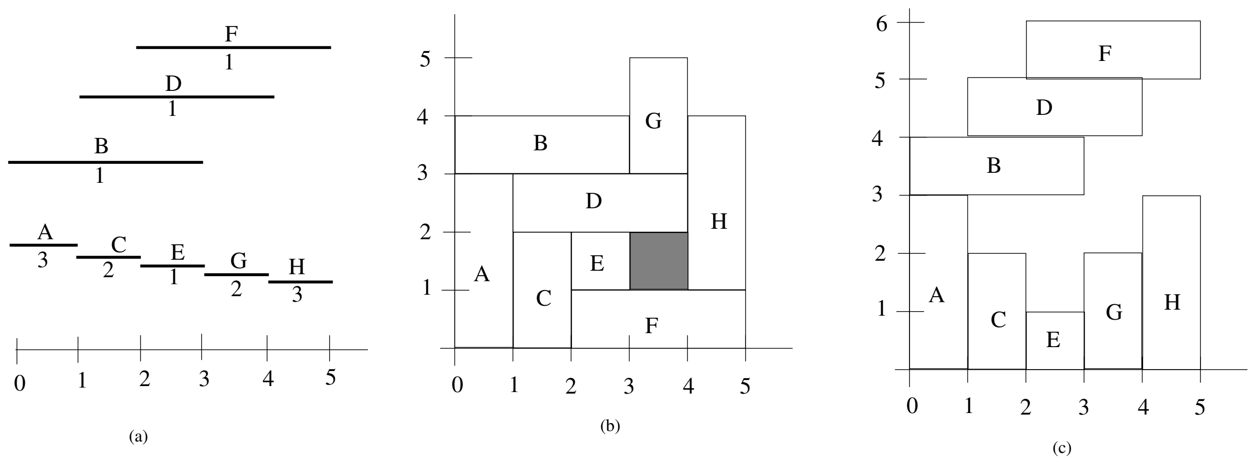

Figure 1: (a) is an interval representation of an interval graph; the "name" and weight of each interval are shown. (b) is a rectangle packing of this interval graph corresponding to an optimal interval coloring, with weight 5. This figure is from [2]. (c) is a rectangle packing corresponding to an optimal max-coloring. In the packing on the right the rectangles are packed into 4 strips: $S_1 = (0, 3)$, $S_2 = (3, 4)$, $S_3 = (4, 5)$, and $S_4 = (5, 6)$, for a total weight of 6.

Algorithm 1: 2-RULING SET(Graph $G = (V, E)$, $\varepsilon > 0$):

$I \leftarrow \emptyset$; $S \leftarrow V$;
for *each scale* $t = 1, 2, \ldots, \lceil \log \Delta \rceil$ **do**
  Let $\Delta_t = \frac{\Delta}{2^{t-1}}$; $S_t \leftarrow S$;
  for *iteration* $i = 1, 2, \ldots, \lceil c \cdot \log^{1/2+\varepsilon} n \rceil$ **do**
    Each $v \in S$ marks itself and joins $M_{i,t}$ with probability $\frac{1}{\Delta_t \cdot \log^{\varepsilon} n}$;
    if *$v \in S$ is unmarked and a neighbor in S is marked* **then**
      v joins $W_{i,t}$;
    end
    $S \leftarrow S \setminus (M_{i,t} \cup W_{i,t})$;
  end
  $B_t \leftarrow \{v \in S \mid deg_S(v) \geq \Delta_t/2\}$;
  $S \leftarrow S \setminus B_t$;
  $I \leftarrow I \cup$ GREEDYRULINGSET$(G[S_t], B_t, 2)$;
end
$I \leftarrow I \cup (\cup_t \cup_i$ GREEDYRULINGSET$(G[S_t], M_{i,t}, 1))$;
return

We will now make a connection between the number of colors used by A and the weight of the coloring produced by `MCA`. This connection, along with known results on on-line coloring of interval graphs will lead to constant factor approximation algorithms for max-coloring for interval graphs.

Theorem 1 *Let $\mathcal{C}$ be a hereditary class[2] of graphs and let A be an algorithm for on-line graph coloring such that for some integer constant $c > 0$ and for any graph $G \in \mathcal{C}$, A colors G with at most $c \cdot \chi(G)$ colors. Then, for any $G \in \mathcal{C}$ and for any weight function $w : V(G) \rightarrow \mathbf{N}$,* `MCA` *produces a coloring for G whose weight is at most $c \cdot OPT_M(G)$.*

[2] A class $\mathcal{C}$ of graphs is hereditary if $G \in \mathcal{C}$ implies that every induced subgraph of G is also in $\mathcal{C}$.

Proof: Let $C_1, C_2, \ldots, C_k$ be a coloring of G that is optimal for the max-coloring problem. Let $w_i = \max_{v \in C_i} w(v)$ and without loss of generality assume that $w_1 \geq w_2 \geq \cdots \geq w_k$. Now note that $k \geq \chi(G)$ and $OPT_M(G) = \sum_{i=1}^{k} w_i$. Let $A_1, A_2, \ldots, A_t$ be the coloring of G produced by MCA. Let $a_i = \max_{v \in A_i} w(v)$ and without loss of generality assume that $a_1 \geq a_2 \geq \cdots \geq a_t$. From our hypothesis it follows that $t \leq c \cdot \chi(G) \leq c \cdot k$. For notational convenience, define sets $A_{t+1} = A_{t+2} = \ldots = A_{c \cdot \chi(G)} = \emptyset$ and let $a_i = 0$ for i, $t < i \leq c \cdot \chi(G)$. We will now claim that for each i, $1 \leq i \leq k$, and each j, $c(i-1) < j \leq c \cdot i$, we have $w_i \geq a_j$. Showing this would imply the result we seek because the coloring produced by MCA has weight

$$\sum_{\ell=1}^{c \cdot \chi(G)} a_\ell = \sum_{i=1}^{\chi(G)} \sum_{j=c(i-1)+1}^{c \cdot i} a_j \leq \sum_{i=1}^{\chi(G)} c \cdot w_i \leq c \cdot OPT_M(G).$$

Since w_1 is the maximum weight of any vertex in G, the claim is trivially true for $i = 1$. For any $i \geq 2$, let $V_i \subseteq V$ be defined as $V_i = \{v \mid w(v) > w_i\}$. The coloring $C_1, C_2, \ldots, C_k$ of G, restricted to V_i is an $(i-1)$-coloring of $G[V_i]$, the subgraph of G induced by V_i. Because of the order in which vertices are presented to A, all vertices in V_i are presented to A before any vertex with weight w_i. Therefore, by our hypothesis, algorithm A colors $G[V_i]$ with no more than $c \cdot (i-1)$ colors. Therefore, the weight of the heaviest vertex in color classes A_j for j, $c(i-1) < j \leq c \cdot i - 1$ is at most w_i. $\square$

From this and the induction hypothesis, it follows that

$$\begin{aligned}
\rho_e(i) &= \rho_e(j,i) + \rho_e(j-1) \\
&\geq \frac{1}{4}(i-j+1) + \frac{1}{4}(\rho_e(j-1) + \phi_e(j-1)) \\
&\geq \frac{1}{4}(\rho_e(j,i) + \phi_e(j,i)) + \frac{1}{4}(\rho_e(j-1) + \phi_e(j-1)) \\
&= \frac{1}{4}(\rho_e(i) + \phi_e(i))
\end{aligned}$$

Acknowledgements.

We thank Narayanaswamy and Subhash Babu for letting us present their clever improvement [4] of our analysis of First-Fit in the preliminary version [5]. We also thank Brightwell, Kierstead, and Trotter for sharing their improved analysis [1] with us. Finally, we thank the anonymous referees whose suggestions have improved the paper.

References

[1] G. Brightwell, H. A. Kierstead, and W. T. Trotter. A note on the first-fit coloring of interval graphs. personal communication, 2003.

[2] Adam L. Buchsbaum, Howard Karloff, Claire Kenyon, Nick Reingold, and Mikkel Thorup. OPT versus LOAD in dynamic storage allocation. *SIAM J. Comput.*, 33(3):632–646, 2004.

[3] D.G. Corneil, S. Olariu, and L. Stewart. The ultimate interval graph recognition algorithm? (extended abstract). In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 175–180, 1998.

[4] N.S. Narayanaswamy and R. Subhash Babu. Analysis of first-fit coloring of interval graphs. personal communication, 2004.

[5] S.V. Pemmaraju, R. Raman, and K. Varadarajan. Buffer minimization using max-coloring. In *Proceedings of The ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 562–571, 2004.