Informatique théorique et applications

Maciej Ślusarek

Optimal on-line coloring of circular arc graphs

Informatique théorique et applications, tome 29, nº 5 (1995), p. 423-429

<http://www.numdam.org/item?id=ITA_1995__29_5_423_0>

© AFCET, 1995, tous droits réservés.

L'accès aux archives de la revue « Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/conditions). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

Numdam

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 29, n° 5, 1995, p. 423 à 429)

OPTIMAL ON-LINE COLORING OF CIRCULAR ARC GRAPHS (*)

by Maciej Ślusarek ([1])

Communicated by W. Rytter

Abstract. – *We show that a certain optimal on-line coloring algorithm for interval graphs given independently by Kierstead and Ślusarek can be applied to a wider class of circular arc graphs. We prove that the competitive ratio of the algorithm is equal to 3 which improves a previous result by Marathe, Hunt and Ravi.*

Résumé. – *Nous montrons qu'un certain algorithme de coloriage optimal on-line pour les graphes d'intervalles donné indépendamment par Kierstead et Ślusarek, peut être appliqué à une classe plus large de graphes circulaires d'arcs. Nous montrons que l'efficacité de l'algorithme (rapport entre le nombre de couleurs utilisées dans l'algorithme et dans l'algorithme optimal) est 3, ce qui améliore un résultat précédent de Marathe, Hunt et Ravi.*

1. INTRODUCTION

Assume that a finite family $F = (A_1, \ldots, A_n)$ of closed arcs on a unit circle is given. By a *circular arc graph* represented by F we mean an undirected graph $G(F) = (V, E)$ where the set of vertices V is equal to F and two arcs A_i, A_j are adjacent iff they intersect. We are concerned with a well known problem of graph coloring, and the class of circular arc graphs is of particular interest. A *coloring of a graph* is a function which assigns to each vertex a positive integer (color) in such a way that every two adjacent vertices are assigned different colors. The coloring problem consists in finding a coloring which uses the smallest possible numbers of colors.

While the general version of coloring problem is widely known to be NP-hard, for certain classes of graphs it is not difficult. In particular coloring of the class of circular arc graphs is also NP-hard [2], but is polynomial for a

Research done while the author was visiting Warsaw University, Institute of Informatics. Supported by Project ALTEC.
(*) Received June 9, 1994; revised August 20, 1994; accepted September 21, 1994.
([1]) Department of Computer Science, Jagiellonian University ul. Nawojki 11, 30-072 Kraków, Poland.

0988-3754/95/05/$ 4.00/© AFCET-Gauthier-Villars

subclass of interval graphs. An *interval graph* is formed in the same way as a circular arc graph but instead of arcs we have closed intervals on the real line. It is easy to observe that when there exists a point on a circle which is not covered by any arc from a family F then $G(F)$ is an interval graph.

On-line coloring of a graph is a version of the coloring problem in which we impose restrictions on the class of algorithms considered. The input to the coloring algorithm $\mathcal{A}$ is a sequence of vertices $v_1, \ldots, v_n$. We say that $\mathcal{A}$ colors the vertices *on-line* if for each i it assigns a color to v_i before $v_{i+1}, \ldots, v_n$ are input. In other words, when v_i is being colored $\mathcal{A}$ is given only the induced subgraph generated by $v_1, \ldots, v_i$. Of course, once assigned the color cannot be changed later on. The efficiency of the algorithm, called the competitive ratio, is measured in terms of a ratio of the number of colors used over the optimal (in usual off-line sense) number of colors (the worst-case ratio).

For interval graphs it has been shown that there exists an on-line algorithm achieving competitive ratio of 3 and that it is the best possible [1, 3, 4, 6]. Since interval graphs form a subclass of circular arc graphs, the lower bound for on-line coloring of the former holds also for the latter.

As concerns the upper bound so far the best result was given in [5]. Using that approach one starts by choosing any point x on the circle. Then the arcs are processed one by one. Those which contain x are colored using first-fit, the others are dealt with by an optimal coloring algorithm for interval graphs, using separate set of colors. Since first-fit for the overlapping arcs uses one distinct color for each arc and the optimal coloring of the rest is 3-competitive it is not hard to see that the method has competitive ratio of 4.

In this paper we show that the on-line interval coloring algorithm from [3, 6] can be also used to properly color circular arcs and that its competitive ratio remains the same – namely 3. In other words, the best achievable efficiency of on-line coloring is the same for arcs and intervals. This contrasts with the off-line versions for which there is a huge gap between the complexities: polynomial for intervals and NP-hard for arcs.

2. THE COLORING ALGORITHM

For a given point x on the circle and a given family F of arcs by the *overlap set of* x in respect to F (in short: *overlap at* x) we mean the set of arcs from F containing x, and by $r(x/F)$ we denote the number of arcs in this set. For any set X of points on the circle and a family F of arcs denote

$$r(X/F) = \max_{x \in X} r(x/F).$$

By $r_{\max}(F)$ the maximal overlap size in F is denoted. As usual, $\gamma(F)$ stands for the chromatic number of F, that is the minimal number of colors needed to color all arcs of F (in a standard off-line sense). Since each overlap set is a clique (the opposite is not true) it is obvious that $\gamma(F) \geqq r_{\max}(F)$. For any coloring algorithm $\mathcal{A}$ (off-line or on-line) by $\mathcal{A}(F)$ we denote the number of colors used by $\mathcal{A}$ on F.

Algorithm COLOR described below colors on-line a family of arcs $F = \{A_1, \ldots, A_n\}$. By a color of an arc A we mean a pair of integers: a shelf number (nonnegative) and a level number (positive), denoted respectively $sh(A)$ and $lv(A)$. For $i = 0, 1, \ldots$ a shelf Sh_i is defined as a subfamily of F consisting of all those arcs which have been assigned the same shelf number, namely i.

The algorithm tries to spread the incoming arcs evenly among the shelves in such a way that maximal overlap on a shelf is small. More precisely, for a given arc A it looks for a shelf with lowest possible number i such that maximal overlap along A in respect to the sum of the contents of shelves $Sh_0, \ldots, Sh_i$ does not exceed i. Having found such a shelf COLOR assigns to A the lowest possible level, on the first-fit principle.

Algorithm COLOR

Input: A sequence of circular arcs $F = (A_1, \ldots, A_n)$

Output: For each arc A a valid color in the form $(sh(A), lv(A))$

begin
For each i set $Sh_i = \emptyset$;
repeat
  read next arc A; set $i = 0$;
  while $r(A/Sh_0 \cup \ldots \cup Sh_i) > i$ **do**
    set $i = i + 1$;
  set $l = 1$;
  while exists $A' \in Sh_i$ s.t. $lv(A') = l$ and $A \cap A' \neq \emptyset$ **do**
    set $l = l + 1$;
    set $Sh_i = Sh_i \cup \{A\}$, $sh(A) = i$; $lv(A) = l$
until end of input
end

For a sample output of the algorithm see Figure 1.

3. ANALYSIS OF THE ALGORITHM

For the analysis of algorithm COLOR we need notation for specification of overlap sizes in respect to sums of shelves contents at the time when arc

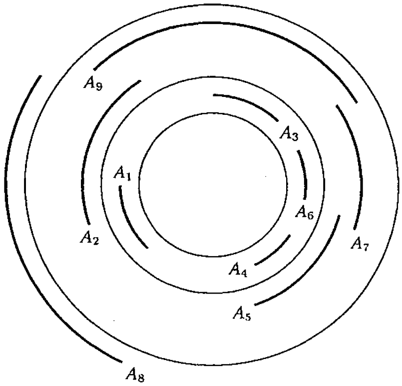

Figure 1. – Sample output of algorithm COLOR

A_j is being colored. By F_j we denote the subfamily $\{A_1, \ldots, A_{j-1}\}$ of all arcs *colored before* A_j and by T_i a subfamily of F consisting of all arcs put onto shelves $Sh_0, \ldots, Sh_i$ *after completion* of the algorithm. Thus, for example, an expression like this:

$$r(A_j/F_j \cap T_i)$$

describes the maximum overlap size along A_j in respect to the sum of the shelves $Sh_0, \ldots, Sh_i$ *just at the moment* when algorithm COLOR starts to color A_j.

LEMMA 1: *If $p < q$ then for each i*

$$r(A_p/F_p \cap T_i) \leq r(A_p/F_q \cap T_i).$$

Proof: Obvious since $F_p \subseteq F_q$. □

LEMMA 2: *If for some $p \neq q$ and $i > 0$, $A_p \cap A_q \neq \emptyset$ and $sh(A_p) = sh(A_q) = i$ then*

$$r(A_p \cap A_q/F_{\max(p,q)} \cap T_{i-1}) \leq i - 1.$$

Observe that $A_p \cap A_q$ can be treated as a family consisting of one or two arcs – the second case happens when A_p and A_q cover the whole circle.

Intuitively, if two overlapping arcs have been put onto the same shelf Sh_i then it is not their intersection part(s) which caused them not to be put onto some lower shelf.

Proof: Let $p < q$ and assume that

$$r(A_p \cap A_q/F_q \cap T_{i-1}) > i - 1.$$

Then, since $A_p \in Sh_i$ and A_q is colored later than A_p, it must be

$$r(A_p \cap A_q / F_q \cap T_i) > i$$

which contradicts $A_q \in Sh_i$. $\square$

Remark: Observe that on the same assumptions as in Lemma 2, if $p < q$ then $F_p \subseteq F_q$ and from Lemma 1 it follows that also

$$r(A_p \cap A_q / F_{\min(p,q)} \cap T_{i-1}) \leq i - 1.$$

LEMMA 3: *For each i, if A_p, $A_q \in Sh_i$, $p \neq q$, then neither $A_p \subseteq A_q$ nor $A_q \subseteq A_p$.*

Proof: For $i = 0$ the thesis is obvious. Let $i > 0$ and assume that for $p \neq q$, $sh(A_p) = sh(A_q) = i$ and $A_p \subseteq A_q$. If $p > q$ then from Lemma 2

$$r(A_p / F_p \cap T_{i-1}) \leq i - 1.$$

If $p < q$ then we apply Lemma 2 and Lemma 1 which yields the same inequality. However, as a result of this inequality, arc A_p must have been put onto Sh_j for some $j < i$ which is a contradiction. $\square$

LEMMA 4: *For each i the largest level used in Sh_i is not greater than* 3. *In Sh_0 only one level is used.*

Proof: The lemma is obvious for Sh_0. Fix $i > 0$ and assume that some arc A_j has been put on level 4 of Sh_i.

From the first-fit property it follows that A_j has nonempty intersection with at least three arcs on Sh_i which are colored 1, 2 and 3. From Lemma 3 none of them can be fully contained in A_j, hence at least two of them must overlap one of the endpoints of A_j – assume it is the clockwise endpoint. Denote this endpoint by β_j and these two arcs by A_k and A_l. Without loss of generality assume that A_k spans a shorter length counterclockwise from β_j than A_l does. This situation is depicted in Figure 2.

Now apply Lemma 2 to A_j and A_l. We obtain:

$$r(A_j \cap A_l / F_{\max(j,l)} \cap T_{i-1}) \leq i - 1.$$

Analogously for A_l and A_k,

$$r(A_l \cap A_k / F_{\max(l,k)} \cap T_{i-1}) \leq i - 1.$$

Hence, as $A_l = (A_l \cap A_k) \cup (A_j \cap A_l)$ and $l \leq \max(j, l)$ and $l \leq \max(l, k)$, we get

$$r(A_l / F_l \cap T_{i-1}) \leq i - 1$$

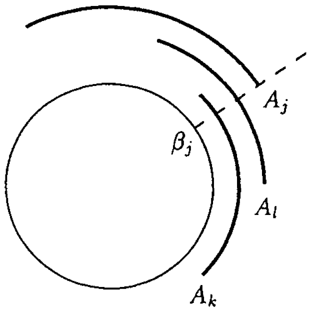

Figure 2. – Three overlapping arcs on a single shelf.

in which case algorithm COLOR puts A_l on a shelf with a smaller number than i which is a contradiction and finishes the proof. □

LEMMA 5: *The number of shelves used by algorithm COLOR on a family F of circular arcs does not exceed $r_{\max}(F)$.*

Proof: Consider any arc A_j which has been put onto the topmost shelf Sh_i. From the algorithm properties it follows that

$$r(A_j/F_j \cap T_{i-1}) > i - 1.$$

Hence for some $x \in A_j$ it must be

$$r(x/F_{j+1}) > i.$$

Therefore, as the shelves are numbered from 0, we have $r_{\max}(F) \geq i + 1$. □

From the last two lemmata the main result follows:

THEOREM 6: *The number of colors used by algorithm COLOR on any family F of circular arcs does not exceed $3\, r_{\max}(F) - 2$.* □

This way we have proved that any circular arc graph G can be colored on-line with no more than $3\gamma(G) - 2$ colors, or in other words, that there exists a 3-competitive on-line coloring strategy for such gaphs.

REFERENCES

1. M. CHROBAK and M. ŚLUSAREK, On some packing problem related to dynamic storage allocation, *RAIRO Theoretical Informatics and Applications*, 1988, *22*, pp. 487-499.
2. M. R. GAREY, D. S. JOHNSON, G. L. MILLER and C. H. PAPADIMITRIOU, The complexity of coloring circular arcs and chords, *SIAM J. Alg. Disc. Meth.*, 1980, *1*, pp. 216-227.
3. H. A. KIERSTEAD, A polynomial time approximation algorithm for Dynamic Storage Allocation, *Discrete Mathematics*, 1991, *88*, pp. 231-237.

4. H. A. Kierstead and W. T. Trotter, An extremal problem in recursive combinatorics, *Congressus Numerantium*, 1981, *33*, pp. 143-153.
5. M. V. Marathe, H. B. Hunt III and S. S. Ravi, Efficient approximation algorithms for Domatic Partition and On-line coloring of Circular Arc Graphs, to appear. Extended abstract in: *Proc. Intl. Conf. on Computing and Information ICI'93, Sudbury*, May 1993, pp. 26-30.
6. M. Ślusarek, *A coloring algorithm for interval graphs*, Mathematical Foundations of Computer Science '89, LNCS 379, pp. 471-480.