AN APPROXIMATION ALGORITHM FOR COLORING CIRCULAR-ARC GRAPHS

Wei-Kuan Shih[1] and Wen-Lian Hsu[2]

Key Words: graph, clique, coloring, matching, algorithm

ABSTRACT

Consider families of arcs on a circle. The minimum coloring problem on arc families has been shown to be NP-hard by Garey, Johnson, Miller and Papadimitriou. It is easy to show that 2q colors are sufficient for any arc family F, where q is the size of a maximum clique in F and 3q/2 colors are necessary for some families. It has long been open problem to find a coloring algorithm which uses no more than $\alpha \cdot q$ colors , where α is strictly less than 2. In this paper we present such an algorithm with α=5/3. Our algorithm is based on: (1) an extension of an earlier result of Tucker on coloring special families and (2) a characterization of the existence of perfect matching in bipartite graphs.

1 Department of Computer Science, National Tsing-Hua University, Republic of China.

2 Institute of Information Sciences, Academia Sinica, Republic of China.

An Approximation Algorithms for Coloring Circular-Arc Graphs

1. Introduction

A circular arc family F is a collection of arcs on a circle. A graph G is a ***Circular - arc graph*** if there is a circular arc family F and a one-to-one mapping of the vertices in G and the arcs in F such that two vertices in G are adjacent if and only if their corresponding arcs in F overlap. Circular-arc graphs have applications in compiler design and traffic light sequencing [4]. Various characterization and optimization problems on circular-arc graphs have been studied [4]. A ***clique*** in F is a set of pairwise overlapping arcs. The size of a maximum clique in F is denoted by $\omega(F)$. A ***coloring*** of F is an assignment of colors to its arcs such that no two overlapping arcs have the same color. The minimum number of colors needed in coloring F is denoted by $\gamma(F)$. Obviously, we have $\omega(F) \leq \gamma(F)$. The maximum clique problem for circular-arc graphs can be solved efficiently [2,6]. However, it has been shown by Garey, Johnson, Miller and Papadimitriou [2] that the minimum coloring problem for circular-arc families is NP-hard. Based on the property of interval graphs, it is easy to color circular-arc families using no more than 2q colors, where $q = \omega(F)$. It has long been an open problem to find an approximation algorithm which uses no more than $\alpha \cdot q$ colors, where α is strictly less than 2. In this paper, we show that it is sufficient to use at most 5q/3 colors.

Without loss of generality, assume all arc endpoints are distinct. Each arc has two endpoints. Define the ***clockwise*** (resp. ***counterclockwise***) ***endpoint*** of arc i to be the first endpoint of i encountered in a clockwise (resp. counterclockwise) traversal from any interior point of i. Denote the clockwise endpoint of i by b_i, the counterclockwise endpoint by a_i and the arc i by (a_i, b_i). Define the continuous part of the circle from a point c along the clockwise direction to another point d as the ***segment*** (c, d) of the circle. We shall reserve the term "arc" for members of F. An arc u=(a, b) is said to ***cross a point*** p if $p \in$ segment (a, b). For each arc i, define its ***overlapping set*** S_i to be the set of arcs in F (including i itself) crossing a_i. Let s be

$\max_{i \in F} | S_i|$, which is called the ***number of layers*** in F.

In section 2, we describe a special property of an arc family which enables us to use 3q/2 colors. For an arc family which does not satisfy this property, we describe in section 3 how to remove some arcs and force the remaining subfamily to satisfy. Those arcs removed are colored in such a way that the total number of colors used is no more than 5q/3.

2. Using 3q/2 Colors for Special Arc Families

Intuitively speaking, arcs which overlap with many other arcs need to be colored more prudently in order to save the total number of colors used. To make this notion more precise, we define the following terms. Three arcs covering a circle are said to form a ***singular triple***. Two arcs covering a circle are said to form a ***singular twin***. An earlier result on approximate coloring is the following.

Lemma 2.1 (Tucker [7]). An arc family containing no singular triples needs at most 3s/2 colors.

That 3s/2 is also necessary can be verified by considering a family of five arcs whose corresponding circular-arc graph is a five cycle without chords (edges connecting non-consecutive vertices). Tucker [7] conjectured that any arc family F needs at most 3q/2 colors. The special property we considered for an arc family F is the following:

(2.2) There exists a clique Q in F satisfying that each singular triple of F contains exactly one arc in Q.

For a clique Q containing neither singular triples nor singular twins, there exists a point p on the circle such that no arcs of Q crosses p. The arc u of Q such that a_u is first encountered in a clockwise traversal from p is called the ***left-most*** arc of Q. Two arcs which are non-overlapping are said to be ***independent***. For any arc i = ($\boldsymbol{a_i}$, $\boldsymbol{b_i}$), define NEXT(i) to be the arc j such that a_j belongs to segment ($\boldsymbol{b_i}$, $\boldsymbol{a_i}$) and is the first such endpoint encountered in a clockwise traversal from $\boldsymbol{b_i}$. Note that arc i and NEXT(i) are not necessarily independent. The basis of our algorithm is the following theorem, which is an extension of Lemma 2.1.

Theorem 2.3. If F is an arc family satisfying (2.2), then it needs at most 3s/2 colors.

Proof. If F is a clique, then (2.2) implies that there is no singular triples in F and the theorem follows from Lemma 2.1. Hence, assume F is not a clique. Let Q be a clique in F satisfying (2.2). Clearly, Q contains neither singular triples nor singular twins within itself (a singular twin in Q coupled with any arc in F\Q gives rise to a singular triple in F containing two arcs in Q). The same holds for F\Q if $|F| \geq 3$. Hence, it is meaningful to define the left-most arc of Q.

The arguments we used are similar to those of Tucker's [7]. Since F and Q will be updated at every iteration, we use the same notations to denote the "current" F and Q satisfying property (2.2). We shall repeatedly remove two layers of arcs in F and use three new colors on the removed arcs until F becomes empty. The algorithm is divided into two parts. We first execute the procedure described in Case 1 until Q becomes empty and then go on to Case 2.

Case 1. Q is not empty: let $u_1=(a_{u1}, b_{u1})$ be the left-most arc in Q. Starting from u_1, we find a sequence of arcs $u_1, u_2, \ldots, u_k$ such that $u_i = NEXT(u_{i-1})$ for $i \geq 2$ and u_k is the first arc with $a_{u1} \in$ segment (b_{uk-1}, b_{uk}). Note that a_{uk} is either in segment (a_{u1}, b_{uk}) or in segment (b_{uk-1}, a_{u1}). Let $L_1=\{u_1, \ldots, u_{k-1}\}$.

If $a_{uk} \in$ segment (a_{u1}, b_{uk}), then assign the same color to all arcs in L_1 and remove them from F. Since each arc u not in L_1 has a_u crossed at least once by some arc in L_1 in the clockwise traversal, s will be reduced by 1.

Now, assume $a_{uk} \in$ segment (b_{uk-1}, a_{u1}). Starting from u_k, we find a sequence of arcs $u_k, \ldots, u_r, \ldots, u_t$ such that $u_i = NEXT(u_{i-1})$ for $i \geq k+1$, u_r is the first arc with $a_{uk} \in$ segment

(b_{ur-1}, b_{ur}) and u_t is the first arc with $a_{u1}\in$segment (b_{ut-1}, b_{ut}) (it is possible that $u_r = u_t$). Let $L_2 = \{u_k, \ldots, u_{r-1}\}$ and $L_3 = \{u_r, \ldots, u_t\}$. If $a_{ut}\in$segment (a_{u1}, b_{ut}), then each arc u not in the sequence $u_1, \ldots, u_{t-1}$ has a_u crossed at least twice in the clockwise traversal. Thus, removing arcs in the above sequence reduces s by 2. Now assign one new color to all arcs in L_1, another to all arcs in L_2 and a third color to all arcs in $L_3\backslash\{u_t\}$. Remove all colored arcs from F.

Assume, now, $a_{ut}\in$segment (b_{ut-1}, a_{u1}). In this case, we need to remove all arcs in $L_1\cup L_2\cup L_3$ to reduce s by 2. Now assign one new color to all arcs in L_1 and another color to arcs in L_2. All we are left to show is the following.

Claim. Arcs in L_3 are independent.

Proof. Suppose not. Then $u_r \neq u_t$. Since arcs in $u_r, \ldots, u_{t-1}$ are independent, u_r must overlap with u_t. Because both u_r and u_t overlap with u_k, we have a singular triple u_k, u_r and u_t as shown in Fig.1. By assumption, both u_k and u_t cross the endpoint a_u of the left-most arc u of Q. Hence, neither of them belongs to Q. By this implies that u_r belongs to Q by property (2.2). Hence, u_r crosses b_{u1}. But then, u_1, u_k and u_r form a singular triple with two arcs in Q, a contradiction. ■

Case 2. Q is empty. Then F contains neither singular triples nor singular twins. As long as $s\geq 2$, we can choose an arbitrary arc as u_1, construct a sequence of arcs $u_1, \ldots, u_t$ and repeat the coloring procedure as before. Each time, we assign three new colors to the removed arcs and reduce s by 2. If s is reduced to 1 at the final iteration, then all arcs in the current F are independent and can share one color.

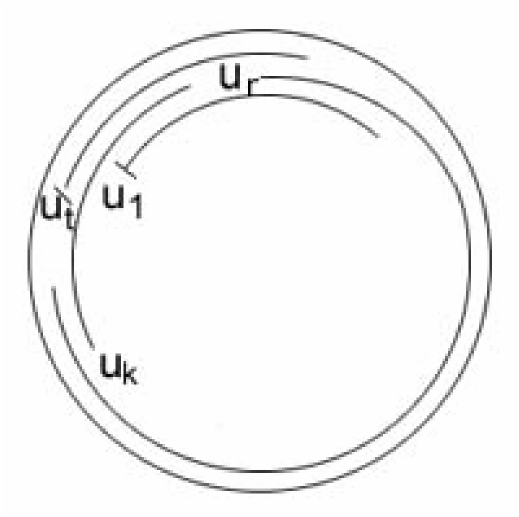

Figure 1. The singular triple u_k, u_r and u_t

In each iteration of the above two cases, we either reduce s by 1 and use one new color on the removed arcs or reduce s by at least 2 and use 3 new colors. The number of iterations is at most $\lceil s/2 \rceil$. Hence, the total number of colors used is at most $3 \cdot (s/2) = 3s/2$. ■

3. Using 5q/3 Colors for General Arc Families

In this section, we describe how to remove arcs from a family F to yield a subfamily satisfying (2.2). Those arcs removed will be assigned colors in such a way that the total number of colors used is at most 5q/3. Define a clique Q to be ***simple*** if every arc in Q crosses a common point on the circle. The coloring method described in Theorem 2.3 is useful in reducing the number of layers in F, which in turn reduces the size of largest simple cliques in F. However, reducing s does not necessarily reduce the maximum clique size q, especially when there are maximum cliques in F containing singular triples or singular twins. Fig.2 gives an example in which q remains unchanged until s is reduced to 2/3 of its original size. Note that larger examples can be obtained by duplicating arcs.

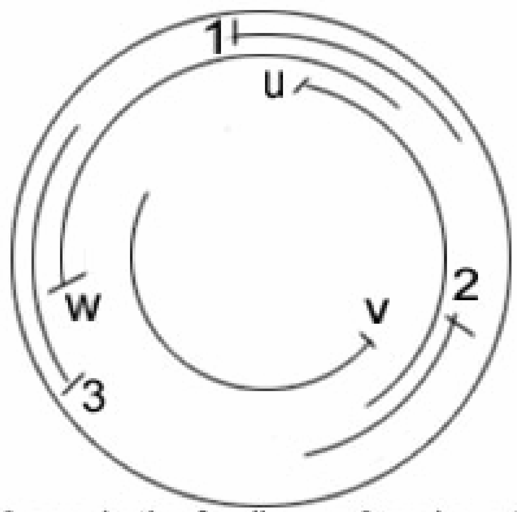

Figure 2. The relationship between s and q

Instead of trying to reduce the size of maximum cliques in the current subfamily (which can be very difficult), we pick a specific simple clique Q (initially maximum in the original family) and manage to reduce its size q and the number of layers s simultaneously by removing certain arcs. The advantage of using simple clique is that F\Q becomes a family of intervals, which can be colored optimally. To do this, we require the family F to satisfy the following property:

(3.1) There exists a maximum simple clique Q in F such that F\Q contains no singular triples.

The next lemma states that, as far as the approximation is concerned, one could assume that the given family F satisfies (3.1).

Lemma 3.2. Let F be any arc family. Then there exists a superfamily $F' \supseteq F$ such that $\gamma(F') = \gamma(F)$, $\omega(F') = \omega(F)$ and the number s' of layers in F' equals $\omega(F')$ (namely, there exists a maximum simple clique in F')

Proof. Let s be the number of layers of F. Let i be an arc in F with $|S_i| = s$. If $s = \omega(F)$, then we can let F' be F. If $s < \omega(F)$, then add $\omega(F) - s$ tiny arcs to F each of which crosses exactly one endpoint, a_i. In this new family F', $S_i \cup$ {new arcs} is a maximum simple clique. Hence, $\omega(F') = \omega(F)$. Furthermore, if F is colored using $\gamma(F)$ colors, then there obviously exist $\omega(F) - s$ colors for the new arcs in F'. Hence, $\gamma(F') \leq \gamma(F)$. Since $F' \supseteq F$, $\gamma(F) \leq \gamma(F')$. Therefore, $\gamma(F') = \gamma(F)$. ■

From now on, we assume that the given family F* contains a maximum simple clique Q* satisfying (3.1). At each iteration of the reduction, we use F and Q to denote the current updated subfamily and the remaining arcs in the clique. We use q* (resp. s*) to denote the initial (resp. layers of F*) size of Q* and q (resp. s) to denote its current size. Since Q* is simple, s* = q*. Note that, at each iteration, property (3.1) is maintained for the current F and Q regardless how we remove arcs. However, for the algorithm to work, the current size of Q does not have to be maximum in the current subfamily F.

We will make use of an important property of bipartite matching, which is stated as follows.

Lemma 3.3. Let Q be a maximum clique of F. Let D be any other clique in F that is disjoint from Q. Then we can match, for each arc v in D, a distinct arc u in Q which does not overlap with v.

Proof. Construct a bipartite graph $G=(X \cup Y,E)$, where there is a one-to-one mapping of vertices in X (resp. Y) and arcs in D (resp. Q) such that two vertices are adjacent in G if their corresponding arcs in F are not overlapping. Because Y is a maximum independent set in G, there is a perfect matching for X in G.

The main idea of our algorithm is to remove singular triples and singular twins from F until either

(3.3) F and Q satisfy (2.2) and we can use the algorithm in Theorem 2.3.

or (3.4) the size of Q is small enough ($\leq q^*/3$) so that we can assign each arc of Q a new color and color the remaining interval family F\Q optimally.

At each iteration, the arcs removed will satisfy the following condition:

(3.5) s is reduced by at least 1, q is reduced by at least 2 and at most 2 new colors are assigned to the removed arcs.

Property (2.2) can be violated by one of the following three cases in F: (1) the clique Q contains a singular twin; (2) there exists a singular twin with one arc in Q another in F\Q; (3) there exists a singular triple in F with exactly two arcs in Q. We shall eliminate these situations one by one below.

Case 1. Q* contains singular twins.

Exhaustively, we find all singular twins in Q* and list them as follows: (u_1,v_1), (u_2,v_2),…, (u_k,v_k). We first analyze the simple clique Q' obtained by removing these arcs from the simple clique Q. Because F\Q contains no singular triples, Q' contains no singular twins or singular triples. Hence, there exists a point d on the circle which no arc of Q' crosses. Define D* to be the set of arcs in F*\Q* crossing d. By Lemma 3.3, each arc v in D* can be matched to a distinct arc M(v) in Q*, which does not overlap v (conversely, the arc in D* which is matched to an arc u in Q* is denoted by M(u); if an arc u in Q* is not matched with any arc in D*, define M(u) to be ϕ).

Now, remove these singular twins one at a time. Each time a singular twin (u_i,v_i) is removed we assign one new color to u_i, $M(u_i)$ and another color to v_i, $M(v_i)$. It is easy to check that condition (3.5) is satisfied at each iteration. At the end of the Case 1 procedure, Q contains no singular twins and we go on to Case 2.

Case 2. There exists a singular twin in F containing one arc in Q and another in F\Q.

Again, we find all such singular twins and list them as follows: (u_1, v_1), (u_2, v_2),…, (u_r, v_r) with u_i in Q and v_i in F\Q. Now, remove these singular twins one by one. Each time a singular twin (u_i,v_i) is removed we assign one new color to u_i, M(u_i) and another color to v_i, M(v_i). It is easy to check that condition (3.5) is satisfied at each iteration. At the end of the Case 2 procedure, F contains no singular twins and we go on to Case 3.

Case 3. There exist a singular triple in F with two arcs in Q and one arc in F\Q (which must belong to D).

Because, Q contains no singular twins or singular triples, it is meaningful to define the left-most arc, say u_1, of Q. It is easy to verify that if there exists a singular triple in F having two arcs in Q, then we can always choose one of these arcs to be u_1. Let u_1, u and v be such a singular triple, where $u \in Q$, $v \in D$.

Starting from u_1, we find a sequence of arcs u_1, u_2,…, u_k such that u_i = NEXT(u_{i-1}) for $i \geq 2$ and u_k is the first arc with $a_v \in (b_{uk-1}, b_{uk})$. Note that u_1 is the only arc in Q in the above sequence. Consider two subcases:

Subcase 1. u_1 does not overlap with u_k as shown in Fig. 3. Assign one new color to all arcs in {u_1,…,u_k} and another to arcs v, M(v). Remove all colored arcs.

Subcase 2. u_1 overlaps with u_k as shown in Fig.3. Then u_k must belong to D. Assign one new color to all arcs in {u_1,…,u_{k-1}} and another to arcs u_k, M(u_k). Remove all colored arcs. Note that v is not removed unless v = u_k.

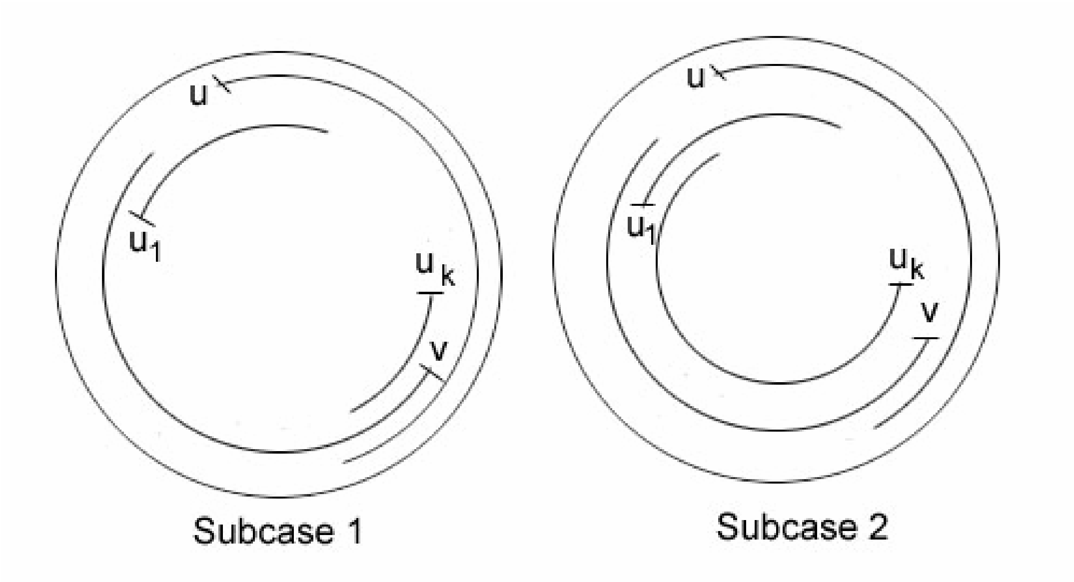

Figure 3. The two subcases in Case 3

Note that the M(v) in Subcase 1 and the M(u_k) in Subcase 2 must remain uncolored before this iteration for the following reasons (it suffices to argue for M(v)): (a) M(v) cannot possibly be colored in Case 1 or Case 2 (otherwise, v must have also been colored and removed); (b) M(v) cannot be colored in Case 3 previously because its counterclockwise endpoint has never been the left-most in Q and its matched arc v in D has not been colored. It is easy to verify that (3.5) is satisfied in each of the two subcases.

Our algorithm will execute the procedures in the above three cases either until F and Q satisfy (2.2) or until the number of iterations reaches [q*/3], whichever happens first.

Theorem 3.6. Let F* be an arc family containing a maximum simple clique Q* satisfying (3.1). Then the algorithm uses at most 5q*/3 colors.

Proof. Suppose we have executed all procedures in the three cases and the number t of iterations taken is still no greater than [q*/3]. At this point, we have used 2t colors, the number of layers of F is reduced by at least t and the size of Q* is reduced by at least 2t. Since F and Q satisfy (2.2), we can apply the algorithm in Theorem 2.3 to color the remaining arcs using at most (3/2) (q*-t) colors. Hence, the total number of colors used is bounded by

$$2t + (3/2)\cdot(q^*-t) = t/2 + 3q^*/2 \leq q^*/6 + 3q^*/2 = 5q^*/3.$$

Suppose we have reached the [q*/3]-th iteration in executing the procedures in the above three cases. At this point, we have used 2[q*/3] colors, the number of layers of F is reduced by at least [q*/3] and the size of Q* is reduced by at least 2[q*/3]. Now, apply (3.3) to assign a different color to each arc in Q and to color arcs in F\Q based on any coloring algorithm on interval graphs using at most q* - [q*/3] (an upper bound on the current s) colors. Hence, the total number of colors used is bounded by

$$2[q^*/3] + (q^* - 2[q^*/3]) + (q^* - [q^*/3]) = 2q^* - [q^*/3] \leq 5q^*/3$$

There are two major steps involved in our algorithm. One is to find a maximum clique in F, which can be done in $O(n^2 \log\log n)$ time [1]. Another is to find a bipartite matching, which takes at most $O(n^{2.5})$ time [5]. Hence, the entire approximation takes at most $O(n^{2.5})$ time.

REFERENCES

1. A. Apostolico and S.E. Hambrus, Finding maximum cliques on circular-arc graphs, Inform. Process. Lett., 26(1987), pp 209-215.

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. Dis. Meth., 1(1980), pp. 185-200.

3. F. Gavril, Algorithms on circular-arc graphs, Networks, 4(1974), pp.357-369.

4. M.C. Golumbic, Algorithmic graph theory and perfect graphs, Academic Press, New York, 1980.

5. J.E. Hopcroft and R.M. Karp, A$n^{2.5}$ algorithm for maximum matching in bipartite graphs, SIAM J. Comput., 2(1973), pp. 225-231.

6. W.-L. Hsu, Maximum weight clique algorithms for circular-arc graphs and circle graphs, SIAM J. Comput., 14(1985), pp. 224-231.

7. A. Tucker, Coloring a family of circular-arc graphs, SIAM J.Appl. Math.,29(1975), pp 493-502.