Local 7-Coloring for Planar Subgraphs of Unit Disk Graphs

J. Czyzowicz[1], S. Dobrev[2], H. González-Aguilar[3], R. Kralovic[4], E. Kranakis[5], J. Opatrny[6], L. Stacho[7], and J. Urrutia[8]

[1] Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. Research supported in part by NSERC.
[2] School of Information Technology and Engineering (SITE), University of Ottawa, 800 King Eduard, Ottawa, Ontario, Canada, K1N 6N5.
[3] Centro de Investigacion en Matematicas, Guanajuato, Gto., C.P. 36000, Mexico.
[4] Department of Computer Science, Faculty of Mathematics, Physics and Informatics Comenius University, Bratislava, Slovakia
[5] School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6. Research supported in part by NSERC and MITACS.
[6] Department of Computer Science, Concordia University, 1455 de Maisonneuve Blvd West, Montréal, Qubec, Canada, H3G 1M8. Research supported in part by NSERC.
[7] Department of Mathematics, Simon Fraser University, 8888 University Drive, Burnaby, British Columbia, Canada, V5A 1S6. Research supported in part by NSERC.
[8] Instituto de Matemáticas, Universidad Nacional Autónoma de México, Área de la investigación cientifica, Circuito Exterior, Ciudad Universitaria, Coyoacán 04510, México, D.F. México.

Abstract. The problem of computing locally a coloring of an arbitrary planar subgraph of a unit disk graph is studied. Each vertex knows its coordinates in the plane, can directly communicate with all its neighbors within unit distance. Using this setting, first a simple algorithm is given whereby each vertex can compute its color in a 9-coloring of the planar graph using only information on the subgraph located within at most 9 hops away from it in the original unit disk graph. A more complicated algorithm is then presented whereby each vertex can compute its color in a 7-coloring of the planar graph using only information on the subgraph located within a constant number of hops away from it.

1 Introduction

We are interested in the graph vertex coloring as applicable to wireless ad-hoc networks. The wireless ad-hoc networks of interest to us are geometrically embedded in the plane and consist of a number of *location aware* nodes, say n, whereby two nodes are adjacent if and only if they are within the transmission range of each other. If all the nodes have the same transmission range then these networks are known as *unit disk graphs*. For such graphs there have been several papers in the literature addressing the coloring problem. Among these it is worth mentioning the work by Marathe et al. [10] (presenting an on-line coloring heuristic which achieves a competitive ratio of 6 for unit disk graphs: the heuristic does not need a geometric representation of unit disk graphs which is used only in establishing the performance guarantees of the heuristics), Graf et al. [7] (which improves on a result of Clark, Colbourn and Johnson (1990) and shows that the coloring problem for unit disk graphs remains NP-complete for any fixed number of colors $k \geq 3$), Caragiannis et al. [2] (which proves an improved upper bound on the competitiveness of the on-line coloring algorithm First-Fit in disk graphs which are graphs representing overlaps of disks on the plane) and Miyamoto et al. [11] (which constructs multi-colorings of unit disk graphs represented on triangular lattice points).

There are also several papers on coloring restricted to planar graphs of which we note Ghosh et al. [6] because it is concerned with a self-stabilizing algorithm for coloring such graphs. Their algorithm achieves a 6 coloring by transforming the planar graph into a DAG of out-degree at most five. However, this algorithm needs the full knowledge of the topology of the graph. The specificity of the problem for ad-hoc networks requires a different approach. An ad-hoc network can be a very large dynamic system, and in some cases a node can join or leave a network at any time. Thus, the full knowledge of the topology of an ad-hoc network might not be available, or possible for each node of the network at all times. Thus algorithms that can make computations in a fully distributed manner, using in each node only information about the network within a fixed distance neighborhood of the node, are of particular interest in ad hoc networks. Examples of algorithms of this type are the Gabriel test [5] for constructing a planar spanner, face routing [8],[1] or an approximation of the minimum spanning tree [14], [3].

To reduce network complexity, the unit disk graph G is sometimes reduced to a much smaller subset P of its edges called a spanner. A good spanner must have some properties so that certain parameters of communication within P are preserved. To ensure all to all communication P must be connected. An important property is having a constant stretch factor s, guaranteeing that the length of a path joining two nodes in G is at most s times shorter that the shortest path joining these nodes in P. A desired property of P is planarity, which, on one hand, permits an efficient routing scheme based on *face routing* and, on the other hand, ensures linear complexity of P with respect to its number of nodes. Planar graphs also have low chromatic number, hence a small set of frequencies is sufficient to realize radio communication.

In this paper, we are interested in *local* distributed coloring algorithms whereby messages emanating from any node can propagate for only a constant number of hops. This model was first introduced in the seminal paper of Linial [9]. One of the advantages of this model is that it aims to obtain algorithms that could cope with a dynamically changing infrastructure in a network. In this approach, each node may communicate with nodes at a bounded distance from it and thus a local change in the network only needs a local adjustment of a solution. In Linial's model of communication, locality results in a constant-time distributed algorithm.

1.1 Network model and results of the paper

We are given a set S of n points in the plane and a planar subgraph G of the unit disk graph induced by S. We assume that G is connected and all the nodes either know their exact (x, y) coordinates (which could be achieved for example by having the nodes equipped with GPS devices) or have a consistent relative coordinate system. If G is not connected, all reasoning can be applied to each connected component of G independently, in which case the coloring constructed applies to each connected component. We propose two local coloring algorithms. The first simple algorithm computes a 9-coloring of the planar graph. We assume that each node knows its 9-neighborhood in the original unit disk graph (i.e. all nodes at distance at least 9 hops away from it), it can communicate with each node of its neighborhood and it is aware which of these nodes belong to the planar subgraph. We then present a more complicated algorithm whereby each vertex can compute its color in a 7-coloring of the planer graph using only information on the subgraph located within at most a constant number $h = 201$ of hops away from it. The constant h is quite large though in practice nodes at much smaller distance will need to communicate. The algorithm does not determine locally either what the different connected components are or even what are the local parts of a component connected somewhere far away. Moreover, the correctness of the algorithm is independent of the connectivity of the planar subgraph.

2 Simple Local Coloring Algorithm

The basic idea of the coloring algorithms in this paper is to partition the plane containing G into fixed sized areas, compute a coloring of the subgraph of G within each such area independently and possibly adjust colors of some vertices that are on the border of an area and thus are adjacent to nodes in another area. This is possible to do consistently and without any pre-processing because the nodes know their coordinates and thus can determine the area in which they belong. Since each area is of fixed size, a subgraph of the given unit disk graph belonging to this area is of a bounded diameter. Hence a constant number of hops is needed for a node to communicate with each other node of the same area.

2.1 Coloring with regular hexagonal tilings

The simplest partitioning we consider is obtained by tiling the plane with regular hexagons having sides of size 1. We suppose that two edges of the hexagons are horizontal and one of the hexagons is centered in coordinates $[0, 0]$. To assure the disjointness of the hexagon areas we assume that only the upper part of the boundary and the leftmost vertex belongs to each hexagon area while the rightmost vertex does not belong to it. Under such conditions, two vertices of G can be connected by an edge only if they are in the same or adjacent hexagon areas.

As each vertex knows its own coordinates, it can calculate which hexagon it belongs to. In the first step each vertex communicates with vertices within its hexagon and learns the part of the subgraph located in its hexagon. By Lemma 1, communication inside a hexagon may be done using at most nine hops.

Lemma 1. *Any connected component of the subgraph of a unit disk graph induced by its nodes belonging to a regular hexagon with sides of size* 1 *has a diameter smaller or equal to nine. Moreover, there exist configurations of nodes inside the hexagon of diameter equal to nine.*

We could apply the standard 4-coloring algorithm for each connected component of the graph induced by G on points within each hexagon. Since the tiling of the plane by hexagons can be 3-colored, three disjoint sets of four colors can be used, one set of colors for vertices in each hexagon area of the same color. This would lead to a 12 coloring of G (see the coloring scheme of the hexagonal

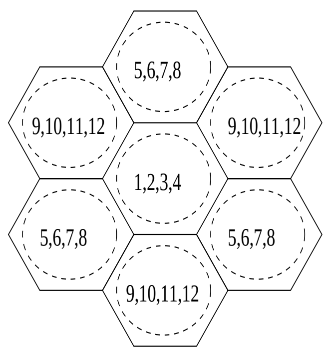

Fig. 1. Coloring of a hexagonal tiling of the plane using 12 colors.

tiling depicted in Figure 1).

The number of colors can be reduced by coloring the outer face of each component in a hexagon with prescribed colors, using the result of the following theorem.

Theorem 1 ($3+2$ Coloring, Thomassen [12]). *Given a planar graph G,* 3 *prescribed colors and an outer face F of G, graph G can be* 5*-colored while the vertices of F use only the prescribed three colors.*

The precise result in Thomassen's paper [12] (see also Dörre [4]) states that every planar graph is L-list colorable for every list assignment with lists of size 5. The proof of this result however implies the following statement taken from Tuza et al. [13] that for a planar graph G with outer face F, every pre-coloring of two adjacent vertices v_1, v_2 of F can be extended to a list coloring of G for every list assignment with $|L(v)| = 3$ if $v \in V(F) - \{v_1, v_2\}$ and $|L(v)| = 5$ if $v \notin V(F)$, where $V(F)$ denotes the vertex set of F.

Using Theorem 1 the number of colors can be reduced to 9 as follows. The idea is that the vertices of G in a hexagon can be adjacent only to the outer face vertices of the graphs induced in the neighboring hexagons. Three disjoint sets of colors of size three are used as the prescribed colors of the vertices on the outer faces, the inner vertices of G in a hexagon can employ, in addition to the three colors used on the outer face of its hexagon, two additional colors of the outer faces of other hexagons (see Figure 2).

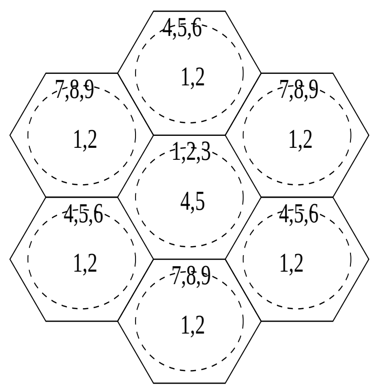

Fig. 2. Colorings of a hexagonal tiling of the plane using 9-colors.

Theorem 2. *Using the partitioning of a plane into regular hexagons with sides of size 1, a vertex can compute its color in a* 9*-coloring of the planar graph using only information on the subgraph located within at most* 9 *hops away from it.*

3 The 7-coloring Algorithm

In this section we give a 7 coloring algorithm and prove its correctness. The trade-off is the larger area of the network a node needs to examine in this algorithm and a more complex partitioning of the plane.

3.1 Reducing the number of colors using mixed tilings

We will employ the tiling using octagons and squares shown in Figure 3. Each square is of size $5 + \epsilon$ while the slanted part of an octagon border is of length $3 + \epsilon$, meaning that these sides can be chosen arbitrarily close to but greater than 5 and 3, respectively. We shall assume that one of the octagons is centered in coordinates $[0, 0]$. The reasons for choosing the sizes of tiles this way is to isolate the meeting places of octagons (the slanted border part) from each other so that each of them can be dealt with independently and locally, and to ensure

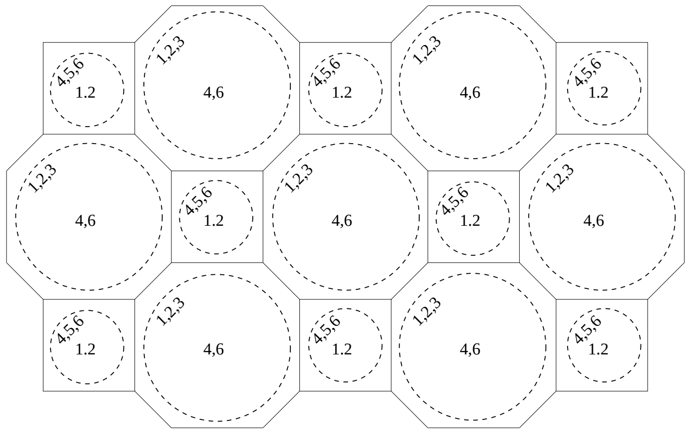

Fig. 3. A coloring of the set of points based on the octagon/square tiling of the plane.

that there is no edge between two vertices in different squares or between two vertices of the same squares that could be recolored for different crossings in the algorithm. In handling a meeting place, only vertices at a distance at most 2 from it will be recolored, therefore it is impossible to have two neighboring vertices recolored due to different crossing.

Similarly as in the previous coloring, the subgraph induced by the vertices in a tile is colored using 5 colors, three of them are used on the outer face and these three colors plus the additional two colors are used on vertices not on the outer face using Theorem 1. Since each node knows its location, it can determine which tile it belongs to. By Lemma 2 the communication inside an octagon may be done using at most 201 hops.

Lemma 2. *Any subgraph of a unit disk graph induced by its vertices belonging to an octagon used in the algorithm has a diameter smaller or equal to* 201*.*

Despite the fact that the simple, surface comparing argument leaves some room for improvement (the packing density is at most $\pi\sqrt{3}/6 = 0.907$), it is possible to construct configurations of nodes, centered inside the octagon, inducing a graph of diameter at least 183.

Since the square tile admits a smaller hop diameter, any node can determine the subgraph induced by the vertices in its tile by examining nodes at hop distance at most 201.

The color sets used in the tiles are as specified in Figure 3. The resulting coloring is using only 6 colors. Due to the chosen sizes of the tiles and the chosen coloring scheme, an edge of G crossing from a square to an octagon is between vertices of different colors. After this initial color assignment, any edge of G whose endpoints are of the same color is necessarily an edge crossing the slanted part of the border of two adjacent octagons. The following construction shows that using one additional color and with careful attention to detail near the common border of adjacent octagons, some of the vertices can be recolored in order to achieve a 7-coloring of G. Details follow in Subsection 3.2.

3.2 Adjusting the coloring

As seen from Figure 3, the centers of the tiles form an infinite regular mesh. We shall denote the hexagon tile that is centered at coordinates $[0, 0]$ as $S_{0,0}$. For a tile denoted $S_{i,j}$, its horizontal left, horizontal right, vertical down, vertical up neighboring tile is denoted $S_{i-1,j}$, $S_{i+1,j}$, $S_{i,j+1}$, $S_{i,j-1}$, respectively. Let $G_{i,j}$ denote the subgraph of G induced by the vertices located within $S_{i,j}$ and suppose that $F_{i,j}$ denotes the outer face of $G_{i,j}$. In case that $G_{i,j}$ is not connected, we consider each connected component of $G_{i,j}$ separately. (Notice that we only need to consider those components of $G_{i,j}$ that contain vertices adjacent to more than one octagon, for otherwise the coloring of such a component could be added to the coloring of $G_{i+1,j+1}$.)

Let $S_{i,j}$ be one of the hexagonal tiles, i.e., $i + j$ is even. Consider the place where $S_{i,j}$ and $S_{i+1,j+1}$ meet (we call it a $CR_{i,j}$ crossing). Consider the sequence of vertices obtained in a counterclockwise cyclical traversal of the outer face $F_{i,j}$ of $G_{i,j}$ and let $L = \{u_1, u_2, \ldots, u_k\}$ be the *shortest subsegment* of this traversal containing all the vertices connected to $G_{i+1,j+1}$, i.e., as in Figure 4.

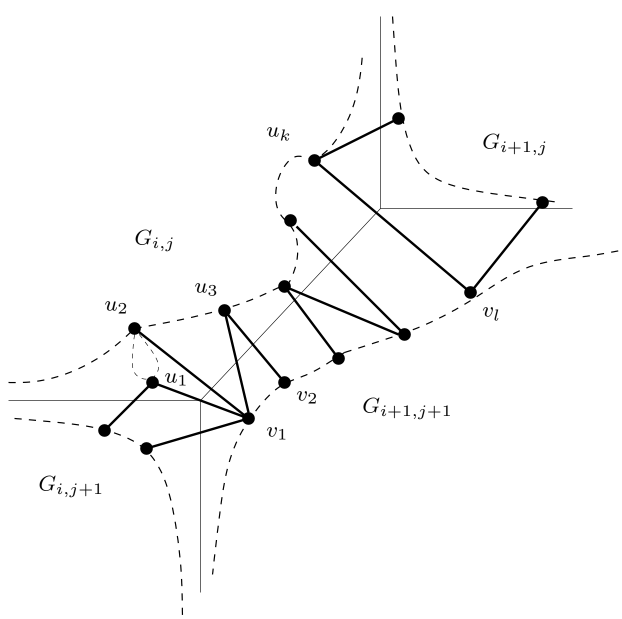

Fig. 4. A typical simple crossing.

Notice that L could contain some vertex more than once if the outer face is not a simple curve. Define $M = \{v_1, v_2, \ldots, v_l\}$ analogously for $G_{i+1,j+1}$, using clockwise traversal.

We say that crossing $CR_{i,j}$ is *simple* if no inside vertex of L different from u_1 and u_k is connected both to a vertex of M and to a vertex in $G_{i,j+1}$ or $G_{i+1,j}$, and the same analogous condition holds for the inside vertices of M. If the crossing is simple, the problem of having some edges between vertices of L or M having both endpoints of the same color can be resolved using the following lemma.

Lemma 3. *Let $CR_{i,j}$ be a simple crossing (see Figure 4). Then the vertices of L and M and some of the neighbors of u_1, v_1, u_k and v_k can be recolored, possibly with the help of color 7, in such a way that no edge incident to L or M or a neighbor of either of L or M has both endpoints of the same color.*

Proof. After the initial coloring, the only edges which might have endpoints of the same color are the edges connecting vertices of L to vertices of M. Without loss of generality we may assume that the vertices of L and M use colors $1, 2, 3$, the inside of $G_{i,j}$ uses in addition colors $4, 5$, while the inside of $G_{i+1,j+1}$ uses colors $4, 6$ in addition to $1, 2, 3$.

The recoloring is done in two steps, where in the first step the conflict vertices of L and M (i.e. the ones with a neighbor of the same color) are recolored as indicated in Table 1.

Conflict vertex of	old color	new color
L	1	6
M	2	5
L	3	7

Table 1. First step of recoloring

This ensures that no edge incident to an *inner vertex* of L or M, i.e, different from $\{u_1, v_1, u_k, v_l\}$, has both endpoints of the same color since:

- if there was an edge from L to M connecting two vertices of the same color, one endpoint of this edge has been recolored.
- As no color 6 was used in $G_{i,j}$, the vertices of L recolored to 6 have no neighbors of color 6 in $G_{i,j}$ (and they cannot be neighbors, as both had color 1 in the coloring of $G_{i,j}$). They also do not have neighbors of color 6 in $G_{i+1,j+1}$ as all their neighbors in $G_{i+1,j+1}$ are on the outer face and thus of colors $1, 2, 3$ (and newly 5). Finally, from the second property of simple crossing it follows that the inner conflict vertices of L and M do not have neighbors in $G_{i,j+1}$ and $G_{i+1,j}$,
- Analogous argument applies for vertices recolored to 5 in M and vertices recolored to 7 in L.

It remains to consider edges incident to $\{u_1, v_1, u_k, v_l\}$ that might have endpoints of the same color; for example u_1 was recolored from 1 to 6 but it has a neighbor of color 6 in $S_{i,j+1}$ (note that there is no problem if the new color was 7). In such a case, these same-color neighbors in $S_{i,j+1}$ are recolored to color 7. We claim that this does not create same-color edges. First note that if u_1 recolored its neighbors of color 6 in $G_{i,j+1}$, then v_1 necessarily kept its original color, since v_1 might change its color only if it was originally 2. Hence, by the simplicity of the crossing, only the neighbors of v_1 of color 6 (or, by symmetry, only the neighbors of u_1 of color 5) need to be recolored to color 7. The cases for other extreme vertices of L and M are analogous. Since the width of the gap between the squares is greater than 3, it ensures that the recoloring of vertices in $G_{i,j+1}$ and $G_{i+1,j}$ does not create any conflict in coloring.

Furthermore, any two vertices of $G_{i,j+1}$ which were recolored to 7 due to different crossings with octagons cannot be neighbors since the size of the squares is $5 + \epsilon$.

While the width of the gap between the squares ensures the first condition for a crossing to be simple is always satisfied, there can be a case when several different vertices of L are connected both to a vertex of M and to a vertex in $G_{i,j+1}$ or $G_{i+1,j}$, see Figure 5. Notice that this may happen when some inner vertices of L or M are cut vertices in $G_{i,j}$ or $G_{i+1,j+1}$.

We resolve the problem of a crossing that is not simple by a pre-processing phase in which some of the vertices in the octagons are assigned to the neighboring squares, with the goal to make the crossing simple.

Consider a crossing $CR_{i,j}$ and let L and M be defined as before. Let u' be the last (in L) occurrence of a node connected to both M and $G_{i,j+1}$ (if there is no such node, set $u' = u_1$). Similarly, let u'' be the first occurrence in L of a node connected to both M and $G_{i+1,j}$. Define v' and v'' analogously in M, using clockwise traversal, see Figure 5. Any vertex of L which is connected to both $G_{i,j+1}$ and $G_{i+1,j+1}$ must occur in the segment of L from u_1 to u', since the edges incident with u' connecting it to $G_{i,j+1}$ and $G_{i+1,j+1}$ act as separators in the planar graph. Similarly, any node of L which is connected to both $G_{i+1,j}$ and $G_{i+1,j+1}$ must occur in the segment of L from u'' to u_k (see Figure 6).

We now partition L into three parts: Let L_1 be the shortest initial segment of L from u_1 to to first occurrence of u' so that all vertices connected to both $G_{i,j+1}$ and $G_{i+1,j+1}$ are contained in L_1, let L_3 be the shortest final segment of L starting with an occurrence of u'' so that all vertices connected to both $G_{i+1,j}$ and $G_{i+1,j+1}$ are contained in L_3, and L_2 be the remaining part of L. We define M_1, M_2, and M_3 analogously as segments of M using v' and v''.

To make the crossing simple, we assign the components of $G_{i,j}$ separated by u' and encountered in the traversal of L_1 to $G_{i,j+1}$, and the components of $G_{i,j}$ separated by u'' and encountered in the traversal of L_3 are assigned to are $G_{i+1,j}$. The same is applied to the segments of $G_{i+1,j+1}$ separated by v' and v'' and encountered in the traversal of M_1 and M_3, (see Figure 7). All the components that are assigned to $G_{i,j+1}$ are inside the area bordered by the

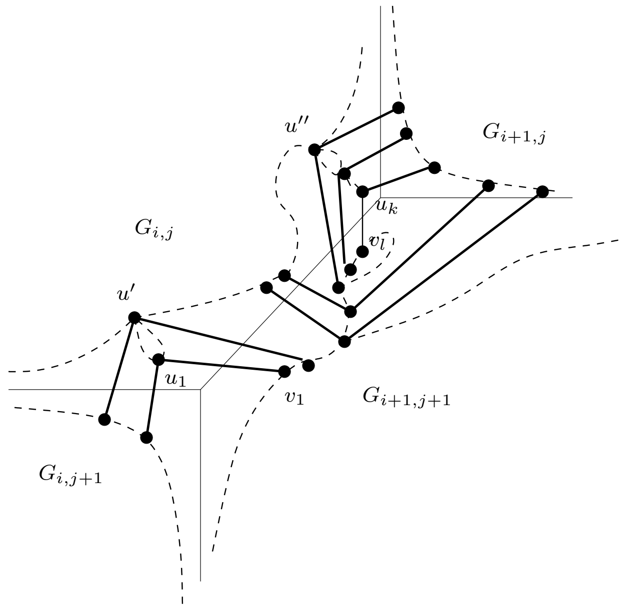

Fig. 5. A not simple crossing.

edges connecting u' or v' to $G_{i+1,j+1}$ and $G_{i,j+1}$ or $G_{i,j}$ and $G_{i,j+1}$. Similarly all the components that are assigned to $G_{i+1,j}$ are inside the area bordered by the edges connecting u'' or v'' to $G_{i+1,j+1}$ and $G_{i+1,j}$ or $G_{i,j}$ and $G_{i+1,j}$. Since the length of the crossing is more than 3, there cannot be any edge between vertices assigned to $G_{i+1,j}$ and $G_{i,j+1}$. Furthermore, after this reassignment, u' is the only vertex in $G_{i,j}$ that can be connected to both $G_{i+1,j+1}$ and $G_{i,j+1}$ and u'' is the only vertex in $G_{i,j}$ that can be connected to both $G_{i+1,j+1}$ and $G_{i+1,j}$. The analogous statement can be made about v' and v'', Thus the crossing L' and M' between $G_{i,j}$ and $G_{i+1,j+1}$ is a subset of u', L_2, u'' and v', M_2, v'' and this modified crossing $CR_{i,j}$ satisfies both conditions of a simple crossing, see Figure 7, and we can thus proceed with the coloring as stated in Lemma 3.

The following lemma, together with the size of the squares being selected as $5 + \epsilon$, allows us to apply Lemma 3 to each crossing independently.

Lemma 4. *Any vertex recolored due to resolving conflicts in crossing $CR_{i,j}$ is at a distance at most 2 from the line separating $S_{i,j}$ and $S_{i+1,j+1}$.*

3.3 Local 7-coloring algorithm

Putting the pieces together we have the following local, fully distributed algorithm that is executed at each vertex of the graph to obtain a valid 7-coloring of the graph.

The results of this section can be summarized in the following theorem.

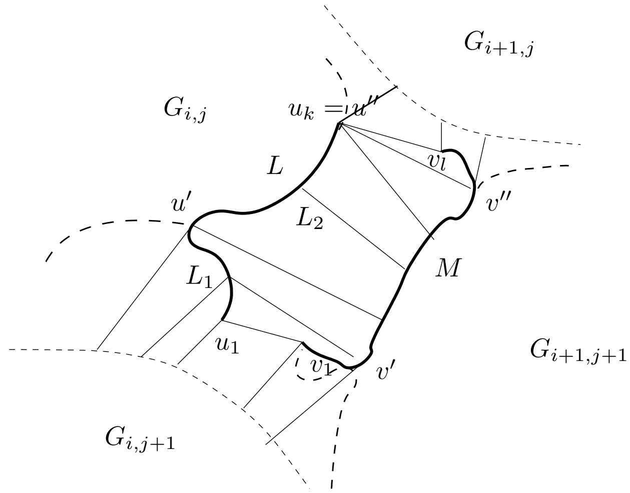

Fig. 6. L and M in a crossing.

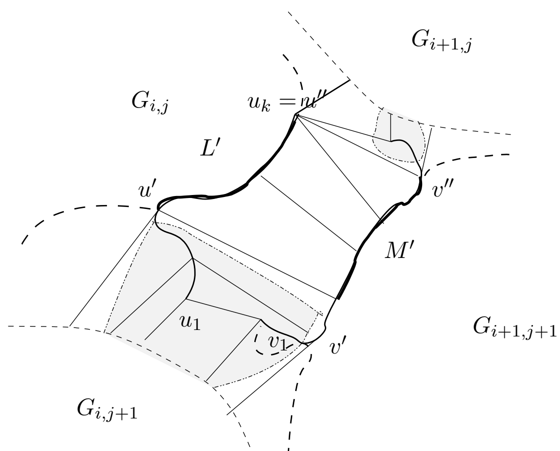

Fig. 7. L' and M' after the transformation. The shaded areas belong to $G_{i,j+1}$ and $G_{i+1,j}$.

Theorem 3. *Given a planar subgraph of the unit disk graph whose vertices correspond to hosts that are each aware of its geometric location in the plane, Algorithm 1 computes locally a 7-coloring of this subgraph using only information on the subgraph located within a constant number of hops away from it.*

Algorithm 1 The local 7-coloring algorithm for a vertex v

1: Learn your neighborhood up to distance 201 // *Note that all steps can be performed locally using the information learned in the first (communication) step, without incurring further communication.*

2: From your coordinates, identify the square/octagon $S_{i,j}$ you are located in, and calculate the connected component of $G_{i,j}$ you belong to.
// *The next step is for vertices near a crossing*

3: Calculate L and M, and then L' and M'. Determine whether you have been shifted to a neighboring square. Determine whether L' and M' are connected, if not but the squares are now connected, repeat the process until the final L^* and M^* are computed.

4: Apply the $3+2$ coloring algorithm from Theorem 1 for each $G_{i,j}$, as in Figure 3

5: Apply the recoloring from Lemma 3.

References

1. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7:609–616, 2001.
2. I. Caragiannis, A. V. Fishkin, C. Kaklamanis, and E. Papaioannou. A tight bound for online coloring of disk graphs. In A. Pelc and M. Raynal, editors, *SIROCCO 2005*, volume 3499 of *LNCS*, pages 78–88. Springer, 2005.
3. E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, and J. Urrutia. Local construction of planar spanners in unit disk graphs with irregular transmission ranges. In M. K. j. Correa, A. Hevia, editor, *LATIN 06*, volume 3387 of *LNCS*, pages 286–297. Springer Verlag, 2006.
4. P. Dörre. Every planar graph is 4-colourable and 5-choosable a joint proof. Fachhochschule Südwestfalen (University of Applied Sciences), unpublished note.
5. K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systemic Zoology*, 18:259–278, 1972.
6. S. Ghosh and M. Karaata. A self-stabilizing algorithm for coloring planar graphs. *Distributed Computing*, 7:55–59, 1993.
7. A. Gräf, M. Stumpf, and G. Weißenfels. On coloring unit disk graphs. *Algorithmica*, 20(3):277–293, 1998.
8. E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. of 11th Canadian Conference on Computational Geometry*, pages 51–54, August 1999.
9. N. Linial. Locality in distributed graph algorithms. *SIAM J. COMP.*, 21(1):193–201, 1992.
10. M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(1):59–68, 1995.
11. Y. Miyamoto and T. Matsui. Multicoloring unit disk graphs on triangular lattice points. In *SODA*, pages 895–896. SIAM, 2005.
12. C. Thomassen. Every planar graph is 5-choosable. *Combinatorial Theory Series B*, 62(1):180–181, Sep. 1994.
13. Z. Tuza and M. Voigt. A note on planar 5-list colouring: non-extendability at distance 4. *Discrete Mathematics*, 251(1):169–172, 2002.
14. Y. Wang and X.-Y. Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *DialM: Proceedings of the Discrete Algorithms and Methods for Mobile Computing & Communications*, 2003.