Local Algorithms for Edge Colorings in UDGs

Iyad Kanj DePaul University, Chicago

Joint Work With

Andreas Wiese Technische Universitat, Berlin
Fenghui Zhang Google, Seattle

Problem Definition

- Edge Coloring: color the edges of a given graph with the minimum number of colors such that no two adjacent edges receive the same color

- Strong Edge Coloring: color the edges of a given graph with the minimum number of colors such that no two edges of distance ≤ 2 receive the same color

- Both problems are NP-hard (on general graphs)

Edge Coloring: Known Results

- Under the Centralized Model:

Vizing [1964]	$\Delta + 1$	Non-constructive
Misra & Gries [1992]	$\Delta + 1$	Polynomial time
Ramanathan [1999]	Ratio 2	Simple greedy algorithm

Edge Coloring: Known Results

- Under the Distributed Model:

Gandam et al. [2005]	Δ +1
Nandagopal [2005]	2

Strong Edge Coloring: Known Results

- On planar graphs:

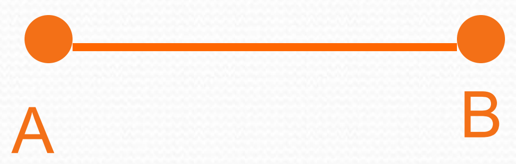
Barrett et al. [2006]	17
Ito et al. [2007]	2

Edge Colorings on UDGs and quasi-UDGs

- We consider the Edge Coloring and Strong Edge Coloring problems on UDGs and quasi-UDGs
- We are motivated by applications in wireless computing: channel assignment

Motivation: Wireless Ad Hoc Networks

- A wireless ad hoc network is commonly modeled as a UDG U:
  - Devices are the points of U in the 2-D Euclidean space



  - AB is an edge in U if and only if the two corresponding devices can communicate (i.e., $|AB| \leq 1$)

Quasi-UDGs

- A quasi-UDG U with parameter $0 < r \leq 1$ is a generalization of UDG:
  - if $|AB| \leq r$ then AB is an edge in U
  - if $|AB| > 1$ then AB is not an edge in U
  - if $r < |AB| \leq 1$ then AB may or may not be an edge in U

Motivation: Wireless Ad Hoc Networks

- Channel Assignment Problem: Assign the channels frequency ranges so that channels that are close to each other receive different frequency ranges

- This is equivalent to coloring the edges of the graph so that edges that are “close” to each others receive different colors

The Local Distributed Model

- A distributed algorithm is *k-local* if the computation at each node depends solely on the initial states of the nodes that are at most k hops away

- A distributed algorithm is *local* if it is k-local for some fixed integer k

- Local distributed algorithms are important, especially in wireless ad-hoc and sensor networks, because they are naturally scalable, robust, and fault tolerant

Our Results On quasi-UDGs: Edge Coloring

- For a quasi-UDG with parameter r:

Ratio	**Locality**
2	$(8r^2 + 64r + 88)/(\pi r^2)$
2	$(8r^2 + 40r + 40)/(\pi r^2)$

Our Results On UDGs: Edge Coloring

- By setting r=1 in the previous results for quasi-UDGs:

Ratio	**Locality**
2	50
2	28

Our Results On UDGs: Strong Edge Coloring

Ratio	**Locality**
128	22
10	10

Note. The techniques can be extended to quasi-UDGS

A Local Approximation Algorithm for Edge Coloring

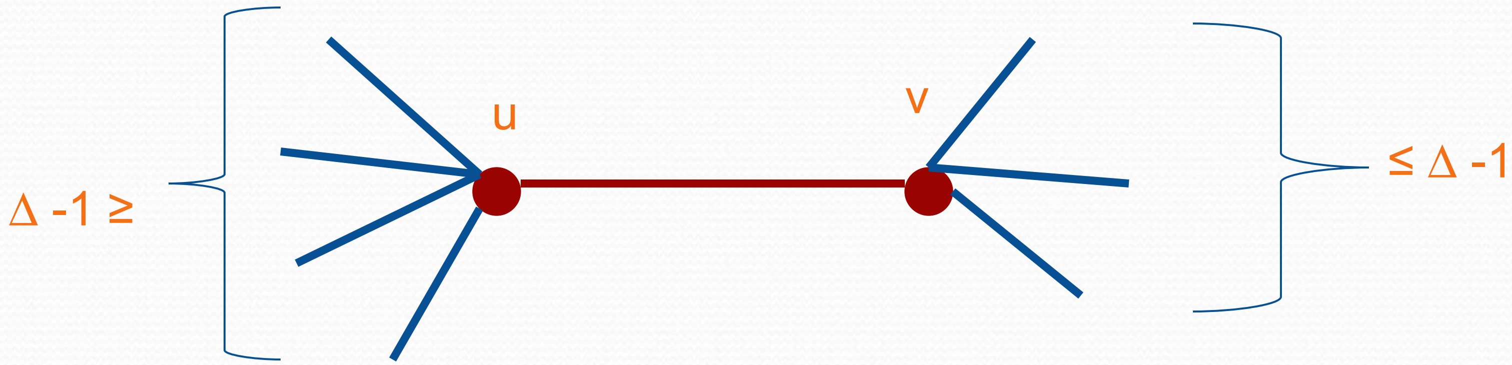
- The centralized greedy algorithm
  - Order the edges in the graph arbitrarily
  - For each edge e=(u, v) considered in the given order:

    color e with the smallest available color
    (i.e., the smallest color that no edge incident on u or v is colored with)

- The greedy algorithm has ratio 2

The Centralized Greedy Algorithm

- Let e be an edge with the highest color number
- Color(e) $\leq$ $(\Delta - 1) + (\Delta - 1) + 1 = 2\Delta - 1$



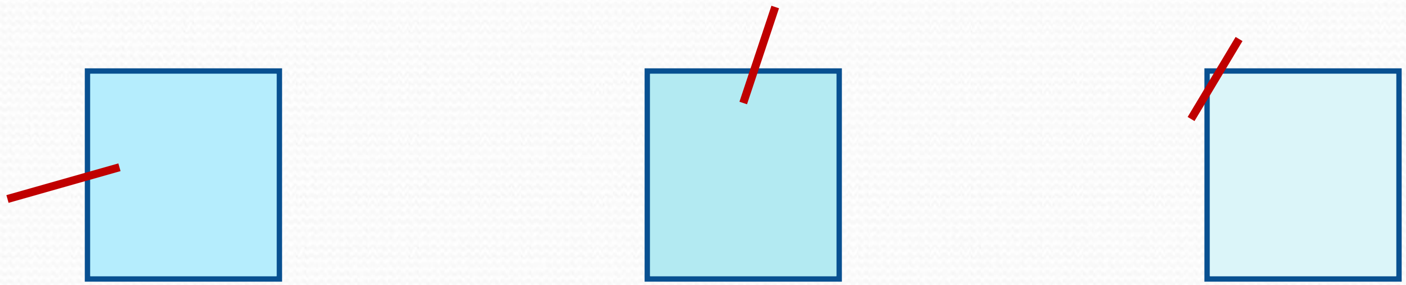
Clustering

- To adapt the greedy algorithm to the local model, we use a “clustering” idea

- We tile the plane with square tiles of dimension α, for some constant $\alpha > 2$

- All edges in the same connected component interior to a tile form a cluster

Edges Crossing Tiles



$T_H = (\alpha/2, 0)$ $T_V = (0, \alpha/2)$ $T_D = (\alpha/2, \alpha/2)$

Fact. For every edge e, there exists a translation $T \in \{T_I = (0,0), T_H, T_V, T_D\}$ such that $T(e)$ resides in some tile

The Local Algorithm Edge-Coloring-APX

1. For each translation $T \in \langle T_I, T_H, T_V, T_D \rangle$:

    1.1. All the points in U whose translations reside in the same connected component within a single tile form a cluster

    1.2. All the points in a cluster apply the centralized greedy algorithm to the subgraph of U induced by the points in the cluster

Note. When choosing the smallest available color, we take into account the colors assigned in previous rounds

The Local Algorithm Edge-Coloring-APX

Theorem. The algorithm Edge-Coloring-APX has ratio 2

Proof. The same proof used for the centralized greedy algorithm works

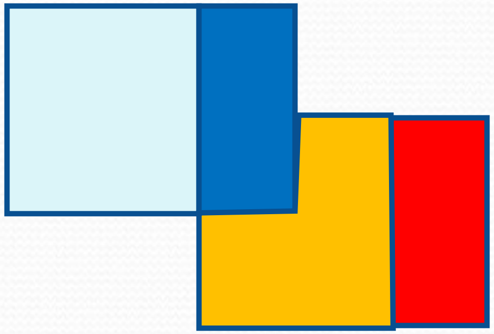
Theorem. The algorithm Edge-Coloring-APX is k-local where
$k = (22\,\alpha^2 + 32\,\alpha + 8)/\pi$

Proof. (Sketch)

- We use a geometric argument to bound the propagation of information
- For two edges e and e', we prove that if e affects e' then e' must reside within a bounded region in the vicinity of e
- More specifically, we prove that the endpoints of e and e' must be k-neighbors, where $k = (22\,\alpha^2 + 32\,\alpha + 8)/\pi$

Locality Bound

- Assume that the translations are applied in the following order: $< T_I, T_H, T_V, T_D >$



Locality Bound

- Any two points residing in this area must be is k-neighbors where $k = (22\alpha^2 + 32\alpha + 8)/\pi$

Concluding Remarks

- We gave local distributed algorithms for approximating Edge Coloring and Strong Edge Coloring on UDGs and quasi-UDGs

- The locality of the algorithms is high

- The complexity of the problems (on UDGs) is open