Improved Sparse Covers for Graphs Excluding a Fixed Minor

Ryan LaFortune (RPI), Costas Busch (LSU), and Srikanta Tirthapura (ISU)
Sparse Covers

- Distributed systems represented by graphs
- Fundamental data structure
- Numerous Applications
  - Name-independent compact routing schemes
  - Directories for mobile objects
  - Synchronizers
Definitions

- A cover is a collection of connected sets called clusters, where every node belongs to some cluster containing its entire k-neighborhood.
- Locality parameter
- Radius and degree (latency and load)
- Sparse cover
Extreme Cases

- Large radius, small degree
  - $G$ is one cluster
- Small radius, large degree
  - Every node forms a cluster
Sparse Cover for Arbitrary Graphs

- Impossible to achieve optimality in both metrics [Peleg: 2000]
  - Radius $O(k)$ and degree $O(1)$
- Best known algorithm [Awerbuch and Peleg: FOCS 1990]
  - Radius $O(k \log n)$ and degree $O(\log n)$
  - Based on coarsening
- Special graphs?
Contributions

- Improved algorithm for H-minor free graphs
  - Radius $\leq 4k$ and degree $O(\log n)$

- Improved algorithm for planar graphs
  - Radius $\leq 24k - 8$ and degree $\leq 18$
Outline of Talk

Consequences
Related Work
Shortest-Path Clustering
$P$-Path Separable Algorithm
Planar Algorithm
Conclusions
Name-Independent Compact Routing Schemes

- Deliver a message given the ID of the destination node
- Cannot alter IDs
- Tradeoff between stretch and memory overhead

- General algorithm [Awerbuch and Peleg: FOCS 1990]
  - Stretch = $O(\log n)$ and memory = $O(\log^2 n)$ bits per node

- Planar algorithm
  - Stretch = $O(1)$ and memory = $O(\log^2 n)$ bits per node

- H-minor free algorithm
  - Stretch = $O(1)$ and memory = $O(\log^3 n)$ bits per node
Directories for Mobile Objects

- Given an object’s name, returns its location
- Wireless sensor networks, cellular phone networks
- Tradeoff between $Stretch_{\text{find}}$ and $Stretch_{\text{move}}$

- General algorithm [Awerbuch and Peleg: JACM 1995]
  - $Stretch_{\text{find}} = O(\log^2 n)$ and $Stretch_{\text{move}} = O(\log^2 n)$

- Planar algorithm
  - $Stretch_{\text{find}} = O(1)$ and $Stretch_{\text{move}} = O(\log n)$

- H-minor free algorithm
  - $Stretch_{\text{find}} = O(\log n)$ and $Stretch_{\text{move}} = O(\log n)$
Synchronizers

- Distributed programs that allow the execution of synchronous algorithms in asynchronous systems
- Logical rounds simulate time rounds
- Tradeoff between time steps and average messages per node

**ZETA** [Shabtay and Segall: WDAG 1994]
- Time steps = \(O(\log z n)\) and messages = \(O(z)\) per node

**Planar algorithm**
- Time steps = \(O(1)\) and messages = \(O(1)\) per node

**H-minor free algorithm**
- Time steps = \(O(1)\) and messages = \(O(\log n)\) per node
Related Work

- Algorithm for graphs excluding $K_{r,r}$
  - Diameter $= 4(r + 1)^2 k$ and degree $= O(1)$
    - [Abraham et al.: SPAA 2007]

- Algorithm for $H$-minor free graphs
  - Weak diameter $= O(k)$ and degree $= O(1)$
    - [Klein et al.: STOC 1993]

- Algorithm for graphs with doubling dimension $a$
  - Radius $= O(k)$ and degree $= 4^a$
    - [Abraham et al.: ICDCS 2006]
Shortest-Path Clustering

- If called with 2k, the path can be removed
- All nodes are satisfied, radius $\leq 4k$, and $\text{deg} \leq 3$
H-Minor Free Definitions

- The *contraction* of edge $e = (u, v)$ is the replacement of $u$ and $v$ by a single vertex
- A *minor* of $G$ (subgraph after contractions)
- *H-minor free*
  - Trees, exclude $K_3$
  - Outerplanar graphs, exclude $K_4$ and $K_{2,3}$
  - Series-parallel graphs, exclude $K_4$
  - Planar graphs, exclude $K_5$ and $K_{3,3}$
$P$-Path Separable Algorithm

- **Path Separator** (shortest paths, components have at most n/2 nodes)
- Every H-minor free graph is $P$-path separable
  - $P$ depends on the size of H
    - [Abraham, Gavoille: PODC 2006]
- Recursively cluster path separators
Initial graph, suppose $k=1$
Choose a path separator
Break the path separator up into sub-paths of length $2k = 2$
Cluster $2k = 2$ around the first sub-path
Radius = $O(k)$ and degree $\leq 3$
\[ \leq \frac{n}{2} \text{ nodes} \quad \text{and} \quad \leq \frac{n}{2} \text{ nodes} \]
Continue Recursively
(terminates in a logarithmic number of steps)
Analysis

- Only satisfied nodes are removed, thus all nodes are satisfied
- **Shortest-Path Cluster** was always called with 2k, so clearly the radius is $O(k)$
- Degree is $O(\log n)$ due to the logarithmic number of steps
Planar Definitions

- The *external face* of a graph consists of the nodes and edges that surround it.
- The *depth* of a node is the minimum distance to an external node.
Planar Algorithm

- If \( \text{depth}(G) \leq k \), we only need to \( 2k \)-satisfy the external nodes to satisfy all of \( G \)
- Suppose that this is the case
Step 1: Take a shortest path (initially a single node)

Step 2: 4k-satisfy it

Step 3: Remove the 2k-neighborhood
Continue recursively…
4k-satisfy the path
Remove the 2k-neighborhood
Discard A, and continue
And so on ...
Analysis

- All nodes are satisfied because all external nodes are 2k-satisfied
- *Shortest-Path Cluster* was always called with 4k, so clearly the radius is $O(k)$
- Nodes are removed upon first or second clustering, so degree $\leq 6$
If depth(G) > k

- Satisfy one zone $S_i = G(W_{i-1} U W_i U W_{i+1})$ at a time
- Adjust for intra-band overlaps…
We can now cluster an entire planar graph.

Radius increased due to the depth of the zones, but is still $O(k)$.

Overlaps between bands increase the degree by a factor of 3, degree $\leq 18$. 
Conclusion

- We have significantly improved sparse cover construction techniques
  - H-minor free graphs
  - Planar graphs
- We can also construct optimal sparse covers for graphs with constant stretch spanners (unit disk graphs)
- Name-independent compact routing schemes, directories for mobile objects, and synchronizers