Bipartite Grid Partitioning of a Random Geometric Graph

Zizhen Chen and David W. Matula
Computer Science and Engineering Department
Southern Methodist University
Dallas, Texas, USA
Email: {zizhenc, matula}@smu.edu

***Abstract*—We investigate the problem of efficient computation of a partition of a Random Geometric Graph (RGG) into a limited number of densely packed bipartite grid subgraphs. The study focuses on the collection of subgraphs each individually having similar size and structure and the union employing most (e.g. over 85%) of the vertices. The residual vertices we seek to minimize are attributed to the inherent variations in densities of the randomly placed vertices and to any shortcomings of our greedy algorithms.**

RGG's have been extensively employed in recent times to model the deployment of numerous instances of Wireless Sensor Networks (WSN's) [1]. The properties investigated in our selected bipartite grid backbones are those deemed most relevant for applications to the foundations of this widely growing field.

Distributed algorithms are primarily used to determine backbones. Our results review what backbone grid partitions exist in the data. This provides a metric to measure the effectiveness of any distributed algorithm against an existing optimal result. The visual display of selected backbone grids suggests local algorithm design strategies. Furthermore, these partitions must be efficiently computable for highly scalable computation, e.g. WSN's with 100's of thousands of vertices and millions of edges in the resulting RGG.

We consider distributions over a segment of the plane and over the surface of the sphere to model sensor distributions both in limited planar regions, all around the globe or on distant planets.

***Keywords*-Random Geometric Graph; Wireless Sensor Network; Bipartite Subgraph; Grid Partitioning; Linear Time Algorithm**

I. Introduction and Problem Specification

WSN's composed of large numbers of inexpensive autonomous electronic sensors distributed over regions where the sensors communicate with each other wirelessly [2], [3] are becomming more prevalent [4]. The sensors employ low power multi-hop communication to gateway nodes used as the physical interface to the outside network. Kenniche and Ravelomananana [1] provide an excellent summary on why WSN's in practice are modeled by RGG's. WSN's are typically composed of a number of nodes which can be several orders of magnitude larger than the size of conventional networks [3]. Thus we require our algorithms on the RGG model to be scalable from thousands to millions of nodes. Sensor deployment is modeled in an RGG by placing points at random over a planar region or on a sphere, the latter to model undersea and planetary exploration [5].

Each point of the RGG monitors a small disc of radius r where other points within that disc are joined to the point by edges of the RGG to denote interference. The radius is assumed sufficiently small that only from ten's to hundred's of other sensors are neighbors (modeling low power sensor operation) yielding large sparse RGG's.

An independent set in the RGG is a set of points where no two are joined by an edge. Employing results from models of cellular systems, a triangular grid of points each at distance slightly greater than radius r then provides a maximum density of points [6]. This deployment corresponds to sensors that can all operate on the same frequency and cover most of the region from at least three points. This allows for triangulation to locate a specific object in the region.

Employing two such independent sets yields a bipartite grid such as in the haxagonal honeycomb lattice. This leads to the principle question we investigate in this paper.

Question: *Can we partition an RGG such that:*

- *the partition includes a reasonably small number (e.g. 10 - 20) of disjoint bipartite grids,*
- *the set of bipartite grids comprises a significant majority (e.g. over 80%) of the total number of vertices,*
- *each bipartite grid is similarly structured and covers most of the region with triple coverage?*

The partition into disjoint "backbone" dense grids allows each grid to model a set of sensors turned on for a brief period and rested to preserve power. Each grid requires only two frequencies for multihop ping-pong communication. When some sensors fail in some grids, the rotation between backbone grids sufficiently facilitates coverage of the whole region.

Our emperical study of large sparse RGG's demonstrates such partitions exist. While distributed sensor networks require distributed algorithms to form such backbones, it is important to first demonstrate that such partitions exist. Then distributed algorithms can be judged against an existing partition known to be essentially optimal.

The current paper expands our previous short presentations at workshops where only abstracts were archived [7], [8].

978-0-7695-6166-0/17 $31.00 © 2017 IEEE
DOI 10.1109/DCOSS.2017.31

In Section 2 we provide more details on bipartite lattice grids. We then characterize the bipartite grid partition we seek to find in a large sparse RGG.

In Section 3 we characterize random deployments of points over planar regions and the sphere. To determine for our investigation the edges of the RGG, we present an algorithm that requires us to check less than five vertex pairs for Euclidean distance per edge determined. This linear time RGG generation algorithm is essential to our investigation of large sparse RGG's.

In Section 4 we describe our backbone grid partition algorithm and its properties. The two phase algorithm first uses a particular sequential greedy coloring alogrithm empericaly demonstrated [9] to find near maximal triangulated "primary" independent sets. These sets are of similar size, collectively utilizing about half of the vertices. The second phase adaptive coloring algorithm operates to find and pair up somewhat sparser "relay" independent sets. All relay sets are of similar size and near triangular structure, collectively including about 40% of the vertices. The algorithm confirms in practice that a bipartite grid partition of 10 - 20 parts exists utilizing about 90% of the total number of the randomly distributed sensors in a WSN.

Section 5 provides experimental results on RGG's of sizes from 6400 to 128000. Visual displays of triangulations of the independent sets employing the theory of Gabriel graphs [10], [11] enhance the tables for assesing the structure of the independent set partitions. Visual displays of selected backbone grids from RGG's on 20000 vertices are shown to illustrate selected typical resulting grid structures.

Section 6 provides a summary of goals and results of this investigation.

II. Characterizing Bipartite Grid Partitions

Generally, the deployment (geometry) of backbones in WSN's can be various according to different application requirements [12]. Our approach is to build several disjoint dominating backbones with sensors deployed with multiple-coverage. In Figure 1, (a) and (b) provide two idealized manual placement grids (a bipartite planar hexagonal lattice and a bipartite planar Cartesian lattice). These grids can be offset and replicated k times to form k backbones using all vertices with face sizes six and four respectively. Bipartite graphs contain two independent sets where all edges are regularly placed only between vertices of different independent sets. Deployment with the bipartite feature has the property that two frequency channels allow routing the message in the backbone.

The honeycomb lattice grid in Figure 1(a) is a 6-coverage deployment and the Cartesian lattice grid in Figure 1(b) is a 4-coverage deployment. Figure 1(c) shows a bi-regular degree three and four lattice grid which has k-coverage $(4 \leq k \leq 7)$.

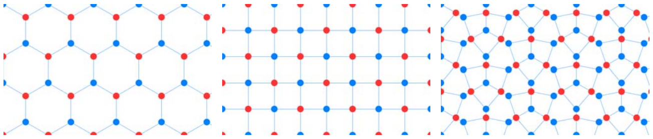

(a) Hexagon (b) Cartesian (c) Bi-regular 3,4

Figure 1: Lattice Grids with Face Size Six, Four and Combined

The bi-regular 3, 4 lattice grid provides an alternative deployment where the denser blue triangular lattice can primarily provide triple coverage and the sparser red triangular lattice can provide relays yeilding additional multiple connectivity for routers in each backbone.

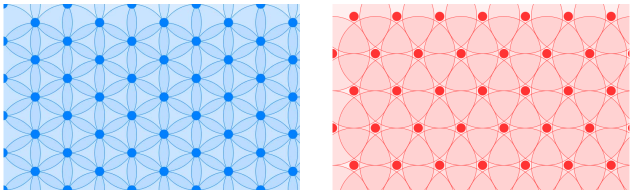

(a) Blue Independent Set (b) Red Independent Set

Figure 2: Coverages of Bi-regular 3, 4 Independent Sets

We prefer the multi grid approach offering better scalability and versatility. Energy savings and variable time to failure of sensor network nodes suggest redundant networks with higher connectivity than the single connection occuring in partitions into trees. With several disjoint backbone grids, we can sequentially rotate through them (duty cycle) to ensure full coverage and enlarge the whole network lifetime [13]. Requiring k-coverage rather than single coverage will increase the accuracy of tracking, improve robustness (fault-tolerance) and provide better performance in intruder detection application [14], [15].

The problem previously stated in this paper is: given a random deployment of sensors, can we partition the sensor nodes into a reasonably small number of disjoint bipartite grid backbones where the backbones in total have over half of the sensor nodes in denser primary (blue) independent sets. The remaining sensors are then sufficient to provide sparser relay (red) independent sets for each backbone yielding multiple connectivity for routing in each backbone.

This problem will be characterized by specifying a RGG and a desired bipartite component partition whose existance we shall demonstrate by a constructive algorithm.

Let a RGG denote a graph $G(N, r)$ with vertex set formed by choosing n points in a uniform random manner on the unit square, and introducing an edge between every vertex pair whose Euclidian distance is less than r. Let $V_1, V_2, ..., V_{k-1}$ be a partition of a majority of the vertices of $G(N, r)$ into disjoint sets where each set V_i (for $1 \leq i \leq k-1$) induces a connected bipartite subgraph of $G(N, r)$ composed of a primary and relay indepenent set. Specifically, we shall term $V_1, V_2, ..., V_{k-1}$ a bipartite component partition $BCP(\delta, \epsilon)$ of the RGG $G(N, r)$ if the union of the vertex sets V_i comprise $(1 - \delta)N$ of the vertices and if the induced subgraphs $\langle V_i \rangle$ (for $1 \leq i \leq k-1$) on average each dominate $(1 - \epsilon)N$ of the vertices. V_k is the residual vertex set not employed in the bipartite grid backbone partition. Our goal is to determine such partitions $BCP(\delta, \epsilon)$ for δ and ϵ suitably small.

III. Modeling Sensor Deployment

Regardless of the radio technology used, from our topological point of view, at any instant in time a WSN can be represented as a graph with a set of vertices consisting of the sensors of the network and a set of edges consisting of the communication links between the sensors [1].

A. Topology

We consider RGG's on three surfaces: the unit square, unit disk and unit sphere. The unit square allows visualizing scaling by considering 4 squares as four times the number of vertices with one half the value of the distance bound r yielding essentially the same number of backbones with only half the boundary bias. The unit disk removes the four corner small degree bias. The unit sphere removes the boundary bias entirely and allows an easy count of the number of faces without a boundary face bias. The sphere also allows for modeling sensor networks spanning our globe or distant planet.

B. Vertex distribution

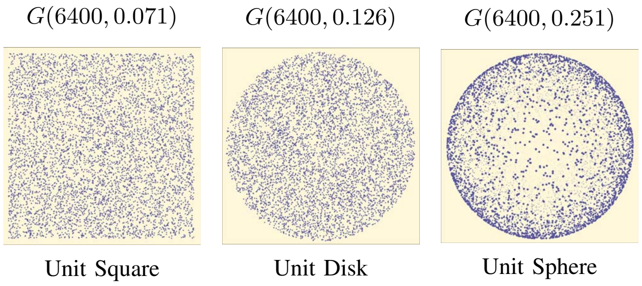
We consider uniformly random distribution of vertices over the surfaces. This provides a sufficient first approximation to sensor deployments subject to geographical constraints (e.g. in practice consider actual cell tower location vs the theoritical model [6]).

For different surfaces and sensor densities, we prefer to specify the parameters N and "expected average degree" of the RGG, letting r be determined by these specifications. Figure 3 shows sample graphs of 6400 vertex RGG's on our three surfaces with r values computed to yield expected average degrees of 100 in each case. Specifying the average degree provides convenient density parity over alternative surfaces for comparison of the number and quality of bipartite grid backbones in the partition.

Our backbone partition is topologically determined from the graph. We thus avoid further geographical (i.e. geometric) variations other than those due to the typical boundary regions of the unit square and unit disk.

C. Cell method edge generation algorithm

We employ a cell method for determining the edges of the RGG. This method requires only a small expected number of vertex pair distance computations per edge determination. This linear time generation ("Cell method" algorithm) of sample RGG's allows scalability to very large numbers of vertices in our studies.



$G(6400, 0.071)$ $G(6400, 0.126)$ $G(6400, 0.251)$

Unit Square Unit Disk Unit Sphere

Figure 3: RGG Deployment Region

Let $G(N, r)$ be the RGG construction illustrated in Figure 4 providing a cell method example on the unit square region. Divide the region into r by r cells. Note that only vertices located in the r radius circle are connected to each "center vertex" in the blue cell. With sequential traversal through the cells, for each vertex in the blue cell, you will only need to check all vertices in the red cells to determine its neighbors.

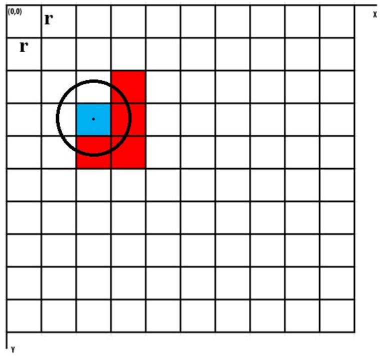


Figure 4: Cell Method Example

Let $\overline{d(G)}$ represent the average degree of the graph G. Then we estimate the value of $\overline{d(G)}$ by the ratio of the r radius circle area to the whole region area (which is 1 in this example).

$$\frac{\overline{d(G)}}{N} \approx \frac{\pi r^2}{1} \Rightarrow \overline{d(G)} \approx N\pi r^2 \tag{1}$$

Let E be the number of edges of this RGG, we know $\overline{d(G)} = \frac{2E}{N}$, then $\frac{2E}{N} \approx N\pi r^2$. With the number of pairs checked each time in the cell method being $5Nr^2$, we observe:

$$5Nr^2 < (2\pi)Nr^2 \approx \frac{4E}{N} < 4E = O(|E|) \tag{2}$$

Equation (2) confirms the linear time nature of our cell method algorithm. We also expanded the "Cell method algorithm" on to the surface of the sphere topology, but the details are beyond the scope of this paper.

IV. Backbone grids partition algorithm

We introduce an efficient (linear time) algorithm including a two-phase sequential coloring procedure (smallest-last coloring and adaptive coloring). We then employ only the topology (not the geometry) of $G(N, r)$ for partitioning vertices of the RGG into k disjoint sets $\{V_1, V_2, ..., V_k\}$. The induced subgraphs $\langle V_1\rangle, \langle V_2\rangle, ..., \langle V_{k-1}\rangle$ are then connected densely packed bipartite subgraphs.

A sequential coloring algorithm of a graph is an algorithm that operates in the following two stages: (1) Determine a sequence S of ordering vertices in the graph; (2) Greedy-Color the graph in the sequence of S. The Greedy-Color procedure assigns to a given vertex n in the sequence S the smallest color value that was not assigned to any previously colored neighbors of n.

A. *Smallest-last coloring*

Smallest-last coloring (SL-coloring) uses a greedy procedure to color the graph according to a "Smallest-last ordering (SL-ordering)" [16] sequence. To explain it briefly: sequentially delete the minimum degree vertex in the remaining graph and place on a stack until the graph is empty; then sequentially pop the stack and greedily color each vertex with the smallest color value not on a previously colored adjacent vertex. The algorithm pseudocode could be written as follow:

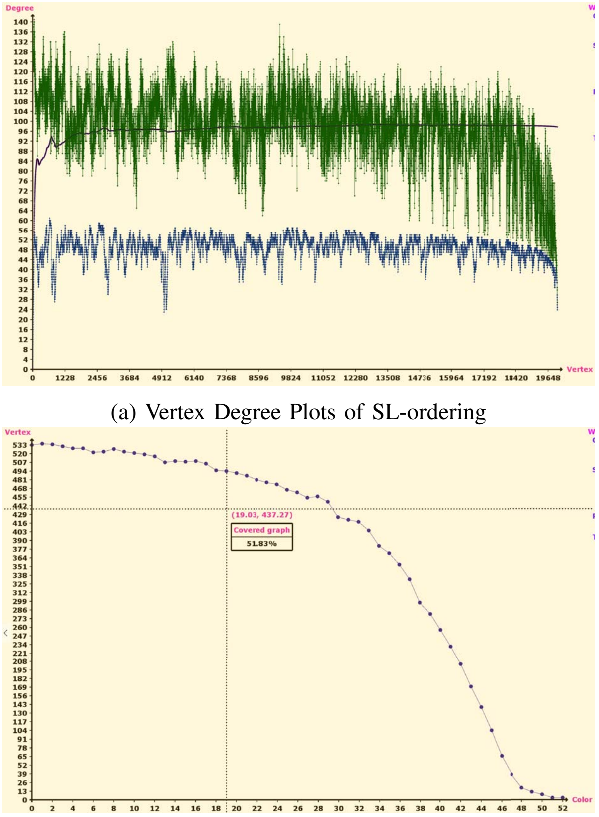
Algorithm 1 Smallest-last Coloring

```
Input: Graph $G(V, E)$
Output: Vertex Ordering $S$
  $S \leftarrow \emptyset$
  while $V \backslash S \neq \emptyset$ do
    append to $S$ the vertex with smallest degree
    in the subgraph induced by $V$ in $S$
  end while
  Greedy-Color $G \backslash S$ in inverse order of $S$
```

SL-ordering has been proved to be efficient (in linear time). It compares favorably to other sequential algorithms like Largest First, Lexicographic and Random [9], [16] in providing resulting independent sets of comparable size and structure for over half of the total vertices.

Figure 5(a) shows the plots of original degree (in green color at top), average degree (in black color at middle) and degree when deleted (in blue color at bottom) during the SL-ordering procedure. The plots are for one RGG on the unit square $G(20000, 0.040)$ with average degree 97.65.

Figure 5(b) shows a plot of the number of vertices in each color set after the SL-coloring procedure is completed on this RGG. We can see the initial several color sets each have a similar number of vertices and the total number of vertices in the initial color sets cover about half the number of vertices in the original RGG. In this case, the initial 19 color sets (from color # 0 to color #18) cover about 49.37% of the total vertices of the graph.

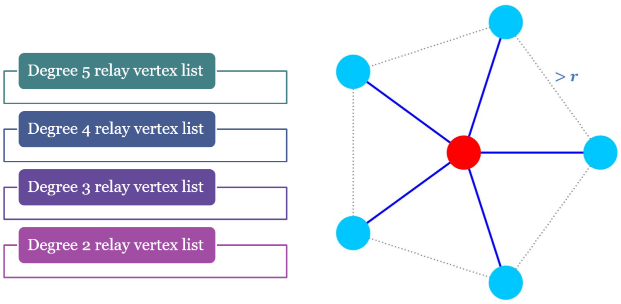


(a) Vertex Degree Plots of SL-ordering

(b) Color Size Plot of SL-coloring

Figure 5: Degree Plots and Color Size Plot of SL-coloring

B. *Adaptive coloring*


Degree 5 relay vertex list
Degree 4 relay vertex list
Degree 3 relay vertex list
Degree 2 relay vertex list
> r

(a) Relay Degree Lists

(b) Maximum One-color Adjacencies

Figure 6: Relay Degree Lists and Maximum Adjacency

With the observation that initial color sets have similar size after the SL-coloring procedure, we select the intial k color sets covering around 50% of the vertices of the orignal RGG as "primary independent sets" ("primary sets").

We term the remaining vertices "relay candidates" and recolor them with k colors paired to maximize adjacency to their cooresponding primary sets. The generated new k-color adaptive sets are called "relay independent sets" ("relay sets").

In order to maximize adjacency, we selected and ordered relay candidates according to the number of neighbors which belong to primary color sets. Therefore, the adaptive coloring is also a special sequential coloring algorithm. With simple geometric arguments we show that the number of neighbors with the same color can only be at most five. Figure 6(b) suggests that if we have 6 neighbors with the same color, we will form a hexagon contradicting the RGG rules. We can create 4 relay degree lists from degree 2 to degree 5. Note that each vertex of a relay set should connect to at least 2 primary set vertices to build a robust connected backbone. For each of the relay candidate vertices, we push the vertex into the relay list according to their maximum degree of same color neighbors as showing in Figure 6(a). Then we can use a degree list coloring strategy in the original smallest-last order [16] to color the relay list. This is also linear time $O(|V| + |E|)$. Since this adaptive sequential coloring alogrithm is colored according to the relay degree list, we also call it the adaptive "Relay Coloring" algorithm. The algorithm pseudocode could be written as follows:

Algorithm 2 Relay Coloring

```
Input: Graph $G(V, E)$ and relay lists $L[i](2 \leq i \leq 5)$
Output: Relay colors allocated to vertices in $V_r$ (relay candidates set)
  for $i = 5$ to $2$ do
    while !$L[i]$.isEmpty() do
      Try to assign a relay color paired with its primary neighbor color
      if the relay color is aleady assigned to one of its relay neighbour vetices then
        continue
      else
        assign this relay color
      end if
    end while
  end for
```

Figure 7 shows the color size plot after the adaptive sequential coloring procedure. In the figure, you can see both the SL-coloring color size plot and the second part relay coloring color size plot. Note that the gap between the color size of the primary sets and the relay sets. This corresponds to the bi-regular 3, 4 lattice grid in Figure 1 (c). The bipartite grid has primary independent sets that are more dense than the relay independent sets where both independent sets are triangular lattices.

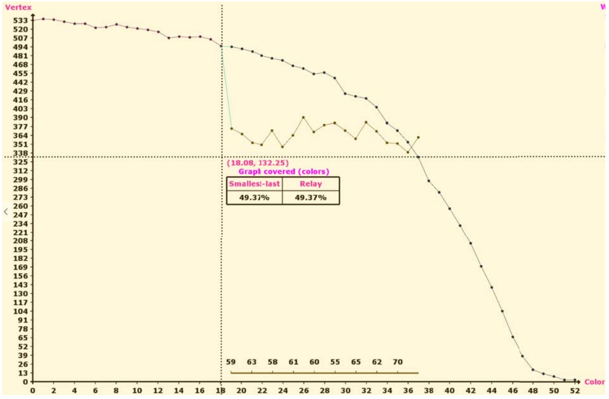


Figure 7: Color Size Plot after Adaptive Relay Coloring

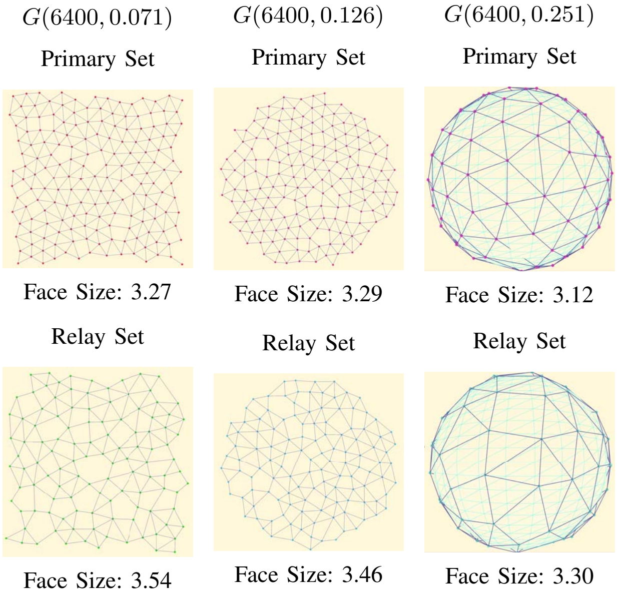


Figure 8: Average Face Sizes of Primary and Relay Sets with Grabriel Rules for Edge Generation yielding Planar Graphs

V. Examples with Visual Displays

We employ the theory of Gabriel graphs to determine if our generated backbones are close to the bi-regular lattice. The Gabriel graph is a subgraph of the Delaunay triangulation. It can be found in linear time if the Delaunay triangulation is given [10], [11]. After extensive tests, we found the Gabriel graph has average face size (how many edges one face of the graph has on average) about 3. This means the Gabriel graphs are close to triangular lattices. The relay sets are sparser than the primary sets. Then we can say our generated backbones are close to the manual bi-regular 3, 4 lattice placement illustrated in Figure 1(c). Figure 8 shows the Gabriel rules applied to one primary independent set and one relay independent set generated from the three RGG topologies of Figure 3. You can see in the unit sphere

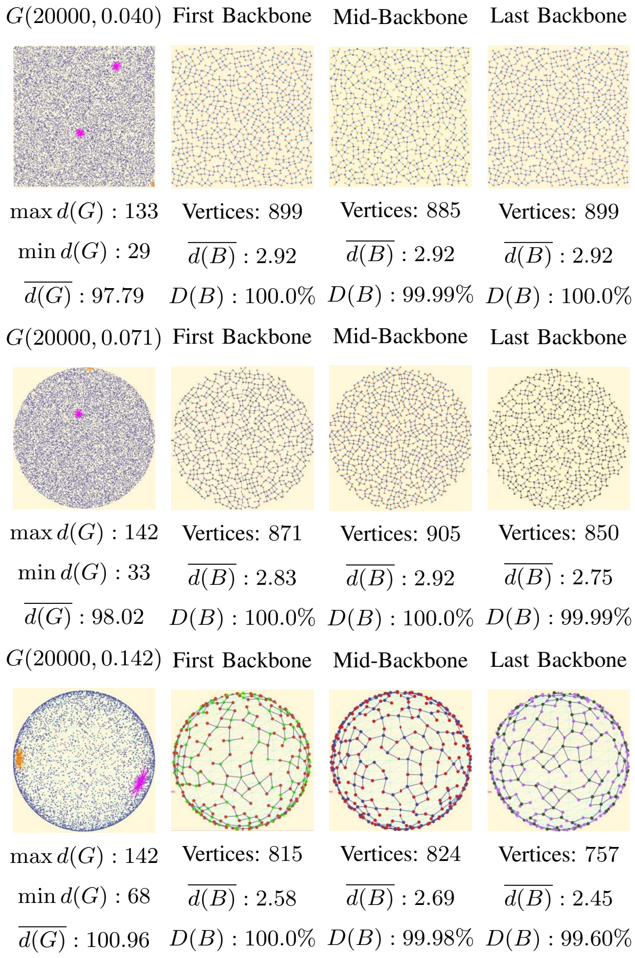


Figure 9: Screenshots of Unit Square, Disk and Sphere

model, the average face sizes on both primary sets and relay sets are closer to 3 compared to the other 2 topologies. The reason is the other 2 topologies are on a plane which has a large boundary face size bias effect.

We implemented a tool for calculating test benchmarks with a 3D display feature. This feature emperically discovered and verified that such backbone partitions exist employing relatively few dense backbones. The fact that there is little loss to random distribution noise is of interest to the rapidly growing field of WSN's [17], [18], [19].

Figure 9 shows some example screenshots of our test benchmarks of RGG's of 20000 vertices with average degree around 100. Let $d(G)$ represent the degree of the RGG, then $\max d(G)$, $\min d(G)$ and $\overline{d(G)}$ mean the maximum, minimum and average degree of the graph respectively. We use $d(B)$ to represent the degree of the backbone and $D(B)$ to represent the percentage of the vertices of the original RGG covered by the vertices of the backbone (domination percentage).

Tables I to III show detailed data from benchmarks of RGG's of average degree around 80 with vertices running from 8000 to 128000 on the unit square, disk and sphere. We define the large connected component of a generated bipartite subgraph as a "backbone". We refined the "backbone" further by sequentially deleting the "tails" (vertices of degree one) resulting in a subgraph with minimum degree 2 of the original "backbone" which is termed a "2-core". This could meet some requirements for situations with higher backbone quality demands. Here we list the meaning of some expressions in the following tables:

$\overline{N(B)}$: the average number of vertices in a backbone;
$\overline{Avg(d(B))}$: the average value of the average degrees of all backbones;
$\overline{Avg(F(B))}$: the average value of the average face sizes of all backbones;
$\overline{D(B)}$: the average domination percentage of the backbones.

Table I: Unit Square Test Benchmarks

N	8000	16000	32000	64000	128000
r	0.057	0.040	0.028	0.020	0.014
$\overline{d(G)}$	76.52	77.99	78.99	79.62	79.97
Backbones	17	18	18	18	18
Surplus	10.16%	8.01%	8.51%	8.45%	8.51%
$\overline{N(B)}$	422.76	817.67	1626.50	3255.28	6505.89
$\overline{Avg(d(B))}$	2.67	2.64	2.66	2.67	2.69
$\overline{Avg(F(B))}$	8.07	8.49	8.32	8.30	8.08
$\overline{Avg(F(\text{2-core}))}$	6.80	6.84	6.77	6.76	6.72
$\overline{D(B)}$	99.58%	99.12%	99.05%	99.48%	99.37%

Table II: Unit Disk Test Benchmarks

N	8000	16000	32000	64000	128000
r	0.101	0.071	0.050	0.036	0.025
$\overline{d(G)}$	77.30	78.86	79.31	79.85	80.17
Backbones	17	18	18	18	18
Surplus	10.29%	8.74%	8.49%	8.12%	8.65%
$\overline{N(B)}$	422.18	811.22	1626.83	3266.94	6496.00
$\overline{Avg(d(B))}$	2.67	2.65	2.66	2.68	2.69
$\overline{Avg(F(B))}$	8.12	8.37	8.34	8.18	8.09
$\overline{Avg(F(\text{2-core}))}$	6.73	6.80	6.75	6.73	6.72
$\overline{D(B)}$	99.78%	99.07%	99.36%	99.67%	99.44%

Table III: Unit Sphere Test Benchmarks

N	8000	16000	32000	64000	128000
r	0.201	0.142	0.100	0.071	0.050
$\overline{d(G)}$	81.00	80.91	80.99	80.95	80.98
Backbones	18	18	18	18	18
Surplus	8.66%	8.26%	8.26%	8.37%	8.34%
$\overline{N(B)}$	406.67	815.44	1630.94	3258.06	6517.94
$\overline{Avg(d(B))}$	2.72	2.72	2.71	2.72	2.72
$\overline{Avg(F(B))}$	7.67	7.78	7.78	7.83	7.83
$\overline{Avg(F(\text{2-core}))}$	6.63	6.60	6.64	6.63	6.64
$\overline{D(B)}$	99.86%	99.91%	99.82%	99.86%	99.85%

VI. Summary

In this paper we investigated RGG's in order to improve our design of WSN's. For this investigation we needed to generate large sparse RGG's. We introduced an algorithm for constructing such RGG's where we only needed to check at most an average of five vertex pairs for a Euclidean distance threshold per edge determined for the RGG. This yielded a powerful and efficient linear time RGG creation algorithm as a byproduct of our WSN design effort. We followed the paradigm of determining a partition of the RGG into dense backbone subgraphs each essentially covering the region with the desired triple coverage. Our backbones also provided higher connectivity visually yielding a "giant 2-connected component" with great coverage on the sphere. Our bipartite grid backbones are seen to preserve near triangular lattice form locally but drifting away from perfect placement on a larger scale. This result should encourage distributed algorithm development to follow finding dense local packing of sensor connections. In general the visual display of results of our two phase coloring procedure for determining both independent sets and the bipartite backbones is proposed as a valuable teaching tool for devising further backbone algorithms. Further video display of the bipartite backbones provides an even more powerful design tool but that is beyond the capabilities of this print medium.

Acknowledgment

The authors would like to thank the referees for citing more references broadening our presentation.

References

[1] H. Kenniche and V. Ravelomananana, "Random geometric graphs as model of wireless sensor networks," in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 4. IEEE, 2010, pp. 103–107.

[2] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *IEEE internet computing*, vol. 10, no. 2, pp. 18–25, 2006.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[4] P. Morreale, F. Qi, and P. Croft, "A green wireless sensor network for environmental monitoring and risk identification," *International Journal of Sensor Networks*, vol. 10, no. 1-2, pp. 73–82, 2011.

[5] E. Del Re, R. Pucci, and L. S. Ronga, "Ieee802. 15.4 wireless sensor network in mars exploration scenario," in *Satellite and Space Communications, 2009. IWSSC 2009. International Workshop on*. IEEE, 2009, pp. 284–288.

[6] V. H. Mac Donald, "Advanced mobile phone service: The cellular concept," *Bell System Technical Journal*, vol. 58, no. 1, pp. 15–41, 1979.

[7] Z. Chen and D. W. Matula, "Partitioning RGG's into disjoint $(1-\epsilon)$ dominant bipartite subgraphs," in *CSC14: The Sixth SIAM Workshop on Combinatorial Scientific Computing*. SIAM, 2014, pp. 48–50.

[8] Z. Chen and D. W. Matula. (2016, July) Partitioning random geometric graphs into bipartite backbones. SIAM Workshop on Network Science Program Abstracts. SIAM. [Online]. Available: http://www.siam.org/meetings/ns16/program.php

[9] D. Mahjoub and D. W. Matula, "Employing (1- ε) dominating set partitions as backbones in wireless sensor networks," in *Proceedings of the Meeting on Algorithm Engineering & Expermiments*. Society for Industrial and Applied Mathematics, 2010, pp. 98–112.

[10] D. W. Matula and R. R. Sokal, "Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane," *Geographical analysis*, vol. 12, no. 3, pp. 205–222, 1980.

[11] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Biology*, vol. 18, no. 3, pp. 259–278, 1969.

[12] W. Y. Poe and J. B. Schmitt, "Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay," in *Asian Internet Engineering Conference*. ACM, 2009, pp. 77–84.

[13] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1-2, pp. 89–124, 2005.

[14] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 144–158.

[15] P.-J. Wan and C.-W. Yi, "Coverage by randomly deployed wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2658–2669, 2006.

[16] D. W. Matula and L. L. Beck, "Smallest-last ordering and clustering and graph coloring algorithms," *Journal of the ACM (JACM)*, vol. 30, no. 3, pp. 417–427, 1983.

[17] P. Kumar Sahoo, M.-J. Chiang, and S.-L. Wu, "An efficient distributed coverage hole detection protocol for wireless sensor networks," *Sensors*, vol. 16, no. 3, p. 386, 2016.

[18] J. Yu, Q. Zhang, D. Yu, C. Chen, and G. Wang, "Domatic partition in homogeneous wireless sensor networks," *Journal of Network and Computer Applications*, vol. 37, pp. 186–193, 2014.

[19] S. Pandit, S. V. Pemmaraju, and K. Varadarajan, "Approximation algorithms for domatic partitions," 2009.