Determining Bipartite Backbones in a Random Geometric Graph

Zizhen Chen* David W. Matula*

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 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.

Random geometric graphs have been extensively employed in recent times to model the deployment of numerous instances of wireless sensor networks (WSN's). The properties investigated in our selected bipartite backbone grids are those deemed most relevant for applications to the foundations of this widely growing field.

A companion goal of our proposal is to develop metrics for documenting the quality of the WSN backbone grids. The metrics must be meaningful for WSN application while also being 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 regions and all around the globe.

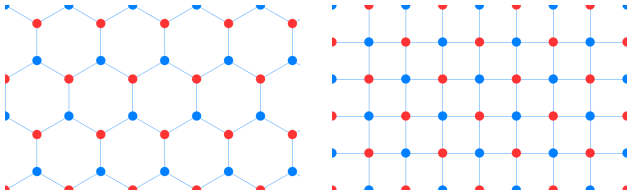
1 Introduction.

Wireless Sensor Networks (WSN's) are becoming more and more prevalent nowadays as an emerging technology with various military and civilian applications like habitat and ecosystem monitoring, seismic monitoring, civil structural health monitoring, groundwater contamination monitoring, outer space investigations[4] and so on. WSN's are made up of autonomous electronic sensors distributed across a region where the sensors communicate with each other wirelessly[9, 5, 15]. The ability to quickly throw together a network of sensors to monitor physical conditions (like sound, temperature, humidity and so on) over a region without the need for a physical infrastructure (beyond the gateway nodes used as a physical interface to the network) makes WSN's attractive for quick and cheap implementation. Each sensor has a limited wireless communication distance to its neighbors, which suggests using an RGG concept in computer science to model WSN's by placing a random set of points either in a planar region or over the sphere is reasonable and practical. The general goal of such a model is to determine disjoint subsets of the sensors that each can serve as a backbone for monitoring the whole region[13, 8].

The challenge for WSN deployment is: given numerous randomly placed wireless sensors, how can we organize them into multiple communicating network grids (backbones) each covering the region[11, 10, 12, 7]?

2 Problem Specification.

Generally, the deployment (geometry) of backbones in WSN's can be various according to different application requirements. One approach is to build a single clustered backbone with minimal cardinality where each sensor should connect to a backbone vertex (sensor) where the backbone sensors are deployed with minimal overlap space (like hexagonal (honeycomb) lattice[14]). Another approach is to build several disjoint dominating backbones with sensors deployed with multi-coverage. Figure 1 provides two idealized manual placement grids (bipartite planar hexagonal lattice and bipartite planar cartesian lattice) that can be offset and replicated k times to form k backbones using all vertices with face sizes six and four respectively. Bipartite graph contains two independent sets where no two vertices of which are adjacent. Deployment with bipartite feature has the ability to send two single channels routing the message in the backbone without interference each other.



(a) Hexagonal Lattice Grid (b) Cartesian Lattice Grid

Figure 1: Two lattice grids with face size six and four

We prefer the multi grid approach offering better scalability and versatility. Energy savings and variable time to failure of sensor network modes suggest redundant networks with higher connectivity than tree like parts is desirable. With several disjoint backbones, we can rotate them (duty cycle) to ensure full coverage and enlarge the whole network lifetime. Requiring k-coverage rather than 1-coverage will increase accuracy of

*Southern Methodist University.

tracking, improve robustness or fault-tolerance and better performance in intruder detection application[12]. The honeycomb lattice grid in Figure 1 (a) is a 6-coverage deployment and the Cartesian lattice grid is a 4-coverage deployment. Figure 2 shows a combination of regular degree three and four (bi-regular 3,4) lattice grids which has k-coverage ($4 \leq k \leq 7$, Figure 3 shows the individual coverage of each independent set of the bi-regular deployment) and provides a more compact deployment with a better energy saving ability because of shorter average distance (requiring less transmission power) between many neighbor sensor pairs. Our question becomes if points are distributed randomly, can we expect to find possibly distorted disjoint 2-connected grids with similar domination and patterns of face size primarily between four and six (like Figure 2 shows the bi-regular 3,4 lattice)?

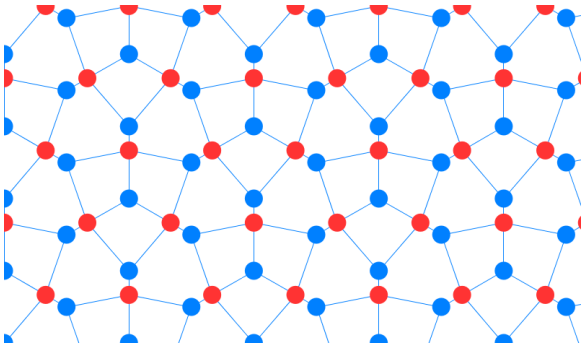
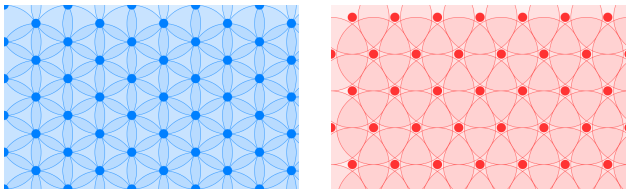


Figure 2: Bi-regular 3,4 Lattice Grid



(a) Blue independent set (b) Red independent set

Figure 3: Coverages of bi-regular 3,4 independent sets

Let a random geometric graph (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. Our problem is to partition vertices into k disjoint sets $\{V_1, V_2, ..., V_k\}$ whose induced subgraphs $\langle V_1\rangle, \langle V_2\rangle, ..., \langle V_{k-1}\rangle$ are connected bipartite subgraphs with each part an independent set that dominates all or nearly all N vertices of $G(N, r)$. 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 $G(N, r)$. Specifically, we shall term $V_1, V_2, ..., V_{k-1}$ a bipartite component partition $BCP(\delta, \epsilon)$ of the random geometric graph $G(N, r)$ 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 backbone partition. Our goal is to determine such partitions $BCP(\delta, \epsilon)$ for δ and ϵ suitably small, practically for example, with $\delta \approx 1/k$ and $\epsilon < 0.01$.

3 Design Considerations.

Regardless of the radio technology used, from the topology 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 links between the sensors. We use a random geometric graph (RGG) concept in computer science to model WSNs by placing n vertices uniformly at random in a region (e.g. unit square) and connecting two vertices if their Euclidian distance is at most r [6].

3.1 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 backbone with only half the boundary bias. The unit disk removes the four corner small degree bias. The unit sphere removes the boundary bias and allows an easy count of the number of faces without a boundary face bias. The sphere also allows for modeling sensor networks spanning the globe.

3.2 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. consider actual cell tower location).

For different surfaces and sensor densities, we prefer to specify the parameters N and "average degree" of the RGG, letting r be determined by these specifications. Figure 4 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 surface for comparison of the number and quality of bipartite backbone in the partition.

Our backbone partition is topologically determined from the graph, we thus avoid further geographical (geometric) variations other than those due to the

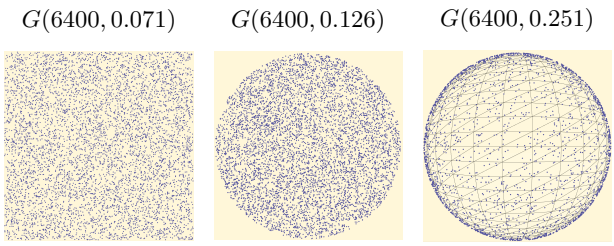


Figure 4: Random Geometric Topologies

typical boundary regions of the unit square and unit disk.

3.3 Edge generation for the RGG's 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 number of vertices in our studies. Let $G(N, r)$ be the RGG shows in Figure 5 providing a cell method example on unit square region: Divide the region into r by r cells, only vertices located in the r radius circle are connected to the vertices in the blue cell, so with sequential operation, for each vertex in blue cell, you only need check all vertices in red cells to determine its neighbors. Let E be the number of edges of this RGG,

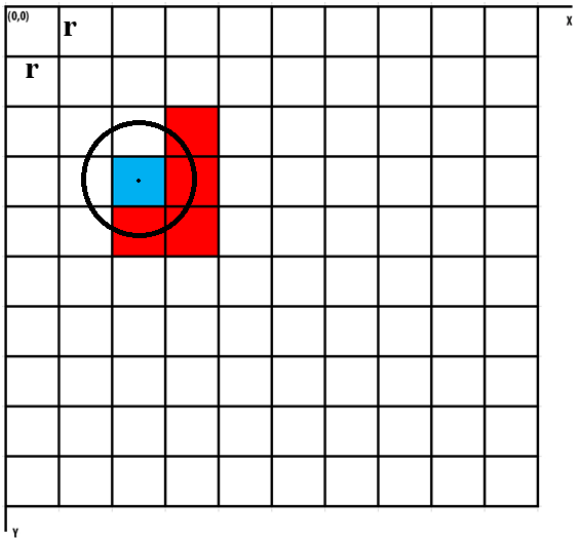


Figure 5: Cell method Example

then calculating the average degree in two different way, we will get $\frac{2E}{N} \approx N\pi r^2$. With the number of pairs checked each time in cell method which is $5Nr^2$, we observe:

$$5Nr^2 < (2\pi)Nr^2 \approx \frac{4E}{N} < 4E = O(E) \quad (3.1)$$

The equation (3.1) shows the linear time nature of our cell method.

4 Algorithms.

We introduce an efficient (linear time) algorithm including two-phase sequential coloring procedure (smallest-last coloring and adaptive coloring) employing only the topology (not the geometry) of $G(N, r)$ for partitioning vertices of the RGG into k disjoint sets $\{V_1, V_2, ..., V_k\}$ whose induced subgraphs $\langle V_1\rangle, \langle V_2\rangle, ..., \langle V_{k-1}\rangle$ are connected densely packed bipartite subgraphs with each part an independent set that dominates all or nearly all N vertices of $G(N, r)$.

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.

4.1 Smallest-last coloring. Smallest-last coloring uses greedy way to color the graph according to a "Smallest-last Ordering"[1] sequence. To explain it in detail: 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: Smallest-last

Algorithm 1 SmallestLastColoring

Input: Graph $G(V, E)$
Output: Vertex Ordering S

```
S ← ∅
while V\S ≠ ∅ do
    append to S the vertex with smallest degree
    in the subgraph induced by V n S
end while
Greedy-Color G\S in inverse oder of S
```

Ordering has proven to be efficient (in linear time) on several classes of graphs compared to several different sequential methods like Largest First, Lexicographic and Random[3, 1] and results in a sequence with vertices of fairly stable number of neighbors to be colored which yields a large complete subgraph generally close or equal to the chromatic number. This is a desirable feature for getting several dominant independent sets

with fair same sizes. Figure 6 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 smallest-last ordering process for RGG of unit square $G(20000, 0.040)$ with average degree 97.65. Figure 7 shows color size plot which is a plot of

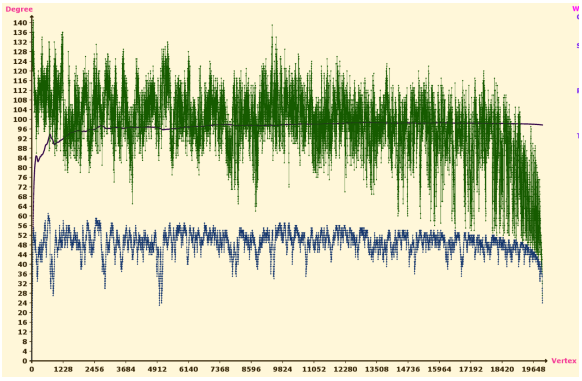


Figure 6: Degree Vertex Plots of Smallest-last Ordering

the number of vertices in each color set after greedy coloring process complete on this RGG. We can see the initial several color sets each having fairly same amount of vertices, in this case, the initial 19 (from color # 0 to color #18) color sets covers about 49.37% vertices of the whole graph.

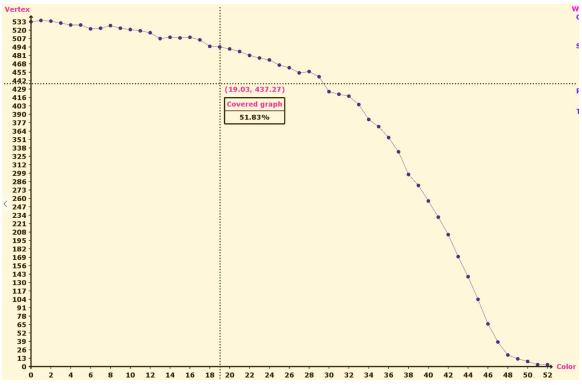
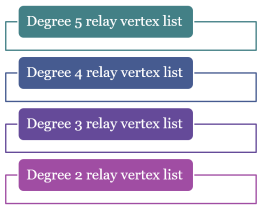
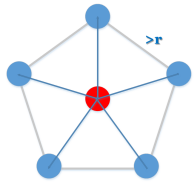


Figure 7: Color Size Plot of Smallest-last Coloring

4.2 Adaptive coloring. With the observation of initial color sets having fairly same size of dominant independent sets after smallest-last coloring procedure, we select the intial k color sets covering around 50% vertices of the orignal RGG as "primary independent sets" or "primary sets". In order to make full use of the residual vertices, we re-color them with colors paired to maximize adjacency with former cooresponding primary set. The generated new k-color adaptive sets are called "relay independent sets" or "relay sets", then the residual vertices for recoloring can be termed as "relay candidates". For adaptively recolor the relay candidates to paired with primary sets with maximize adjacency, we have to select and order relay candidates according to the amount of neighbors which belongs to primary color sets. Therefore, the adaptive coloring is also a special sequential coloring algorithm. With simple geometric provement we can know the maximum number of one color neighbors of a vertex can only be five like Figure 8 (b) shows (assuming we have 6 one color neighbors which will form a honeycomb structure and causes the one color neighbors connecting together then violated the one color independent set constraint). We can create 4 relay degree lists (From degree 2 to degree 5 since each vertex belongs to relay sets should connect at least 2 primary set vertices to form a robust connected backbone). For each vertex in relay candidates, we push them into the relay list according to their maximum one color neighbor degree like Figure 8 (a) shows. Then we can use a similar degree list coloring strategy in smallest-last coloring[1] to color the relay list and obviously, it is also linear time $O(|V| + |E|)$ with V is the number of vertices in the graph and E is the number of edges in the graph. The algorithm pseudocode could



(a) Relay Degree Lists



(b) Maximum One-color Neighbors

Figure 8: Relay Degree Lists and Maximum Neighbors

be written as follow: Figure 9 shows Color-size plot after adaptive coloring procedure. In the figure, you can see both smallest-last coloring color size plot and the second part adaptive coloring color size plot. It looks like you selected partial residual vertices and let them "jump-up" in the plot to pair with each primary set and these relay sets are also fairly same size with a little gap in the amount of primary sets. And now we may recall our bi-regular 3,4 lattice grid which has the similar feature that one independent set is more density while the other independent set is a little sparser compared to the density set in vertex amount and both independent sets are triangular lattices.

To check whether our generated backbones are close to the bi-regular lattice, we add gabriel rule to its primary and relay sets and after massive test

Algorithm 2 AdaptiveColoring

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
```

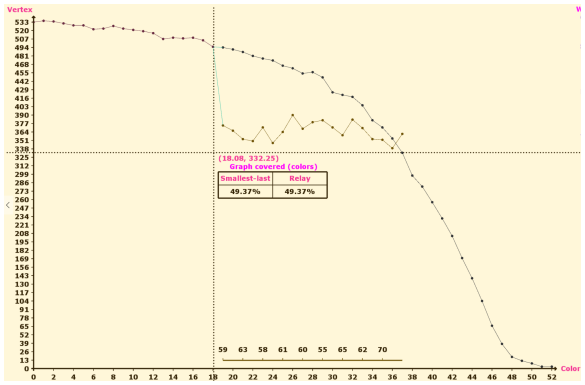


Figure 9: Color Size Plot of Adaptive Coloring

experiments, we found the gabriel graph has average face size close to 3 which means they are close to all triangular lattice just the relay set are sparser in amount. Then we can say our generated backbones are close to the manual bi-regular placement. Figure 10 shows the grabriel rules added to one primary and one relay independent set example generated from the three RGG's of Figure 4 with three topologies. You can see in unit sphere model, the average face sizes on both primary set and relay set are more close to 3 compared to other 2 topologies, the reason it other 2 topologies are all on a plane which a large boundary faze size bias effect.

5 Conclusion and results.

We investigate the problem of existence and a new method to partition a random geometric graph into k disjoint subgraphs satisfying the following conditions.

All but one of the subgraphs are connected ($1 -$

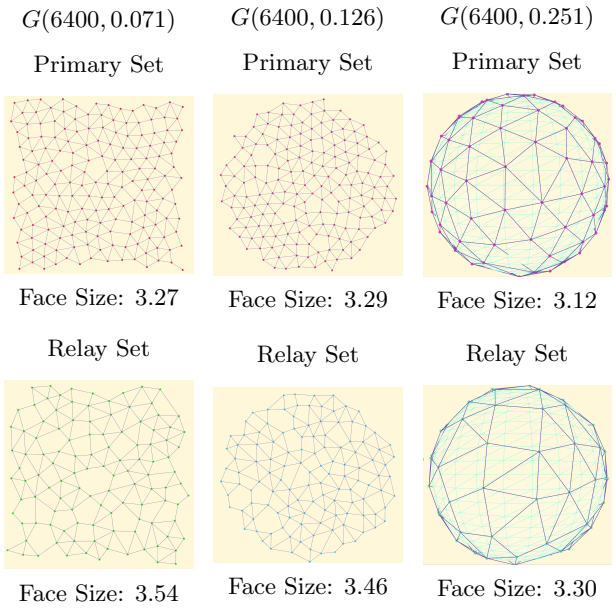


Figure 10: Average Face Sizes of Primary and Relay Sets with Grabriel Rules

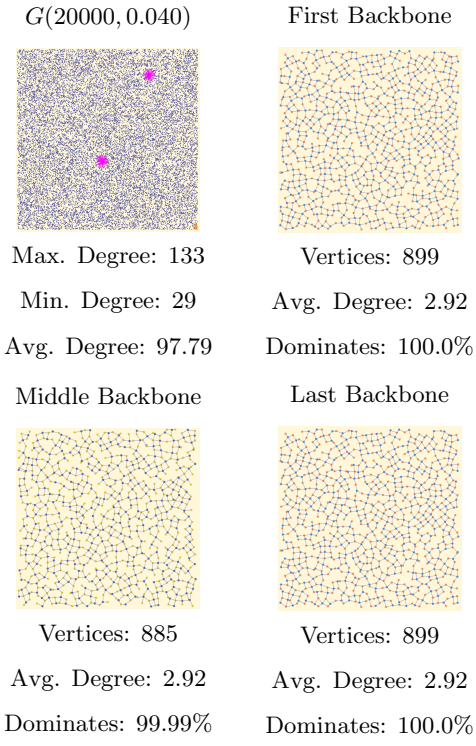


Figure 11: Sample Screenshot results of Unit Square

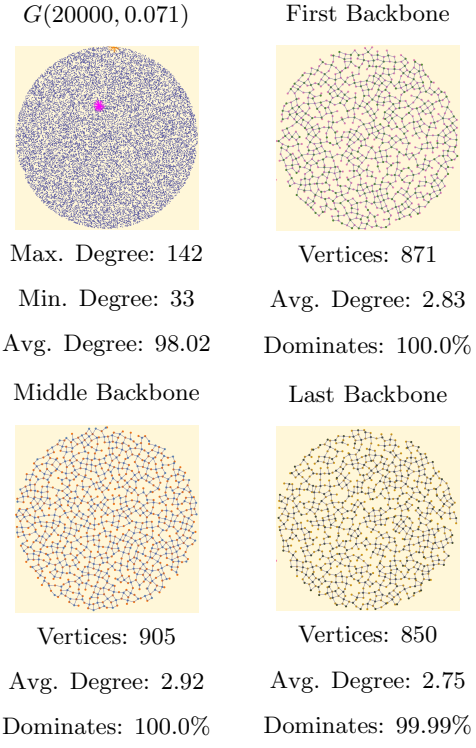


Figure 12: Sample Screenshot results of Unit Disk

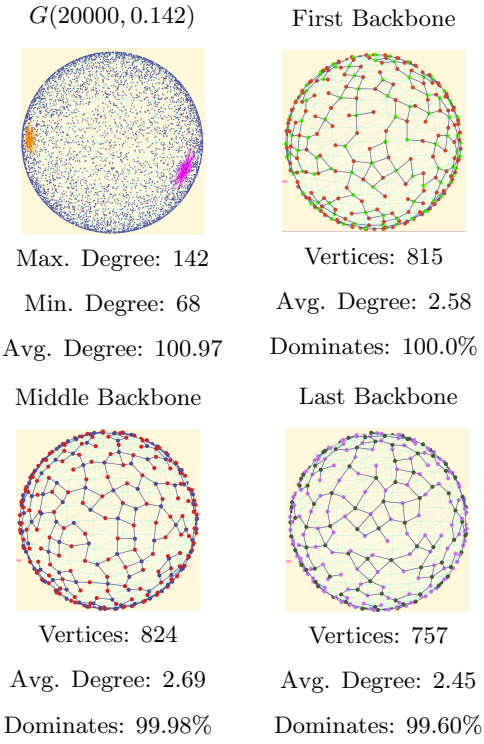


Figure 13: Sample Screenshot results of Unit Sphere

ϵ) dominant bipartite (planar) subgraphs of similar size and structure termed "backbones", the other of comparably small size composed of the "noise" in the random distribution. We implemented a 3D tool for

Table 1: Unit Square Test Benchmarks

N	8000	16000	32000	64000	128000
r	0.057	0.040	0.028	0.020	0.014
Avg.degree	76.52	77.99	78.99	79.62	79.97
Backbones	17	18	18	18	18
Residual vertices	10.16%	8.01%	8.51%	8.45%	8.51%
Avg. backbone size	422.76	817.67	1626.50	3255.28	6505.89
Avg. backbone avg. degree	2.67	2.64	2.66	2.67	2.69
Avg. backbone avg. face size	8.07	8.49	8.32	8.30	8.08
Avg. two-core avg. face size	6.80	6.84	6.77	6.76	6.72
Avg. backbone dominates	99.58%	99.12%	99.05%	99.48%	99.37%

Table 2: Unit Disk Test Benchmarks

N	8000	16000	32000	64000	128000
r	0.101	0.071	0.050	0.036	0.025
Avg.degree	77.30	78.86	79.31	79.85	80.17
Backbones	17	18	18	18	18
Residual vertices	10.29%	8.74%	8.49%	8.12%	8.65%
Avg. backbone size	422.18	811.22	1626.83	3266.94	6496.00
Avg. backbone avg. degree	2.67	2.65	2.66	2.68	2.69
Avg. backbone avg. face size	8.12	8.37	8.34	8.18	8.09
Avg. two-core avg. face size	6.73	6.80	6.75	6.73	6.72
Avg. backbone dominates	99.78%	99.07%	99.36%	99.67%	99.44%

Table 3: Unit Sphere Test Benchmarks

N	8000	16000	32000	64000	128000
r	0.201	0.142	0.100	0.071	0.050
Avg. degree	81.18	81.11	80.97	80.98	81.01
Backbones	18	18	18	18	18
Residual vertices	8.74%	8.56%	8.21%	8.36%	8.29%
Avg. backbone size	405.61	812.78	1631.89	3258.44	6521.83
Avg. backbone avg. degree	2.71	2.71	2.71	2.71	2.71
Avg. backbone face size	7.80	7.82	7.82	7.84	7.85
Avg. two-core face size	6.65	6.60	6.67	6.64	6.64
Avg. backbone dominates	99.80%	99.79%	99.86%	99.82%	99.83%

generating the testbed benchmarks and are surprised at discovering and verified that such backbone partitions exist by employing relatively few dense backbones with little loss to random distribution noise is of interest to the rapidly growing field of wireless sensor networks. We used a two-phase sequential coloring algorithm with combining an old smallest-last coloring alogrithm with a new adaptive coloring algorithm to reach our goal. From Figure 11 to Figure 13 and Table 1 to Table 3 are some example screenshots of our test benchmarks of RGG of 20000 vertices with average degree around 100 and some test data tables of RGG of average degree around 80.

References

[1] David W. Matula and Leland L. Beck, *Smallest-last ordering and clustering and graph coloring algorithms*, Journal of the ACM (JACM), 1983, pp. 417–427.

[2] David W. Matula and Robert R. Sokal, *Properties of Gabriel graphs relevant to geographic variation research and clustering of points in the plane*, Geographical Analysis, 1980, pp. 205–222.

[3] Dhia Mahjoub and David W. Matula, *Employing* $(1-\epsilon)$ *dominating set partitions as backbones in wireless sensor networks*, Proceedings of the 11th Workshop on Algorithm Engineering and Experiments (ALENEX), 2010, pp. 98–111.

[4] Enrico Del Re, Renato Pucci and Luca Simone Ronga, *IEEE802.15.4 Wireless Sensor Network in Mars exploration scenario*, Satellite and Space Communications, (9) 2009, pp. 284–288.

[5] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz and Jonathan Lees, *Deploying a Wireless Sensor Network on an Active Volcano*, IEEE Internet Computing, 2006.

[6] Hichem Kenniche and Vlady Ravelomananana, *Random Geometric Graphs as Model of Wireless Sensor Networks*, Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference, (2) 2010, pp. 103–107.

[7] Honghai Zhang and Jennifer C. Hou, *Maintaining Sensing Coverage and Connectivity in Large Sensor Networks*, Ad Hoc & Sensor Wireless Networks - AHSWN, 2005.

[8] Jiguo Yu, Qingbo Zhan, Dongxiao Yu, Congcong Chen and Guanghui Wang, *Domatic partition in homogeneous wireless sensor networks*, Journal of Network and Computer Applications, (1) 2014, pp. 186–193.

[9] Patricia Morreale, Feng Qi and Paul Croft, *A green wireless sensor network for environmental monitoring and risk identification*, International Journal of Sensor Networks, (6) 2011, pp. 73–82.

[10] Peng-Jun Wan and Chih-Wei Yi, *Coverage by randomly deployed wireless sensor networks*, IEEE Transactions on Information Theory, (6) 2006, pp. 2658–2669.

[11] Prasan Kumar Sahoo, Ming-Jer Chiang and Shih-Lin Wu, *An Efficient Distributed Coverage Hole Detection Protocol for Wireless Sensor Networks*, Sensors. 2016, (3) 2016, pp. 386.

[12] Santosh Kumar, Ten H. Lai and Jzsef Balogh, *On k-coverage in a mostly sleeping sensor network*, Proceedings of the 10th annual international conference on Mobile computing and networking, 9 (2004), pp. 144–158.

[13] Saurav Pandit, Sriram V. Pemmaraju and Kasturi Varadarajan, *Approximation Algorithms for Domatic Partitions of Unit Disk Graphs*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Lecture Notes in Computer Science, 2009, pp. 312.

[14] V. H. MAC DONALD, *Advanced Mobile Phone Service: The Cellular Concept*, The Bell System Technical Journal, (1) 1979, pp. 15–41.

[15] Wint Yi Poe and Jens B. Schmitt, *Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay*, AINTEC '09 Asian Internet Engineering Conference, 2009, pp. 77–84.

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

[17] Zizhen Chen and David W. Matula, *Partitioning Random Geometric Graph into* $(1-\epsilon)$ *Backbones*, Proceedings of The SIAM Workshop on Network Science, 2016.