Distributed Gateway Placement for Cost Minimization in Wireless Mesh Networks

XiaoHua Xu ShaoJie Tang Xufei Mao Xiang-Yang Li
Computer Science Department, Illinois Institute of Technology
Emails: {xxu23,stang7,xmao3}@iit.edu, xli@cs.iit.edu

***Abstract*—We study the problem of *gateway placement for cost minimization* (GPCM) in two-dimensional wireless mesh networks. We are given a set of mesh routers, assume they have identical transmission range r, represented by unit transmission disks around them. A router may be selected as a gateway at certain *placing* cost. A router is served by a gateway if and only if the gateway is within its transmission range.**

The goal of this work is to select a set of mesh routers as gateways to serve the rest routers with minimum overall cost. This problem is NP-hard. To the best of our knowledge, no distributed algorithm with a constant approximation ratio has been given before. When all weights are uniform, the best approximation ratio is 38. We present both centralized and distributed algorithms which can achieve approximation ratios $6 + \epsilon$ and 20 respectively. Our algorithms greatly improve the best approximation ratios.

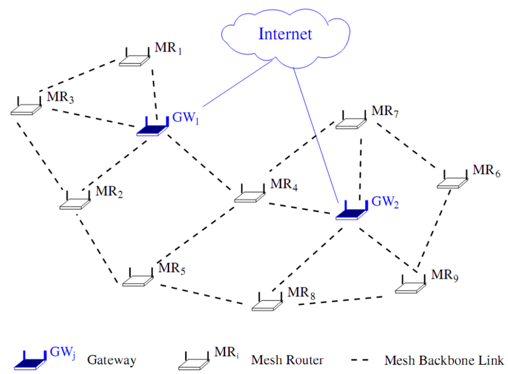

Fig. 1. The network infrastructure of wireless mesh network.

I. Introduction

In recent years, Wireless Mesh Networks (WMNs) [1] attract considerable attentions due to their various potential applications, such as broadband home networking, community and neighborhood networks, and enterprize networking. Many cities and wireless companies around the world have already deployed mesh networks. U.S. military forces are now using wireless mesh networking to connect their computers, mainly ruggedized laptops, in field operations as well. For this application, WMNs can enable troops to know the locations and status of every soldier or marine, and to coordinate their activities without much direction from central command. MWNs have also been used as the last mile solution for extending the Internet connectivity for mobile nodes. For example, in the *one laptop per child* program, the laptops use WMNs to enable students to exchange files and get on the Internet even though they lack wired or cell phone or other physical connections in their area.

WMNs consist of two types of nodes: *mesh routers* and *mesh clients*. Compared with conventional wireless routers, mesh routers may achieve the same coverage with much lower transmission power through multi-hop communications. Among mesh routers, there exist some self-configuring, self-healing links. Mesh routers form an infrastructure for mesh clients. To connect the mesh network (consisting of mesh clients and mesh routers) to the Internet, gateway devices (gateway node) are needed. Usually, in mesh networks some mesh routers (gateway candidates) have the gateway functionality which can provide the connectivity to the Internet. We select a subset of them to function as gateway devices. Note that gateway candidates are different from gateways, a gateway candidate becomes a gateway only if it is selected to function. The common network infrastructure for mesh networks is illustrated in Fig. 1, where dash and solid lines indicate wireless and wired links respectively. We do not include the mesh clients in Fig. 1, as in our work, we mainly focus on the design of the mesh backbone. Hereafter, we will call the ordinary mesh routers as *mesh nodes* or just mesh routers, and call the mesh routers selected as gateway as *gateway nodes* to differ from mesh routers.

The application scenario of this gateway placement problem for a community network is as follows. The mesh nodes are deployed on the roof of houses in a neighborhood, which serve as access points for users inside the homes and along the roads. All these mesh nodes are fixed and form the mesh network. The mesh service provider needs to decide where to place the gateway devices to connect the mesh network to the Internet. Since different gateway placement causes different mesh backbone topology and cost, it is important to find optimal gateway placement to minimize the total cost while ensuring the quality of service, *e.g.*, coverage.

In our paper, we study gateway placement for mesh backbone with minimum cost. Given the mesh backbone consisting of a set of mesh node (some of them are gateway candidates, we assume all mesh nodes have identical transmission range r and each gateway candidate is associated with a placing cost (weight). A mesh node is served by a gateway if and only if the gateway is within its transmission range. We want to select a subset of gateway candidates to function as gateway, so as to serve the mesh nodes with the overall placing cost minimized. This problem is NP-hard. To the best of our knowledge, no

1063-6927/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDCS.2010.51

distributed algorithm with a constant approximation ratio has been given before. When all weights are uniform, the best approximation ratio is 38. We first introduce a centralized algorithm which can achieve approximation ratios $6+\epsilon$. Then we propose a distributed algorithm with approximation ratio 20.

The rest of the paper is organized as follows. Section II formulates the GPCM problem. Section III presents a centralized algorithm for the GPCM problem with the approximation ratio $6+\epsilon$. Section IV presents our distributed algorithm the GPCM problem with the approximation ratio 20. Section V outlines the related work. Finally, Section VI concludes the paper.

II. Network Model and Problem Formulation

A mesh network is modeled by a undirected graph $G = (V, E)$, where $V = \mathcal{P} = \{v_1, \cdots, v_n\}$ is the set of n mesh nodes and E is the set of possible communication links. Every node v_i has identical transmission range r, there is an edge between any two nodes if and only if they are within transmission range of each other, *e.g.*, the euclidian distance is no greater than r. By a proper scaling, we can assume that $r = 1$.

Among the set V of all wireless mesh nodes, some of them (gateway candidates) have gateway functionality and can provide the connectivity to the Internet. Let $\mathcal{D} = \{d_1, d_2, \cdots, d_n\}$ be the set of m ($m \leq n$) gateway candidates, where d_i is actually node v_{n+i-m}, for $1 \leq i \leq m$. All other wireless nodes v_i (for $1 \leq i \leq n-m$) $\in V \backslash \mathcal{D}$ are ordinary mesh nodes. Each ordinary mesh node u will aggregate the traffic from all its users (or mesh clients) and then route them to the Internet through a real gateway node it is served by. We further assume that each gateway candidate d_i is associated with a placing cost w_i, at which it can be selected to function as a real gateway node.

The goal of this work is to select a set of gateway candidates $D \subseteq \mathcal{D}$ as real gateway nodes to ensure that 1) each mesh node can be served by at least one gateway, *e.g.*, there exists at least one gateway within its transmission range, and 2) the overall cost $\sum_{i \in D} w_i$ is minimized. For simplicity, we can assume the transmission range $r = 1$ by proper scaling.

Since our problem is very similar to the *minimum weighted dominating set problem* (MWDS), we will borrow some idea from the existing solutions on MWDS problem to design a centralized algorithm for our problem with constant approximation.

The main contribution of this work is that, we are the first to propose a distributed algorithm with constant approximation for the gateway placement problem. To illustrate our main idea in an easy way, we first introduce a centralized algorithm with $6+\epsilon$ approximation. And we further extend it in a distributed manner with 20 approximation.

III. Centralized Algorithm

In this section, we study the gateway placement problem in a centralized manner. We present an algorithm with the approximation ratio $6+\epsilon$ for GPCM based on an existing algorithm in [9] for MWDS problem.

We employ double partition and divide-and-conquer techniques, similar to [9]. Double partition means that we first partition the plane into large blocks, each block is a square with a side-length $t\mu$, where $\mu = \frac{\sqrt{2}}{2}$ (to ensure the diameter of the square is 1, thus any gateway inside the square can serve all mesh nodes inside the square) and t is a large integer constant to be used for shifting strategy. Then we partition each large block into t^2 small squares with the side-length μ. The process of double partition is illustrated in Figure 2.

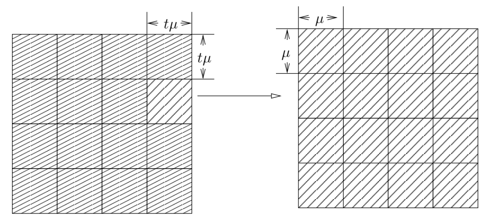

Fig. 2. **Double partition:** Partition plane into blocks and further partition each block into some squares.

After double partition, the algorithm for solving the GPCM problem can be divided into the following phases:

1) Solve GPCM in a $t\mu \times t\mu$ block. This means, for each block, an instance of the GPCM problem can be defined as: to select a subset of gateway candidates to function, so as to serve all the mesh nodes lying inside the block.
2) Use the union of solutions for all instances of GPCM in all blocks obtained in the Step 1) as a solution in the plane.
3) Use shifting strategy to get a set of solutions in the plane similarly. Here the shifting strategy (The details can be found in [9]) is to try all possible t ways along the diagonal direction to partition the plane into blocks of size $t\mu \times t\mu$. For each way of partition, we perform Step 1) and 2) to find a solution in the plane.
4) Selection the one among all t solutions found by the shifting strategy with minimum-weight as the final solution in the plane and return it.

The details for solving GPCM in the plane are shown in Algorithm 1.

Finally, we show the approximation ratio of Algorithm 1 for GPCM in a plane.

Theorem 1: For any constant ϵ, by setting $t = O(1/\epsilon)$, Algorithm 1 always outputs a set of gateway nodes with weight bounded by $(6+\epsilon) \cdot w(OPT)$, where OPT is the optimum solution and $w(OPT)$ is the weight of OPT.

Proof: The proof follows Theorem 1 in [9]. By lemma 6 in appendix (Subsection A), every gateway node in the optimal solution OPT can be counted at most 6 times for solving GPCM in a block. However it may be counted more if it is located in the boundary region of a block. When we shift the whole block many times, for any gateway node in OPT, it would be counted at most 6 times in most cases. Since we return the one with minimum-weight as the final solution, we can achieve a solution with weight bounded by $(6+\epsilon) \cdot w(OPT)$ for any small constant $\epsilon > 0$. ■

Algorithm 1 Centralized Algorithm for GPCM ([9])

Input: a set of mesh nodes $\mathcal{P}$ and a set of gateway candidates $\mathcal{D}$, a weight function w on $\mathcal{D}$

Output: a solution for GPCM which is a subset of nodes from $\mathcal{D}$ selected as gateways.

1: (Double Partition) Partition the whole plane into blocks of size $t\mu \times t\mu$, then partition each block into squares with size $\mu \times \mu$, where $\mu = \frac{\sqrt{2}}{2}$;
2: Find a 6-approximation solution of GPCM for each block that contains mesh nodes to be served and union the solutions together to get a solution for the whole plane. See appendix (Subsection A) for our approach solving GPCM in a block.
3: Move each block one square along its diagonal direction;
4: Repeat Step 2 for this new partition to update the solution if any better solution is found;
5: Repeat Step 3 for t times, and output the final solution.

IV. Distributed Algorithm

In this section, we propose our distributed algorithm with approximation ratio 20 for the GPCM problem.

Then we solve each sub-problem distributively (locally) with a 2-approximation solution. By combining all solutions together, we can achieve a 20-approximation solution for MWDC which implies a solution for the original GPCM problem immediately.

A. Our algorithm

Our algorithm employs the parameter $\mu = \frac{\sqrt{2}}{2}$ as well. We partition the plane into $\mu \times 2\mu$ rectangles (Figure 3). The rectangle S_{ij}, for $i, j \in Z$, contains all nodes (x, y) with

1) $j\mu \leq y \leq (j+1)\mu$ and $2i\mu \leq x \leq 2(i+1)\mu$ if $j = 2k$;
2) $j\mu \leq y \leq (j+1)\mu$ and $(2i+1)\mu \leq x \leq (2i+3)\mu$ if $j = 2k+1$.

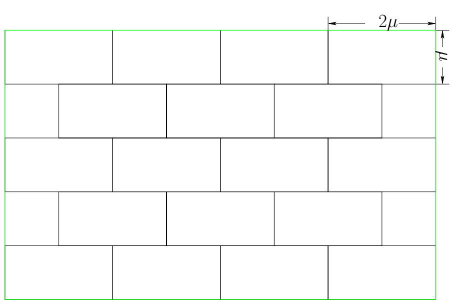

Fig. 3. Partition the plane into $\mu \times 2\mu$ rectangles.

Given an instance of MWDC, let $\mathcal{P}$ be the set of all nodes to be covered, and $\mathcal{D}$ be the set of all weighted disks. For a $\mu \times 2\mu$ rectangle S_{ij} that contains at least a node to cover, let P_{ij} denote the subset of nodes in $\mathcal{P}$ which are located inside S_{ij} and D_{ij} denote the subset of disks in $\mathcal{D}$ that covers at least one node in P_{ij}. Then we consider the following subproblem: find a minimum-weight subset of disks in D_{ij} that covers all nodes in P_{ij}. We will present a 2-approximation solution (note as U_{ij}) for the subproblem in Section IV-C. For each rectangle S_{ij}, we can define the subproblem and find a corresponding solution U_{ij} with constant approximation. We use the union of solutions U_{ij} for all subproblems as the global solution. The details of our distributed algorithm for MWDC in the plane in shown in Algorithm 2. Note that in our algorithm, the nodes in $\mathcal{P}$ are the entities to do computing.

Algorithm 2 Distributed Algorithm for MWDC

Input: a set of nodes $\mathcal{P}$ and a set of weighted disks $\mathcal{D}$

Output: a solution for MWDC (A subset of disks)

1: Each node $p_i \in \mathcal{P}$ broadcasts to all its neighboring nodes within two-hops: (1) ID; (2) which rectangles it lies in. (Here we assume the nodes are wireless devices with communication and computing ability).
2: For each rectangle S_{ij}, the node with the largest ID elects itself as the leader and notify to all nodes in the rectangle.
3: Each node $p_i \in \mathcal{P}$ sends to the leader of the rectangle a message containing the information about where it lies: (1) ID; (2) all disks that covers p_i.
4: For each rectangle S_{ij}, after receiving all messages, the leader determines all nodes lying in S_{ij} (assume they form a set P_{ij}); and all disks that covers at least one node in P_{ij} (assume they form a set D_{ij}).
5: For each rectangle S_{ij} that contains nodes, the leader find a 2-approximation solution of MWDC, here the input is disk set D_{ij} and node set P_{ij}. See Section IV-C for our approach solving MWDC in a rectangle.
6: Output the final solution as the union of solutions found in Step 5. Note that, if a disk has appeared in the union of solutions for multiple times, we only keep one in the final solution by marking method.

It is clear that the global solution is a feasible solution. Next we show that the global solution is within a constant approximation of the optimum solution.

B. Performance analysis

In this section, we analyze the approximation ratio of our method in Section IV-A. As in Algorithm 2, we first reduce the problem of MWDC into a set of sub-problems in a $\mu \times 2\mu$ rectangle, we prove that we only lose an approximation ratio of 10 in the reduction process. Then for each sub-problem, we can find a 2-approximation solution as shown in Section IV-C. Combining the two parts, we can prove that our distributed algorithm can achieve an approximation ratio of 20.

Lemma 2: Each disk can intersect at most 10 $\mu \times 2\mu$ rectangles. Here a disk intersects a rectangle *iff* there exists an common area between the disk and the rectangle. (See Subsection C in the appendix for proof)

Theorem 3: Algorithm 2 can find a 20-approximation solution for the MWDC problem.

Proof: By Algorithm 2, we can find a disk set U whose total weight is at most $\sum_{i,j} w(U_{ij})$. The summation is over all $\mu \times 2\mu$ rectangles that contain at least one node to cover.

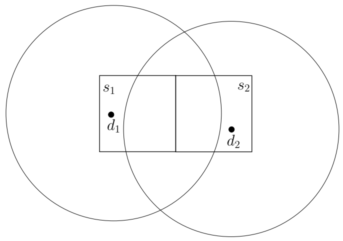

Fig. 4. Case 1 of the OPT covering pattern.

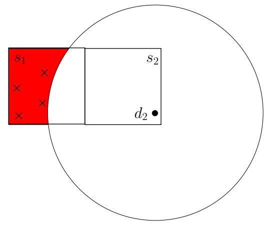

Fig. 5. Case 2 of the OPT covering pattern.

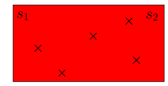

Fig. 6. Case 3 of the OPT covering pattern.

Let OPT_{ij} denote an optimal solution to the subproblem for S_{ij}. We can find a 2-approximation algorithm for each subproblem as proved in Section IV-C, we have $w(U_{ij}) \leq 2 \cdot w(OPT_{ij})$.

Let OPT denote the global optimal solution. Note that $OPT \bigcap D_{ij}$ is a feasible solution to subproblem S_{ij} and OPT_{ij} is an optimal solution. Then,

$$w(OPT_{ij}) \leq w(OPT \bigcap D_{ij})$$

Therefore, we get,

$$w(U) \leq \sum_{i,j} w(U_{ij}) \leq 2 \sum_{i,j} w(OPT_{ij}) \leq 2 \sum_{i,j} w(OPT \bigcap D_{ij})$$

The sum $\sum_{i,j} w(OPT \bigcap D_{ij})$ adds the weight of solutions for all rectangles S_{ij} that contain at least one node in $\mathcal{P}$. Note that a disk d in OPT can be in $OPT \bigcap D_{ij}$ only if it covers a node in P_{ij} and thus intersects the rectangle S_{ij}. By Lemma 2, a disk can intersect at most 10 $\mu \times 2\mu$ rectangles. This means that each disk in OPT contributes its weight to $OPT \bigcap D_{ij}$ at most 10 times. We have,

$$\sum_{i,j} w(OPT \bigcap D_{ij}) \leq 10 \cdot w(OPT).$$

Therefore, $w(U) \leq 20 \cdot w(OPT)$. ■

C. 2-Approximation solution for MWDC in a $\mu \times 2\mu$ rectangle

In this section, we will present a 2-approximation solution for a subproblem of MWDC in a $\mu \times 2\mu$ rectangle.

We divide the $\mu \times 2\mu$ rectangle into 2 squares $S = \{s_1, s_2\}$ (each with side length μ). Let OPT denote a set of disks constituting an optimal solution for the problem. We guess the *covering pattern* of OPT and use the dynamic programming technique to find a 2-approximation solution. If we define that OPT covers a square if OPT contains a disk with center inside the square. Then the covering pattern refers to whether OPT covers any of the two squares. Clearly, for each square, OPT either covers the square or not. If OPT covers a square, then all the nodes inside the square can be covered by an covers disk in OPT. Since we focus only two squares $S = \{s_1, s_2\}$, there are totally at most 2^2 cases for the covering pattern. We consider each case (note that Case 2 include 2 cases) separately as follows:

1) (Figure 4) Both squares in $\{s_1, s_2\}$ contain a disk in OPT. Since the diameter of the squares is one which equals to the radius of disks, one disk with center in a square is enough to cover all the nodes in the square. Thus there is no need to add additional disks. Thus the OPT has only 2 disks. We can get an optimal solution by guessing which 2 disks are in polynomial time $O(n^2)$ for this case.
2) (Figure 5) One square in $\{s_1, s_2\}$ contains disks in OPT. Assume s_2 contains at least one disk d_2. Then d_2 can cover all the nodes in square s_2 and may cover some consecutive area of its neighboring square s_2. We delete the nodes that have already been covered by disks d_2. Since s_1 does not contain disks in it, we need to use all disks outside of square s_1 to cover the remaining nodes in the red region. Ambühl *et al.* [3] showed a 2-approximation to cover nodes in a square with disks all outside of the square. Thus we can get 2-approximation solution for this case.
3) (Figure 6) Neither square in $\{s_1, s_2\}$ contains disks in OPT. This means that two consecutive squares does not contain disk. In this case, we need to use all disks outside of the rectangle $s_1 \cup s_2$ to cover the nodes in the red region. In appendix (Subsection B), we will show a 2-approximation for cover nodes in a rectangle with disks all outside of the rectangle. Thus we can get 2-approximation solution for this case.

Since for each of all three complementary cases, we can find 2-approximation solution, we can achieve an approximation ratio of 2 for MWDC in a $\mu \times 2\mu$ rectangle.

V. Related Work

As we know, GPCM is a essentially a coverage problem which has been extensively studied recently. To evaluate the quality of coverage of wireless networks, Meguerdichian *et al.* [12] formulated the 1-coverage problem under two extreme cases: the best case coverage (maximum support) problem and the worst case coverage (minimum breach) problem. They observed that an optimal solution for the maximum support problem is a path which lies along the edges of the Delaunay triangulation [11] [16] and an optimal solution for the minimum breach problem is a path which lies along the edges of the Voronoi diagram [11] [16]. They further proposed centralized optimal algorithms for both problems. Later, Mehta *et al.* [13] improved these algorithms and made them more computational efficient.

Recently, some work aimed at solving the 1-coverage problem formulated in [12] in a distributed manner. Li *et.al* [19] showed that the maximum support path can be constructed by using edges that belong to the relative neighborhood graph (RNG) of the sensor set. They attempted to address best case 1-coverage problem in distributed manner. This is an improvement since the RNG is a subgraph of the Delaunay triangulation and can be constructed locally. On the other side, Meguerdichian *et.al* [12] implied that a variation of the localized exposure algorithm presented in [16] can be used to solve the worst case coverage problem locally. Another localized algorithm with more practical assumptions was proposed by Huang *et al.* [8].

For the general coverage problem, Huang *et al.* [8] studied the problem of determining if the area is sufficiently k-covered, *i.e.*, every point in the target area is covered by at least k sensors. They formulated the problem as a decision problem and proposed a polynomial algorithm which can be easily translated to distributed protocols. In [4], Huang *et al.* further extended this problem to three-dimensional sensor networks and proposed a solution The connected k-coverage problem was addressed in [20] in which Zhou *et al.* studied the problem of selecting a minimum set of sensors which are connected and each point in a target region is covered by at least k distinct sensors. They gave both a centralized greedy algorithm and a distributed algorithm for this problem and showed that their centralized greedy algorithm is near-optimal. Xing *et al.* [7] explored the problem concerning energy conservation while maintaining both desired coverage degree and connectivity. They studied the integrated work between the coverage degree and the connectivity and proposed a flexible coverage configure protocol.

Some studies focused on the relationship between the coverage degree k, the number of sensors n and the sensing radius r. Kumar *et al.* [17] considered the problem of determining the appropriate number of sensors that are enough to provide k-coverage of a region when sensors are allowed to sleep during most of their lifetime. In [18], Wan *et al.* analyzed the probability of the k-coverage when the sensing radius or the number of sensors changes while taking the boundary effect into account.

Since the coverage problem can be reduced to disk cover problem, we briefly review the recent work [2], [6] about the disk cover problem in which the authors want to deploy some disks (with same radius or not) at some locations on the given area such that all points in the given point set are fully 1-covered. Calinescu *et al.* [5] proposed the first constant factor algorithm with approximation ratio 108. Narayanappa and Vojtechovsky [14] improved this constant to 72. The best result so far is achieved by Carmi *et al.* [6] with an approximation ratio 38.

VI. CONCLUSION

In this paper, we study the problem of gateway placement with minimum cost in wireless mesh networks. We propose a distributed method which can achieve a 20-approximation of the optimum. To the best of our knowledge, this is the first work to give a distributed algorithm with constant approximation for this problem. We also design a centralized algorithm with $6+\epsilon$ approximation which greatly improves the previous results with approximation ratio 38.

As a future research direction, we would like to know whether there is a PTAS for this problem, *i.e.*, whether it is possible to design a polynomial time algorithm such that, given any constant $\epsilon > 0$, we can find a solution whose total weight is at most $1+\epsilon$ times of the optimum. Also, we try to find a optimum gateway placement by taking the connection cost into account, in particular, if each link is associated with certain connection cost, then how to assign each router to different gateway becomes a very interesting while interesting problem.

VII. ACKNOWLEDGEMENT

The research of Xiang-Yang Li is partially supported by NSF CNS-0832120, National Natural Science Foundation of China under Grant No. 60828003, program for Zhejiang Provincial Key Innovative Research Team, program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program), National Basic Research Program of China (973 Program) under grant No. 2010CB328100, and is partially funded by Tsinghua National Laboratory for Information Science and TechnologyTNList).

REFERENCES

[1] AKYILDIZ, I., WANG, X., AND WANG, W. Wireless mesh networks: a survey. *Computer Networks 47*, 4 (2005), 445–487.

[2] ALT, H., ARKIN, E., BRÖNNIMANN, H., ERICKSON, J., FEKETE, S., KNAUER, C., LENCHNER, J., MITCHELL, J., AND WHITTLESEY, K. Minimum-cost coverage of point sets by disks. In *Proceedings of the twenty-second annual symposium on Computational geometry* (2006), ACM New York, NY, USA, pp. 449–458.

[3] AMBUHL, C., ERLEBACH, T., MIHAL AK, M., AND NUNKESSER, M. Constant-Factor Approximation for Minimum-Weight (Connected) Dominating Sets in Unit Disk Graphs. *Lecture Notes in Computer Science 4110* (2006), 3.

[4] C. HUANG, Y. C. T., AND LO, L. The coverage problem in threedimensional wireless sensor networks,. In *Proceedings of IEEE GLOBECOM* (2004).

[5] CĂLINESCU, G., MĂNDOIU, I., WAN, P., AND ZELIKOVSKY, A. Selecting forwarding neighbors in wireless ad hoc networks. *Mobile Networks and Applications 9*, 2 (2004), 101–111.

[6] CARMI, P., KATZ, M., AND LEV-TOV, N. Covering points by unit disks of fixed location. *Lecture Notes in Computer Science 4835* (2007), 644.

[7] G. XING, X. WANG, Y. Z. C. L. R. P., AND GILL, C. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks, vol. 1, pp. 36-72* (2005).

[8] HUANG, C., AND TSENG, Y. The coverage problem in a wireless sensor network. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)* (2003).

[9] HUANG, Y., GAO, X., ZHANG, Z., AND WU, W. A better constant-factor approximation for weighted dominating set in unit disk graph. *J. Comb. Optim 1382* (2008), 6905.

[10] JAIN, K., AND VAZIRANI, V. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM (JACM) 48*, 2 (2001), 274–296.

[11] M. DE BERG, M. VAN KREVELD, M. O., AND SCHWARZKOPF, O. Computational geometry: Algorithms and applications. *Spinger, New York* (1997).

[12] MEGUERDICHIAN, S., POTKONJAK, F. K. M., AND SRIVASTAVA, M. B. Coverage problems in wireless ad-hoc sensor networks. In *Proc. INFOCOM 139-150* (2001).

[13] MEHTA, D.P. LOPEZ, M., AND LIN, L. Optimal coverage paths in ad-hoc sensor networks. In *Proceedings of IEEE ICC 03, 507-511* (2003).

[14] NARAYANAPPA, S., AND VOJTECHOVSKY, P. An improved approximation factor for the unit disk covering problem. In *Proceedings of the 18th Canadian Conference on Computational Geometry (CCCG)* (2006).

[15] PANDIT, S., AND PEMMARAJU, S. Finding facilities fast. In *Proceedings of the 10th International Conference on Distributed Computing and Networks* (2009), Springer.

[16] ROURKE, J. O. Computational geometry in c. *Cambridge University Press, New York* (1998).

[17] S. KUMAR, T. L., AND BARLOGH, J. On k-coverage in a mostly sleeping sensor network. In *Proceedings of MobiCom pp. 144-158.* (2004).
[18] WAN, P., AND YI, C. Coverage by randomly deployed wireless sensor networks,. *IEEE Transactions on Information Theory, vol. 52, pp. 2658-2669* (2006).
[19] X. LI, P. W., AND FRIEDER, O. Coverage in wireless ad-hoc sensor networks. *IEEE Transaction on Computers, vol. 52, no. 6, pp. 753-763* (Jun. 2003).
[20] Z. ZHOU, S. D., AND GUPTA, H. Connected k-coverage problem in sensor networks. In *Proceedings of ICCCN* (2004).

APPENDIX

A. 6-*approximation for GPCM in a block*

In this subsection, we present our algorithm for GPCM in a $t\mu \times t\mu$ block B. As we can map each gateway candidate in $\mathcal{D}$ as a unit disk with the same weight as that of the gateway candidate, each mesh node $\mathcal{P}$ as a node in the plane, our GPCM problem is reduced to a weighted version of *Discrete Unit Disk Cover (DUDC)* as defined in [6]: to cover all nodes in $\mathcal{D}$ using unit disks in $\mathcal{D}$ with minimum weight. Then we present a solution for solving weighted DUDC in a block with 6-approximation.

We first introduce some terms and notations. Since block B consists of t^2 squares of size $\mu \times \mu$, we denote them as $S_{ij}, i, j \in [t]$. S_{ij} is the square in the i-th order from left to right and in the j-th order from up to down. All squares S_{kj} together form a horizontal strip (note as $s_k^x, j \in [t]$). Thus, block B contains t horizontal strips, $s_1^x, \cdots, s_m^x$. Similarly, block B contains t vertical strips, $s_1^y, \cdots, s_m^y$, where s_k^y is formed by combination of all squares $S_{ik}, i \in [t]$. Let D_{ij} and P_{ij} denote the set of disks and points lying in S_{ij} respectively.

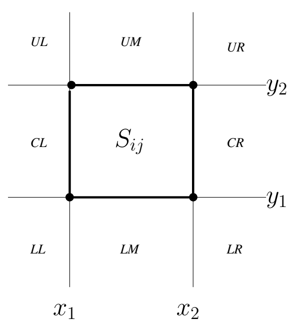

Fig. 7. Partition of outside of a $\mu \times \mu$ square into 8 regions.

For each square S_{ij}, we divide its outside into eight regions $UL, UM, UR, CL, CR, LL, LM, LR$ as shown in Fig. 7. Denote by $Left = UL \cup CL \cup LL$, $Right = UR \cup CR \cup LR$, $Up = UL \cup UM \cup UR$, $Down = LL \cup LM \cup LR$. Assume the four lines forming S_{ij} are $x = x_1, x = x_2, y = y_1, y = y_2$.

Then we briefly describe the idea for solving weighted DUDC in a block.

(1) Guessing the covering pattern. Assume the optimum solution is OPT. For each square S_{ij}, we have the following two complementary cases:
- $d \in OPT \cap D_{ij} \neq \emptyset$. Since the disk radius is one and the diameter of every square is one, any disk d from $OPT \cap D_{ij}$ can cover S_{ij} entirely. Thus d can cover all points in P_{ij}.
- $OPT \cap D_{ij} = \emptyset$. In this case, P_{ij} are covered by disks outside of the square S_{ij}. By Lemma 5, we can use up to 4 points to separate points in P_{ij} into two groups, one group can be covered by disks only from the Up and Down region of the square, and the other can be covered by disks only from the Left and Right region of the square S_{ij}.

Thus, we can guess the covering pattern of OPT for each square S_{ij} by enumeration of all possibilities.

(2) Solving weighted DUDC over strips. Once we guess a pattern, we can decompose the problem into problem in strips. We solve weighted DUDC for t horizontal strips s_j^x. Similarly, We solve weighted DUDC for t vertical strips s_j^y. We combine the $2m$ solutions and use $OPT \cap D_{ij}$ as the solution for this pattern. We then output the minimum solution over all possible enumerated covering patterns.

Lemma 4: ([9]) Suppose $p \in P_{ij}$ is a point inside S_{ij} which can be covered by a disk $d \in LM$. We draw two lines p_l and p_r , which intersect $y = y_1$ by angle $\pi/4$ and $3\pi/4$. Then the shadow P_{LM} surrounded by $x = x_1, x = x_2, y = y_1$, p_l and p_r can also be covered by d. Similar results can be hold for shadow P_{UM}, P_{CL} and P_{CR}, which can be defined with a rotation.

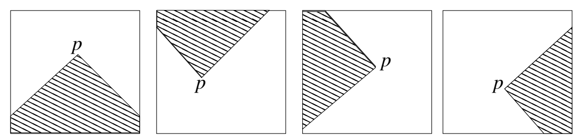

Fig. 8. P_{LM}, P_{UM}, P_{CL} and P_{CR}

Then, we give the definition of sandglass and a lemma which can be used to separate P_{ij} into two groups, with one can be covered by disks from $Up \cup Down$ and the other by disks from $Left \cup Right$.

Definition 1: ([9] Sandglass) If D is a disk set covering P_{ij} and $D \cap S_{ij} = \emptyset$, then there must exist a subset $P_M \subset P_{ij}$ which can **only** be covered by disks from UM and LM (we can set $P_M = \emptyset$. if there is no such points). Select $P_{LM} \subset P_M$, the disks that can be covered by disks from LM, draw p_l and p_r line for each $p \in P_{LM}$. Select the leftmost p_l and rightmost p_r and form a shadow. Symmetrically, choose P_{UM} and form a shadow. The union of the two shadows form a sandglass region $Sand_{ij}$ of S_{ij}.

Lemma 5: ([9]) Suppose D is a disk set covering P_{ij}, and $Sand_{ij}$ are chosen as in Definition 1. Then any points in $Sand_{ij}$ can be covered by disks only from neighbor region $Up \cup Down$, and any point from $S_{ij} \setminus Sand_{ij}$ can be covered by disks only from neighbor region $Left \cup Right$.

Then, we give Algorithm 3 for solving weighted DUDC in a block.

Lemma 6: Algorithm 3 can find a 6-approximation solution for weighted DUDC in a $t\mu \times t\mu$ block .

Proof: Since OPT is a feasible solution, for any square in the block, it is either covered by a disk from OPT inside this square, or covered by some disks from OPT outside of the square. So during the enumeration process in Algorithm

Algorithm 3 DUDC in a block ([9])

Input: a set of points $\mathcal{P}$ in the block and $\mathcal{D}$ covering $\mathcal{P}$;
Output: A solution for DUDC in the block.

1: For each S_{ij}, choose its sandglass or select a disk $u \in D_{ij}$ inside the square;
2: If for a square S_{ij}, a disk u is chosen, then remove all points in P_{ij}. We also remove all other points outside of square S_{ij} covered by u;
3: For each horizontal strip s_i^x, calculate an optimum DUDC for the remaining points in the sandglass of s_i^x.
4: For each vertical strip s_j^y, calculate optimum DUDC for the union of points in the sandglass of s_j^y.
5: Enumerates all possible covering patterns and takes the one with minimum weight.

3, once the covering pattern is guessed correctly, for any disk from OPT inside a square which is used to cover this square, it is selected and then deleted by the algorithm in step 1, hence is only used once.

Consider when we calculating DUDC for t horizontal strips, it is used at most 3 times. For the horizontal strips, the analysis is the same. By adding the horizontal and vertical strips up, for any disk, it could be counted at most 6 times totally.

The two solutions together have weight no more than $6 \cdot OPT$, so Algorithm 3 gives a solution with weight no more than $6 \cdot OPT$ when it guess the pattern correctly. Since the algorithm enumerates all possible covering patterns, and takes the minimum solution, Algorithm 3 can output a solution with weight at most $6 \cdot OPT$. ■

B. 2-approximation for MWDC in a rectangle with all disks outside of the rectangle

We first introduce the concept of *upper-active* and *lower-active*.

Definition 2: For a set of disks D with centers in the lower half-plane of line l, we say that a disk u is upper-active at x_p if its uppest intersection point with vertical line $x = x_p$ has the largest y-coordinate among all upper intersection points of disks from D with that line $x = x_p$. We say a disk is upper-active at node p if it is upper-active at $x = x_p$ where x_p is the x-coordinate of node p.

Definition 3: For a set of disks D with centers in the upper half-plane of line l, we say that a disk u is lower-active at x_p if its lower intersection point with vertical line $x = x_p$ has the smallest y-coordinate among all lower intersection points of disks from D with that line $x = x_p$. We say a disk is lower-active at node p if it is lower-active at $x = x_p$ where x_p is the x-coordinate of node p.

For upper-active, we have the following property.

Lemma 7: Consider two disks $\{d_1, d_2\}$ under the edge l, if disk d_1 is upper-active at x_1 and disk d_2 is upper-active at x_2, then $O_1 < O_2$ *iff* $x_1 < x_2$ (Figure 9).

Proof: Assume A_0 is the intersection point of line l and vertical line $x = x_1$, A_1 is the uppest intersection point of disk d_1 and $x = x_1$, and A_2 is the uppest intersection point of disk d_2 and $x = x_1$. Similarly, we define the points B_0, B_1, B_2. Given the assumption that disk d_1 is upper-active at x_1 and disk d_2 is upper-active at x_2, we have: A_1 lies above A_2 while B_2 lies above B_1. By geometrical property, O_1 lies on the perpendicular bisector of A_1B_1 and O_2 lies on the perpendicular bisector of A_2B_2. Both $\triangle O_1A_1B_1$ and $\triangle O_2A_2B_2$ are isosceles triangles. We can move $\triangle O_1A_1B_1$ down along the line A_1A_0 such that A_1 is coincident with A_2. During the moving processing, only y-coordinates of $\{O_1, A_1, B_1\}$ varied by the moving distance vertically, their x-coordinates is unchanged. If $x_1 < x_2$, then A_i lies on the left of B_i ($i = 0, 1, 2$), then O_1 lies to the left of O_2 and vice versa. Thus the proof is done. ■

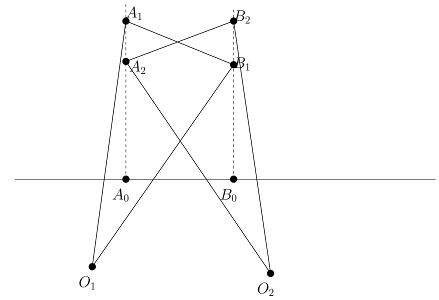

Fig. 9. The property of upper-active.

As a corollary of Lemma 7, assume disk d_i is upper-active at a node with x-coordinate x_i, and disk d_j is upper-active at another node with x-coordinate x_j , if $x_i < x_j$, then disk d_i lies to the left of disk d_j. Thus if we sort all nodes above line l in increasing order of x-coordinate, every disk can only be active for some consecutive nodes. Thus, we have the following lemma.

Lemma 8: Consider a set of disks under a horizontal line l and a set of nodes above line l, each disk can only be upper-active for some consecutive nodes.

Proof: We prove by contradiction. Assume there exists a disk d_1 which is not upper-active for some consecutive nodes. Then there must exist three nodes with x-coordinates x_1, x_2, x_3 respectively and $x_1 < x_2 < x_3$. At the same time, d_1 is upper-active at x_1 and x_3, while another disk d_2 is upper-active at x_2. By Lemma 7, d_2's center should lies both to the right and left of d_1's center. This causes contradiction. Thus the proof is done. ■

Figure 10 serves as an illustration for Lemma 8. In this figure, d_1 is upper-active for all nodes below the red section of line, d_2 is upper-active for all nodes below the green section of line, d_3 is upper-active for all nodes below the brown section of line, d_4 is upper-active for all nodes below the blue section of line.

Now let us return to our problem: to cover all the nodes in a $\mu \times 2\mu$ rectangle $ABCD$ with all disks outside the rectangle which cover at least one node in the rectangle. We divide the outside of rectangle $ABCD$ into 10 regions $UL, UM_1, UM_2, UR, CL, CR, LL, LM_1, LM_2, LR$ as shown in Figure 12. First we only consider all the disks that locate below line AB in OPT (which contains

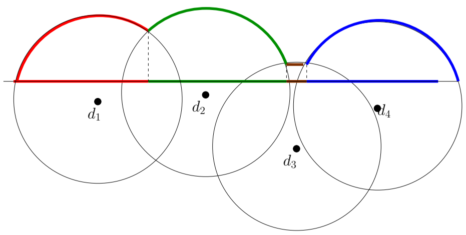

Fig. 10. An example that each disk is upper-active for a set of consecutive nodes.

LL, LM_1, LM_2, LR), order them by their x-coordinates, assume the result is $D = \{d_1, d_2, \cdots, d_n\} \subseteq OPT$. By the optimum of OPT, every disk in D must be active at least in one node from P. For each node, if a disk d is active (either upper-active or lower-active) at the node, then we call the disk d as an active disk for the node. Otherwise, we can delete the redundant disks which are not upper-active at any node from P and get a solution better than OPT which causes contradiction. Thus, we have the following lemma.

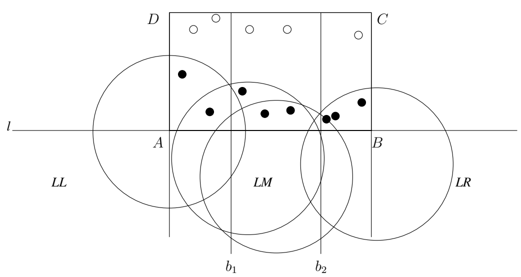

Fig. 11. A set of nodes in a rectangle covered by disks under the rectangle.

Lemma 9: Consider a set of nodes $\mathcal{P}$ located inside a rectangle $ABCD$ and a set of disks located below edge AB, there exist two vertical line b_1, b_2 such that in optimal solution, all nodes in $\mathcal{P}$ lying to the left of line b_1 have active disks in LL, all nodes in $\mathcal{P}$ lying to the right of line b_2 have active disks in LR, and all the nodes lying between line b_1 and b_2 have active disks in LM. Here we say a disk is in LL if its center locates in LL. By similar analysis, we get every sub-rectangle's active disk region.

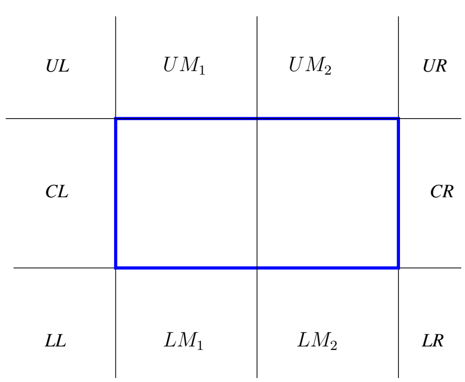

Fig. 12. Partition of outside of 1×2 rectangle into 10 regions.

Based on the lemma, we can partition this $\mu \times 2\mu$ rectangle into 25 small sub-rectangles such that the subset of nodes in $\mathcal{P}$ in each sub-rectangle can be covered by at most two parts in set $\{UL, UM, UR, CL, CR, LL, LM, LR\}$

Take one case for example, sub-rectangle 1 locates on the left of u_1 and b_1, thus it can not be have active disks in UM, UR and LM, LR. Also, sub-rectangle 1 locates on the upper side of both line l_1 and r_1, thus it can not have active disks in CL, LL and CR, LR in OPT. To sum up, the nodes in sub-rectangle 1 can be and only have active disks in UL.

By similar analysis, we can get every sub-rectangle's active region set.

$1 : \{UL\}$ $2 : \{UM\}$ $3 : \{UM, LM\}$ $4 : \{UR, LM\}$
$5 : \{UR\}$ $6 : \{UL, CR\}$ $7 : \{UM, CR\}$ $8 : \{UM, LM, CR\}$
$9 : \{LM, CR\}$ $10 : \{CR\}$ $11 : \{CL, CR\}$
$12 : \{UM, CL, CR\}$ $13 : \{UM, LM, CL, CR\}$
$14 : \{LM, CL, CR\}$ $15 : \{CL, CR\}$ $16 : \{CL\}$
$17 : \{CL, UM\}$ $18 : \{UM, LM, CL\}$ $19 : \{CL, LM\}$
$20 : \{CL, LR\}$ $21 : \{LL\}$ $22 : \{UM, LL\}$
$23 : \{UM, LM\}$ $24 : \{LM\}$ $25 : \{LR\}$.

Now we divide some sub-rectangles further for those that can be covered by either UM or LM.

$7_1 : \{UM_1, CR\}$ $7_2 : \{UM_2, CR\}$
$8_1 : \{UM_1, LM_1, CR\}$ $8_2 : \{UM_1, LM_2, CR\}$
$8_3 : \{UM_2, LM_1, CR\}$ $8_4 : \{UM_2, LM_2, CR\}$
$9_1 : \{LM_1, CR\}$ $9_2 : \{LM_2, CR\}$
$12_1 : \{UM_1, CL, CR\}$ $12_2 : \{UM_2, CL, CR\}$
$13_1 : \{UM_1, LM_1, CL, CR\}$ $13_2 : \{UM_1, LM_2, CL, CR\}$
$13_3 : \{UM_2, LM_1, CL, CR\}$ $13_4 : \{UM_2, LM_2, CL, CR\}$
$14_1 : \{LM_1, CL, CR\}$ $14_2 : \{LM_2, CL, CR\}$
$17_1 : \{UM_1, CL\}$ $17_2 : \{UM_2, CL\}$
$18_1 : \{UM_1, LM_1, CL\}$ $18_2 : \{UM_1, LM_2, CL\}$
$18_3 : \{UM_2, LM_1, CL\}$ $18_4 : \{UM_2, LM_2, CL\}$
$19_1 : \{LM_1, CL\}$ $19_2 : \{LM_2, CL\}$

By Lemma 5, we can divide all nodes into two part in polynomial time such that one part is covered by disks in $UL \cup UM \cup UR$ and $LL \cup LM \cup LR$. The other part is covered by $CL \cup CR$, by Lemma 1 of [3], we can find an optimal solution for both parts. Thus by combining the two optimal solution, we can find a 2-approximation solution for $\mu \times 2\mu$ rectangle covering in this case.

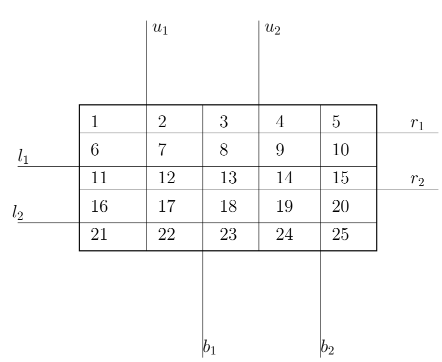

Fig. 13. Partition a $\mu \times 2\mu$ rectangle into 25 small sub-rectangles.

We enumerate all other cases and use the same method to find a 2-approximation solution for 1×2 rectangle. Actually, after fixing the relative position of horizontal lines l_1, l_2, r_1, r_2, there are totally 6 for the relative position of vertical lines u_1, u_2, b_1, b_2. We can divide the nodes in the cell into two area such that we can find 2-approximation algorithm to solve the problem. There are only 36 cases for the partition of the rectangle, thus we can find 2-approximation solution in polynomial time. Therefore, we have the following theorem:

Theorem 10: We can find 2-approximation solution in polynomial time for MWDC in a $\mu \times 2\mu$ rectangle with all disks outside of the rectangle.

C. Proof of Lemma 2

Proof:

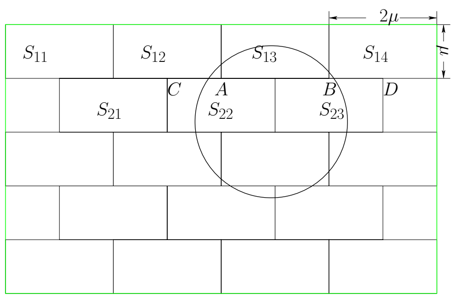

Fig. 14. A disk can intersect at most 5 rectangles in two consecutive strips.

We prove that a disk can intersect at most 5 rectangles from two consecutive horizontal strips. As the width of a rectangle is $2\mu \approx 1.414$, the diameter of a disk is 2, thus a disk can intersect at most 3 rectangles from a strip Consider the disk intersecting 3 disks from a strip, we prove that the disk can intersect at most 2 rectangles from either of its two neighboring strips. We consider the case for the lower neighboring strip first. the case for the upper neighboring strip can be proved similarly. Suppose the disk intersects S_{12}, S_{13}, S_{14} (Figure 14), then the disk must contain point A, B. If the disk intersects 3 rectangles in its lower neighboring strip, then it must contain either C or D. Assume it contains C, thus it contains BC. Since $\|BC\| = 3\mu > 2$, then the radius of disk should be greater than 2, which causes contradiction.

Thus a disk can intersect at most 5 rectangles in two consecutive strips. Since the height of a rectangle is $\mu \approx 0.707$, a disk can intersect at most 4 strips. There it intersects at most $5 \times 2 = 10$ rectangles. This finishes the proof. ■