J Comb Optim (2009) 18: 179–194
DOI 10.1007/s10878-008-9146-0

A better constant-factor approximation for weighted dominating set in unit disk graph

Yaochun Huang · Xiaofeng Gao · Zhao Zhang · Weili Wu

Published online: 1 March 2008

Abstract This paper presents a $(10 + \varepsilon)$-approximation algorithm to compute minimum-weight connected dominating set (MWCDS) in unit disk graph. MWCDS is to select a vertex subset with minimum weight for a given unit disk graph, such that each vertex of the graph is contained in this subset or has a neighbor in this subset. Besides, the subgraph induced by this vertex subset is connected. Our algorithm is composed of two phases: the first phase computes a dominating set, which has approximation ratio $6 + \varepsilon$ (ε is an arbitrary positive number), while the second phase connects the dominating sets computed in the first phase, which has approximation ratio 4.

Keywords Wireless network · Connected dominating set · Unit disk graph

1 Introduction

1.1 Dominating set problem in general graphs

Minimum Dominating Set Problem (MDS) is a famous optimization problem in graph theory. It is widely used in many fields such as wireless network. The for-

This work is supported in part by National Science Foundation under grant CCF-9208913 and CCF-0728851; and also supported by NSFC (60603003) and XJEDU.

Y. Huang · X. Gao (✉) · W. Wu
Department of Computer Science, University of Texas at Dallas, Richardson, USA
e-mail: xxg052000@utdallas.edu

Y. Huang
e-mail: yxh038100@utdallas.edu

W. Wu
e-mail: wxw022000@utdallas.edu

Z. Zhang
College of Mathematics and System Sciences, Xinjiang University, Urumqi, Xinjiang, China
e-mail: zhzhao@xju.edu.cn

mal definition of this problem is: Given an undirected graph $G = (V, E)$, we want to find a subset $D \subseteq V$ of its vertices with minimum number, such that for each vertex in V, it is either in D or has a neighbor in D. Then D is a dominating set of G. If we assign each vertex a weight, which is motivated by applications in wireless ad-hoc networks (e.g., nodes in a wireless ad-hoc network have different limited energies), our problem will become Minimum Weighted Dominating Set Problem (MWDS). The formal definition is: Given an undirected graph $G = (V, E, W)$, where W is the weight set for V, we want to find a minimum weighted dominating set $D \subseteq V$.

In network transmission process, it is not enough to only select minimum weighted dominating set. Each node should also be able to communicate with any other node in the graph. Therefore, we need Minimum Weighted Connected Dominating Set Problem (MWCDS): find a dominating set $D \subseteq V$, such that the subgraph induced by D is connected.

Garey and Johnson (1979) showed that MDS is $\mathcal{NP}$-hard. In fact, MDS for general graphs is polynomially equivalent to the Set Cover problem (Bar-Yehuda and Moran 1984). Therefore, no polynomial time algorithm can achieve an approximation ration better than $O(\log n)$ (Vazirani 2001) (n is the number of vertices in graph), unless $\mathcal{NP} \subseteq DTIME[n^{O(\log\log n)}]$ (Feige 1996). Till now, the best known approximation ratio for MWCDS in general graphs is $O(\log n)$ (Guha and Khuller 1999).

1.2 Dominating set problem in unit disk graph

In this paper we mainly concern MWDS and MWCDS in Unit Disk Graph (UDG). A Unit Disk Graph is a graph in which each vertex is associated with a unit disk in the plane. For a given UDG G, we use d to denote a unit disk (which also means the center of this disk. To distinguish the two expressions, we will call *center d* or *disk d*). Two vertices are adjacent (or there's an edge between two vertices) if and only if their corresponding unit disk intersect each other. In another word, if d_1 and d_2 are adjacent, we will have $dist(d_1, d_2) \leq 2$. Therefore, we can increase the radius of the d_1 and d_2 from 1 to 2 and obtain an equivalent statement: if d_1 and d_2 are adjacent, then the center of d_2 will locate inside disk d_1, or d_1 will cover d_2 (and vice versa). In the following sections, we will use disks with radius 2 for convenience.

A disk d_1 is said to *dominate* (or be *dominated by*) another disk d_2, if d_1 and d_2 are adjacent in G. A vertex set D of a graph G is said to be a *dominating set* if every vertex in G is either in D or is dominated by a vertex in D. A dominating set D is connected if $G[D]$, the subgraph of G induced by D, is connected. Then our problem is just to find a minimum weighted connected dominating set in G (MWCDS).

The motivation to study dominating sets in unit disk graphs comes from wireless ad-hoc networks, where dominating sets have been proposed for the construction of routing backbones (Dai et al. 2002). Each node of the graph represents a wireless device, and two hosts, located closely together within wireless transmission range of each other, are connected by an edge representing that they can communicate with each other. Besides, a message that is broadcasted by all nodes of a dominating set should be received by the whole network. Therefore, a small connected dominating set is an energy-efficient routing backbone (Wu and Li 1999). Recent work has emphasized that ad-hoc networks are often heterogeneous since different nodes have different capabilities. Therefore it is meaningful to assign weights to the nodes (e.g., assign small weight to nodes with a large remaining battery life) and aim to determine a (connected) dominating set of minimum weight (Wang and Li 2005).

Clark et al. (1990) proved that MDS is $\mathcal{NP}$-hard for UDG, and Lichtenstein (1982) showed that MCDS is $\mathcal{NP}$-hard for UDG. Marathe et al. (1995) gave constant-factor approximation algorithms for MDS and MCDS in UDG. Besides, for MDS in UDG, a PTAS was presented by Hunt III et al. (1998), based on the shifting strategy (Baker 1994; Hochbaum and Maass 1985). For the weighted version, Ambühl et al. (2006) gave a constant-factor approximation algorithm for MWCDS in UDG with approximation ratio 89.

1.3 Our result

This paper presents an approximation algorithm for MWCDS problem. The whole algorithm can be divided into two phases. Phase I shows an approximation algorithm to find a MCDS for a given UDG G. This algorithm use two strategies: Using Dynamic Programming to select Dominating Set for a strip; Enumerating all possible condition and then using shifting to eliminate the boundary influence. The whole approximation ratio of Phase I is $6+\varepsilon$, which is the smallest approximation algorithm for MWCDS in UDG.

Phase II uses an approach based on a minimum spanning tree calculation to add disks to the solution in order to make the dominating set connected. It yields a 4-approximation algorithm to connect the dominating set. Therefore, the whole approximation ratio of our algorithm is $6+\varepsilon+4=10+\varepsilon$, where ε is an arbitrary positive number.

The structure of our paper is as follows. Section 2 discusses about the Phase I of the whole algorithm, including three subsections to illuminate the algorithm step by step. It also contains shifting strategy and proof of approximation ratio. Section 3 mainly exhibits the 4-approximation algorithm to find additional disks and connect dominating set which is selected by Phase I. And lastly, Sect. 4 gives final conclusion and future works for the approximation algorithms.

2 Computing minimum weight dominating set

In this section we present an approximation algorithm to the Minimum Weight Dominating Set Problem (MWDS) with performance ratio $6+\varepsilon$. We firstly partition the plane into squares, and then select a dominating subset within a region of $K \times K$ squares (K is a given constant). Lastly we combine each subset together, forming a dominating set for the whole plane. To make convenience, we choose K to be an even number. Using shifting policy, we do previous processes $\frac{K}{2}-1$ times, and choose the minimum solution as our final result.

2.1 Partition

Given a UDG G containing n disks in the plane. Let $\mu < \sqrt{2}$ be a real number which is sufficiently close to $\sqrt{2}$, say $\mu = 1.4$. Partition the area into squares with side length μ. If the whole area has boundary $P(n) \times Q(n)$, where $P(n)$ and $Q(n)$ are two polynomial functions on n, then given the integer even constant K, and let $K \times K$ squares form a Block, our partition will have at most $(\lceil \frac{P(n)}{K} \rceil + 1) \times (\lceil \frac{Q(n)}{K} \rceil + 1)$ Blocks. We will discuss algorithm to compute minimum weight dominating subset for each block firstly, and then combine them together.

2.2 MWDC in $K \times K$ squares

Assume each Block B has K^2 squares S_{ij}, for $i, j \in \{0, 1, \dots, K-1\}$. Let V_{ij} be the set of disks in S_{ij}. If we have a dominating set D for this Block, then for each square S_{ij}, it's corresponding dominating set is (1): either a disk from inside S_{ij} (since $dist(d, d') \le 2$ for any two disks within this square), or (2): a group of disks from neighbor region around S_{ij}, the union of which can cover all disk centers inside the square. Then if we want to select minimum weight dominating set, for each square we will have two choice. However, instead of selecting dominating set square by square, we hope to select them strip by strip to avoid repeated computation for some disks. For this purpose, we have the following lemmas.

Lemma 1 (Ambühl et al. 2006) *Let P be a set of points located in a strip between lines $y = y_1$ and $y = y_2$ for some $y_1 < y_2$. Let D be a set of weighted disks with uniform radius whose centers are above the line $y = y_2$ or below the line $y = y_1$. Furthermore, assume that the union of the disks in D covers all points in P. Then a minimum weight subset of D that covers all points in P can be computed in polynomial time.*

The proof of result for Lemma 1 is in fact constructive. It gives a polynomial time algorithm by dynamic programming. It says that as long as the set of centers P in a horizontal strip can be dominated by a set of centers D above and/or below the strip, then an optimal subset of D dominating P can be found in polynomial time.

Our next work is to select some disks for each square within a strip so that those disks can be covered by disks from the upper and lower strips. To better illuminate the strategy, we divide the neighbor parts of S_{ij} into eight regions $UL, UM, UR, CL, CR, LL, LM, LR$ as shown in Fig. 1. The four line forming S_{ij} are $x = x_1$, $x = x_2$, $y = y_1$ and $y = y_2$. Denote by $Left = UL \cup CL \cup LL$, $Right = UR \cup CR \cup LR$, $Up = UL \cup UM \cup UR$, $Down = LL \cup LM \cup LR$. After that, we will have Lemma 2.

Lemma 2 *Suppose $p \in V_{ij}$ is a disk in S_{ij} which can be dominated by a disk $d \in LM$. We draw two lines p_l and p_r, which intersect $y = y_1$ by angle $\frac{\pi}{4}$ and $\frac{3\pi}{4}$. Then the shadow P_{LM} surrounded by $x = x_1$, $x = x_2$, $y = y_1$, p_l and p_r (shown in Fig. 2) can also be dominated by d. Similar results can be hold for shadow P_{UM}, P_{CL} and P_{CR}, which can be defined with a rotation.*

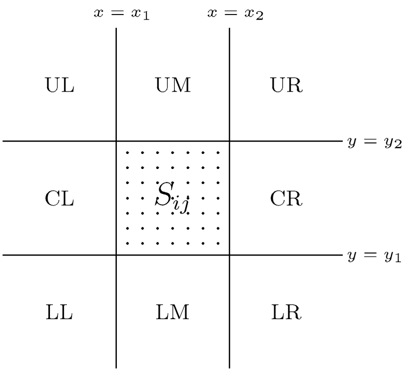


Fig. 1 S_{ij} and its neighbor regions

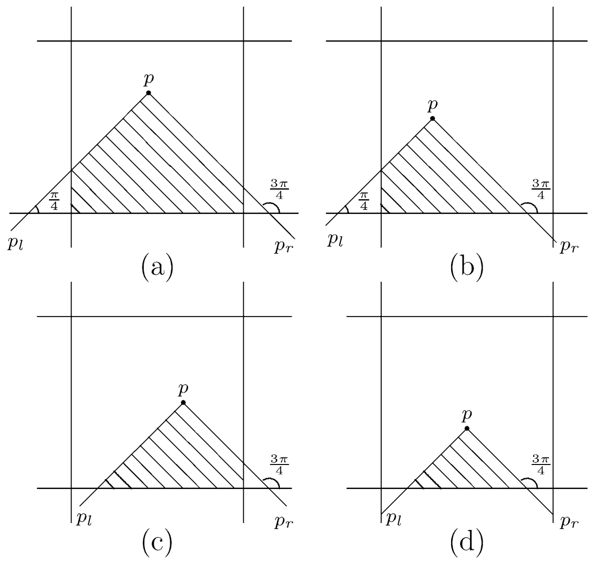


Fig. 2 Different shape for shadow P_{LM}

Proof We split shadow P_{LM} into two half with vertical line $x = x_p$, where x_p is x-coordinate of disk p. Then we prove that the right half of P_{LM} can be covered by d. The left half can be proved symmetrically. Let o be intersection point of $x = x_p$ and $y = y_1$, a that of p_r and $x = x_2$ (or p_r and $y = y_1$), and b that of $x = x_2$ and $y = y_1$. Intuitionally, the right half can be either a quadrangle *pabo* or a triangle *pao*. We will prove both cases as follows.

Quadrangle case Draw the perpendicular line of the line segment pa, namely p_m. When d is under p_m as in Fig. 3a, we will have $dist(d, a) \leq dist(p, d) \leq 2$. Besides, it is trivial that $dist(d, o)$ and $dist(d, b)$ are all < 2. Thus d can cover the whole quadrangle. When d is above the line p_m as in Fig. 3b, we draw an auxiliary line $y = y_a$ parallel with $y = y_1$, and $x = x_d$ intersecting $y = y_a$ at point c. Since d lies

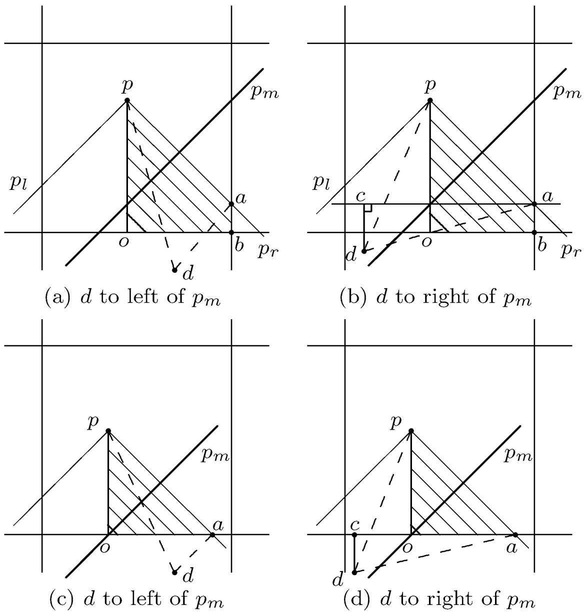


Fig. 3 Shape of shadow and location of d

above p_m, $\angle cad \le \pi/4$, and thus

$$dist(d,a) = \frac{dist(c,a)}{\cos\angle cad} < \frac{\sqrt{2}}{\cos\frac{\pi}{4}} = 2.$$

Note that both $dist(d,o)$ and $dist(d,b)$ are less than 2, d can cover the whole quadrangle.

Triangle case Similarly, draw p_m as above. The proof remains when d is under p_m (see Fig. 3c). When d is above p_m as in Fig. 3d, we draw auxiliary line $x = x_d$ intersecting $y = y_1$ at c. Then we will get the same conclusion. □

With help of Lemma 2, we can select a region from S_{ij}, where the disks inside this region can be covered by disks from *Up* and *Down* neighbor area. We name this region as "sandglass", with formal definition as follows:

Definition 1 (Sandglass) If D is a dominating set for square S_{ij} and $D \cap V_{ij} = \emptyset$, then there exists a subset $V_M \subseteq V_{ij}$ which can only be covered by disks from *UM* and *LM* (we can set $V_M = \emptyset$ if there's no such disks). Choose $V_{LM} \subseteq V_M$ the disks that can be covered by disks from *LM*, draw p_l and p_r line for each $p \in V_{LM}$. Choose the leftmost p_l and rightmost p_r and form a shadow similar as that in Lemma 2. Symmetrically, choose V_{UM} and form a shadow with leftmost and rightmost lines. The union of the two shadows form a "sandglass" region $Sand_{ij}$ of S_{ij}. (See Fig. 4a, where solid circle represents V_{LM}, while hollow circle represents V_{UM}) Fig. 4b, c, d give other possible shapes of $Sand_{ij}$.

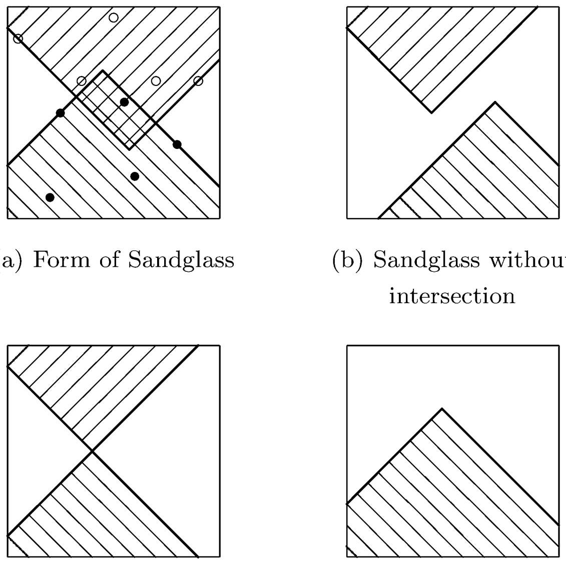


Fig. 4 Sandglass $Sand_{ij}$ for S_{ij}

(a) Form of Sandglass

(b) Sandglass without intersection

(c) Sandglass with Single Disk

(d) Sandglass with Half Side

Lemma 3 *Suppose D is a dominating set for* S_{ij}, *and* $Sand_{ij}$*'s are chosen in the above way. Then any disks in* $Sand_{ij}$ *can be dominated by disks only from neighbor region* $Up \cup Down$, *and disks from* $S_{ij} \setminus Sand_{ij}$ *can be dominated by disks only from neighbor region* $Left \cup Right$.

Proof Suppose to the contrary, there exits a disk $d \in Sand_{ij}$ which cannot be dominated by disks from $Up \cup Down$. Since D is a dominating set, there must be a $d' \in CL \cup CR$ which dominates d. Without loss of generality, assume d belongs to lower half of the sandglass which is formed by p_1 and p_2, and let $d' \in CL$ (see Fig. 5). Based on our assumption, d cannot locate in p_1's triangle shadow to *Down* region (otherwise since p_1 can be dominated by a disk from *LM*, d can also be dominated by this disk). We then draw d_l and d_r to *CL* region and form a shadow to *CL*. Then by Lemma 2 every disk from this shadow can be dominated by d'. Obviously p_1 belongs to this region, but p_1 is a disk which cannot be dominated by disks from *CL*, a contradiction. □

Till now we already find "sandglass" region in which disks can be dominated by disks only from *Up* and *Down* regions. In our algorithm, for each square S_{ij}, we can firstly decide whether to choose a disk inside this square as dominating set, or to choose a dominating set from its neighbor region. If we choose latter case, the algorithm will randomly select 4 disks d_1, d_2, d_3 and d_4 from S_{ij} and make corresponding sandglass (we can also choose less than 4 disks to form the sandglass). By enumeration of all possible sandglasses including the case of choosing one disk

Fig. 5 Proof for sandglass

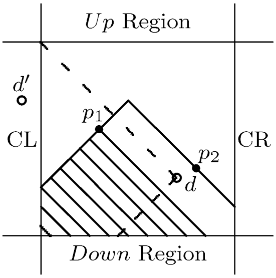
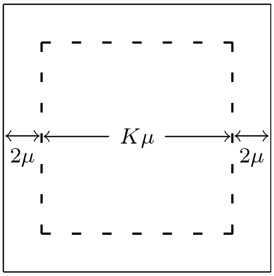


Fig. 6 Block selection



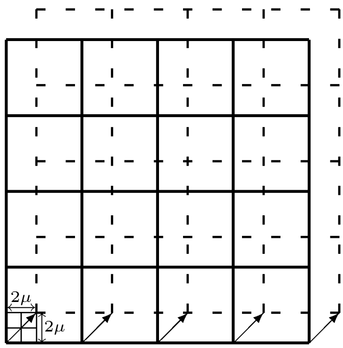
inside the square, for all squares within $K \times K$ area, there are at most $[\sum_{i=0}^{4} C_n^i \cdot 2^i]^{K^2}$ choices (n is the number of disks), which can be calculated within polynomial time. Besides, when consider choosing dominating set from neighbor regions, we should also include regions around this $K \times K$ areas such that we will not miss disks outside the region. Therefore, we should consider $(K+4) \times (K+4)$ area, where the inner region is our selected block and the surrounding four strips are the assistance (shown as Fig. 6).

Algorithm 1: Calculate MWDS in $K \times K$ squares	
Step 1	For each S_{ij}, choose its sandglass or select a $d \in S_{ij}$.
Step 2	If $d \in S_{ij}$ is selected, then remove d and all disks dominated by d.
Step 3	For each strip $\bigcup_{j=1}^{K} S_{ij}$ from $i = 1$ to K, calculate dominating set for the union of disks in the sandglasses.
Step 4	For each strip $\bigcup_{i=1}^{K} S_{ij}$ from $j = 1$ to K, calculate dominating set for the remaining disks not covered by Step 3.
The Union of disks chosen in the above steps form a MWDS for $K \times K$ squares.	

2.3 MWDC for the whole region

As discussed above, if our plane has size $P(n) \times Q(n)$, then there are at most $(\lceil \frac{P(n)}{K} \rceil + 1) \times (\lceil \frac{Q(n)}{K} \rceil + 1)$ Blocks in the plane. We name each block B^{xy}, where $0 \le x \le \lceil \frac{P(n)}{K} \rceil + 1$ and $0 \le y \le \lceil \frac{Q(n)}{K} \rceil + 1$. Then, using Algorithm 1 to calculate dominating set for each block, and by combining them together, we obtain a dominating set for our original partition.

Fig. 7 Move blocks



Next, we move our blocks to different positions by shifting policy. Move every block two squares right and two squares up to its original position, which can be seen from Fig. 7. Then calculate dominating set for each block again, and combine the solution together. We do this process $\frac{K}{2}$ times, choose the minimum solution as our final result. The whole process can be shown as Algorithm 2.

Algorithm 2: Calculate MWDS for the whole plane	
Step 1	For a certain partition, calculate MWDS for each block B^{xy}, sum the weight of MWSD for each block and form a solution.
Step 2	Move each block to two squares to the right and two squares to the top of the original block.
Step 3	Repeat Step 1 for new partition, get a new solution.
Step 4	Repeat Step 2 for $\lceil \frac{K}{2} \rceil$ times, and choose the minimum solution among those steps.
The solution from Step 4 is our final result.	

2.4 Performance ratio

In the following, we extend our terminology 'dominate' to points (a point is a location which is not necessarily a disk). A point p is *dominated* by a set of disks if the distance between p and at least one center of the disks is not more than 2. We say an area is *dominated* by a set of disks if every point in this area is dominated by the set of disks. Let OPT be optimal solution for our problem and $w(OPT)$ the weight of optimal solution.

Theorem 1 *Algorithm* 2 *always outputs a dominating set with weight within* $6 + \varepsilon$ *times of the optimum one.*

Proof Our proof mainly has two phases. The first phase analyzes that our Algorithm 1 gives a 6-approximation for disks in $K \times K$ squares. The second phase proves that result from algorithm 2 is less than $(6 + \varepsilon) \cdot w(OPT)$.

Phase 1 If a disk has radius 2, and our partition has side length $\mu < \sqrt{2}$, then a disk may dominate disks from at most 16 squares, which can be shown from

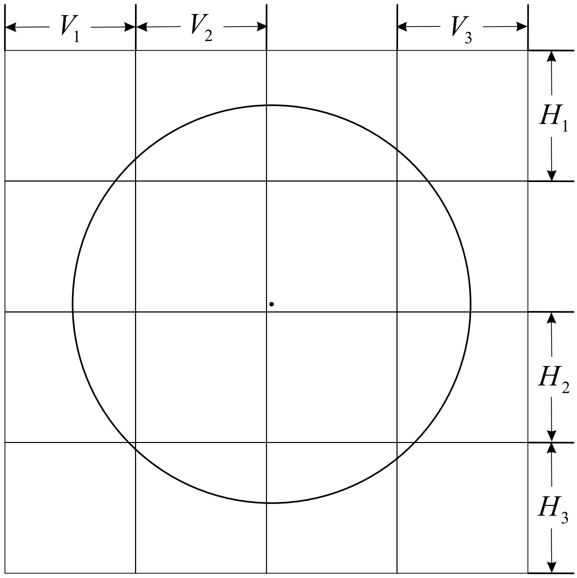


Fig. 8 An example for disk cover region

Fig. 8. Simply, if a disk in *OPT* is used to dominate the square it belongs to, then we will remove this disk before calculating MWDS for strips. Therefore it will be used only once. If a disk is not used to dominate the square containing it, then it may be used 3 times in calculating its 3 horizontal neighbor strips (H_1, H_2 and H_3 as shown in Fig. 8), and another 3 times in calculating its 3 vertical neighbor strips (V_1, V_2 and V_3 in Fig. 8). Therefore, Algorithm 1 is a 6-approximation for each block.

Phase 2 Now we consider the disks in side strips for a block. As discussed above, when calculating MWDS for a strip, we may use disks within $(K+2) \times (K+2)$ squares. Therefore, we can divide a block $B^{(xy)}$ into three kinds of squares, just as shown in Fig. 9 ($0 \le x \le \lceil \frac{P(n)}{K} \rceil$, and $0 \le y \le \lceil \frac{Q(n)}{K} \rceil$). If a disk belongs to inner part $\mathcal{A}$ of $B^{(xy)}$, it will be used at most 6 times during calculating process. We name those disks as d_{in}. If a disk belongs to side part $\mathcal{B}$ of $B^{(xy)}$, it may be used at most 5 times for calculating $B^{(xy)}$, but it may used at most 4 times when calculating $B^{(xy)}$'s neighbor block. We name those disks as d_{side}. If a disk belongs to corner squares $\mathcal{C}$ of $B^{(xy)}$, it may be used at most 4 times for calculating $B^{(xy)}$, and at most 8 times for neighbor blocks. We name those disks as d_{corner} . In addition, we know that during shifting process a node can stay at most 4 times in side or corner square. If we name l as the lth shifting, then our final solution will have the following inequality:

$$W(Solution) = \min_l \left\{ \sum_{Sol_l} [6w(d^l_{in}) + 9w(d^l_{side}) + 12w(d^l_{corner})] \right\}$$

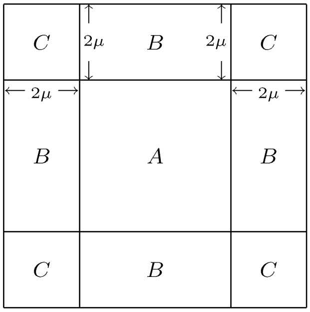


Fig. 9 Divide block $B^{(xy)}$ into 3 parts

$$\le \frac{1}{\frac{K}{2}} \sum_{l=0}^{\frac{K}{2}} \left\{6w(d_{in}^l) + 4 \cdot 12w(d_{side}^l + d_{corner}^l)\right\}$$

$$= 6w(OPT) + \frac{42}{\frac{K}{2}} w(OPT)$$

$$\le (6+\varepsilon)w(OPT)$$

where $\varepsilon = 42/\frac{K}{2}$ can be arbitrarily small when K is sufficiently large. □

3 Connected dominating set

Having computed a dominating set D of G, the subgraph of G induced by D, denoted by $G[D]$, is not necessarily connected. The vertex set of a connected component of $G[D]$ is called a *cluster* of D, and the vertices in D are called *in-centers*, vertices in $V(G) \setminus D$ are called *out-centers*. In the following, we are to add some out-centers to D such that the resulting new dominating set induces a connected subgraph of G.

A path P is said to connect two clusters C_1 and C_2 if the two ends of P are in C_1 and C_2 respectively and all internal vertices $d_1, \ldots, d_t$ of P are out-centers. Denote such a path by $C_1 d_1 \ldots d_t C_2$. The weight of P is $w(P) = \sum_{j=1}^{t} w(d_j)$.

Suppose $\mathcal{C}$ is the set of clusters of D. Construct an auxiliary graph H as follows: The vertices of H correspond to the clusters in $\mathcal{C}$; For every path P of length at most 3 in G which connects a cluster $C_1 \in \mathcal{C}$ to another cluster $C_2 \in \mathcal{C}$, add an edge e between C_1 and C_2, the out-centers on P are said to *define* e, the weight of e is the sum of the weights of the out-centers defining e.

Algorithm 3: Connect clusters into MWCDS	
Step 1	Construct the auxiliary graph H as described above. Set $D_1 = D$.
Step 2	Compute a minimum spanning tree T of H.
Step 3	For each edge $e \in E(T)$, add the out-centers defining e to D_1.
The solution from Step 3 is our final result.	

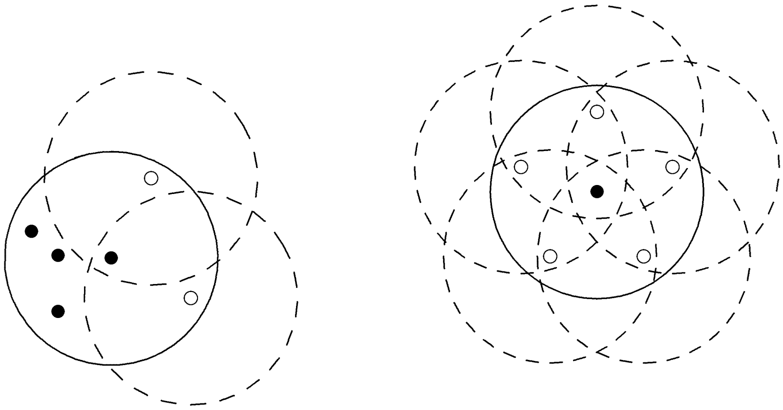


Fig. 10 The left is a disk of Type 2, the right is a disk of Type 5

We are to show that the total weight of the set of added out-centers is within 4 times of the optimal CDS. For this purpose, we need more terminologies. For a point d in the plane, denote by $C(d,2)$ the disk with center d and diameter 2. For a set of centers U, we say that a center d (or the corresponding disk d) is of *Type 1 with respect to U* if there exists exactly one 'imaginary' point $d' \in C(d,2)$ such that d is the only center in U which lies in the area $C(d,2) \cap C(d',2)$; for $2 \le i \le 5$, a center d is of *Type i with respect to U* if there are exactly i imaginary points $d_1, \dots, d_i \in C(d,2)$ such that $\angle d_{j_1} d d_{j_2} > \pi/3$ for any $j_1, j_2 \in \{1, \dots, i\}$ and $j_1 \ne j_2$, and for each $j \in \{1, \dots, i\}$, d is the only center in U which lies in $C(d,2) \cap C(d_j,2)$. A center d which does not belong to any of the above five types is said to be of *Type 0 with respect to U*. The circles in Fig. 10 are two disks of Type 2 and Type 5 respectively, where solid points are centers of disks and small circles are 'imaginary' points, big dashed circles are the imaginary disks whose centers are the small circles. The idea of such a definition is that for a disk d, the larger the 'blank' area (no centers in this area is needed to be covered by disk d) is, the times that disk d is used repeatedly to connect different clusters is less.

A set of clusters $\mathcal{C}$ is said to be connected through a set of centers U if adding U to the center set of $\mathcal{C}$ results in a connected graph.

Lemma 4 *Let G be a connected graph, D be a dominating set of G, $\mathcal{C}$ be the set of clusters of D, and U be a set of centers such that $\mathcal{C}$ can be connected through U. Then there exists a set of paths $\mathcal{P}$ in G such that*

(1) *$\mathcal{C}$ is connected through paths in $\mathcal{P}$,*
(2) *each path in $\mathcal{P}$ is of length at most* 3 *in G, and*
(3) $w(\mathcal{P}) \le 4w(V_0) + 3w(V_1) + 2w(V_2) + w(V_3)$, *where $V_i \subseteq U$ is the set of centers of Type i with respect to U.*

Proof It can be assumed that U does not contain any center in $\mathcal{C}$. We prove the lemma by induction on $|U|$.

If $|U| = 0$, then $\mathcal{C}$ contains only one cluster, and the result is obviously true.

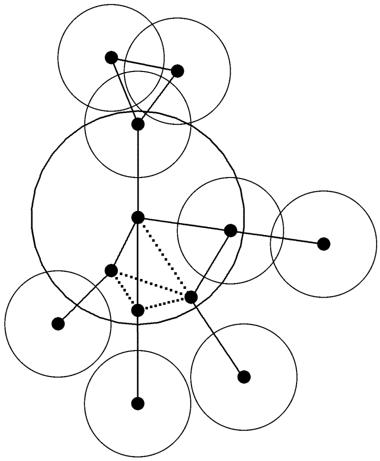


Fig. 11 The subgraph G' of G

If $|U| = 1$, suppose $U = \{d\}$. If $d \in V_i$, then $|\mathcal{C}| \leq 5 - i$. Suppose $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$. Let $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{C}|-1}\}$ where $P_j = C_j d C_{j+1}$ ($j = 1, \ldots, |\mathcal{C}| - 1$). Then $\mathcal{P}$ clearly satisfies conditions (1), (2), and condition (3) follows from $w(\mathcal{P}) = (|\mathcal{C}| - 1)w(d) \leq (4 - i)w(V_i)$.

Next, suppose $|U| = k \geq 2$ and the result holds for smaller set of centers.

Let d be a center in U with the minimum weight. For a center d' and a cluster $C \in \mathcal{C}$, denote by $dist(d', C) = \min\{dist(d', c) \mid c \in C\}$. We construct a subgraph G' of G by specifying the adjacency of centers in $C(d, 2)$: For each cluster $C \in \mathcal{C}$ with $C \cap C(d, 2) \neq \emptyset$, join d to a center $c \in C$ with $dist(d, c) = dist(d, C) (\leq 2)$; for each center $d' \in C(d, 2) \cap U$ with $dist(d', C) \leq 2$ for some $C \in \mathcal{C}$ with $C \cap C(d, 2) \neq \emptyset$, join d' to a center $c \in C$ with $dist(d', c) = dist(d', C) (\leq 2)$; all other centers in $C(d, 2) \cap U$ are joined to d. This is illustrated in Fig. 11, where the dotted lines are edges in G but not in G'. Since G' is a connected subgraph of G, it suffices to find a desired set of paths $\mathcal{P}$ in G'. Suppose i_d is the integer such that d is of Type i_d with respect to U. We distinguish two cases.

Case 1 $t \leq 5 - i_d$. Denote the connected components of $G' - d$ by $G_1, \ldots, G_t$. Suppose, without loss of generality, that the first t_1 components $G_1, \ldots, G_{t_1}$ are connected to d through clusters $C_1, \ldots, C_{t_1}$, and the last $t_2 = t - t_1$ components $G_{t_1+1}, \ldots, G_t$ are connected to d through out-points $d_{t_1+1}, \ldots, d_t$. For each G_j, denote by $U^{(j)} = V(G_j) \cap U$ and $V_i^{(j)}$ the set of centers in $U^{(j)}$ of Type i with respect to $U^{(j)}$. By induction hypothesis, there is a set of paths $\mathcal{P}_j$ in G_j satisfying conditions (1) to (3). In particular, condition (3) has the form

$$w(\mathcal{P}_j) \leq 4w(V_0^{(j)}) + 3w(V_1^{(j)}) + 2w(V_2^{(j)}) + w(V_3^{(j)}). \tag{1}$$

Note that for $j \in \{t_1 + 1, \ldots, t\}$, d is the only center in $C(d, 2)$ which is adjacent with d_j in G', hence d_j is the only center in $U^{(j)}$ which lies in $C(d, 2) \cap C(d_j, 2)$. It follows that if $d_j \in V_i$ with respect to U, then $d_j \in V_{i+1}^{(j)}$ with respect to $U^{(j)}$. Summing up inequalities (1) over $j = 1, \ldots, t$, taking the above consideration into account, we have

$$\sum_{j=1}^{t} w(\mathcal{P}_j) \le 4w(V_0) + 3w(V_1) + 2w(V_2) + w(V_3) - \sum_{j=t_1+1}^{t} w(d_j) - (4 - i_d)w(d).$$

Let $\mathcal{P}'$ be the set of paths $\{C_1 d C_j\}_{j=2}^{t_1} \cup \{C_1 d d_j C_j'\}_{j=t_1+1}^{t}$, where for $j \in \{t_1 + 1, \dots, t\}$, C_j' is a cluster with $dist(d_j, C_j') \le 2$. Then

$$w(\mathcal{P}') = (t-1)w(d) + \sum_{j=t_1+1}^{t} w(d_j).$$

Let $\mathcal{P} = (\bigcup_{j=1}^{t} \mathcal{P}_j) \cup \mathcal{P}'$. Then

$$w(\mathcal{P}) \le 4w(V_0) + 3w(V_1) + 2w(V_2) + w(V_3) + (t + i_d - 5)w(d).$$

The result follows since $t \le 5 - i_d$ in this case.

Case 2 $t > 5 - i_d$. In this case, there are two centers d' and d'' in $C(d,2)$ which are joined to d in G', and $\angle d'dd'' < \pi/3$. We claim that both d' and d'' are in U. Otherwise, suppose $d' \in C \in \mathcal{C}$, then by noting that $dist(d', d'') < 2$, it follows from the construction of G' that d'' should be joined to C instead of d, a contradiction. Furthermore, one of $\angle dd'd''$ and $\angle dd''d'$ is greater than $\pi/3$, say $\angle dd''d' > \pi/3$. Let G'' be the subgraph of G' by erasing the adjacency between centers in $(C(d'',2) \cap C(d',2)) \setminus C(d,2)$ and d'' (note that centers in this area remain their adjacency with d', hence G'' is still connected). Let G_1 be the component of $G'' - d$ containing d'', and $G_2 = G'' - G_1$. For $j = 1, 2$, denote by $U^{(j)} = V(G_j) \cap U$, and $V_i^{(j)}$ the set of centers in $U^{(j)}$ of Type i with respect to $U^{(j)}$. Apply induction hypothesis to the connected subgraph graph G_j, we have a set of paths $\mathcal{P}_j$ connecting the clusters in G_j such that

$$w(\mathcal{P}_j) \le 4w(V_0^{(j)}) + 3w(V_1^{(j)}) + 2w(V_2^{(j)}) + w(V_3^{(j)}).$$

By the construction of G'', we see that if d'' is of Type i in G, then d'' is of Type $i+2$ in G_1. First, suppose d'' is not of Type 3. Then

$$w(\mathcal{P}_1) + w(\mathcal{P}_2) \le 4w(V_0) + 3w(V_1) + 2w(V_2) + w(V_3) - 2w(d'').$$

Let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \{Cdd''C''\}$, where C is a cluster joined to d and C'' is a cluster joined to d''. Then $\mathcal{P}$ satisfies conditions (1), (2) and

$$w(\mathcal{P}) \le 4w(V_0) + 3w(V_1) + 2w(V_2) + w(V_3) - 2w(d'') + w(d) + w(d'').$$

Then the result follows since $w(d) \le w(d'')$. If d'' is of Type 3, then d and d' are the only centers adjacent with d'', and thus G_1 is a singleton. Then the analysis is similar to the above (but easier) by setting $\mathcal{P} = \mathcal{P}_1 \cup \{Cdd''\}$. □

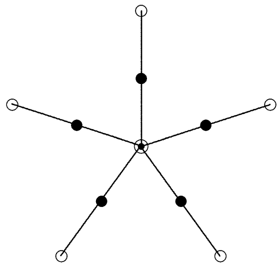


Fig. 12 A tight example of Theorem 2

Theorem 2 *Let D be a dominating set and U be the set of centers added to D by Algorithm 2. Then $w(U) \leq 4w(OPT)$, where OPT is the weight of an optimum solution to MWCDS.*

Proof Let D^* be an optimal solution. Taking $U = D^*$ in Lemma 4, we have a set of paths $\mathcal{P}$ connecting the set of clusters of $G[D]$, and $w(\mathcal{P}) \leq 4w(D^*)$. This set of paths corresponds to a spanning subgraph of H. Since U is obtained from a minimum spanning tree T of H, and $w(U)$ is exactly $w(T)$ by the definition of $w(e)$ for $e \in E(H)$, we have $w(U) \leq w(\mathcal{P})$. The result follows. □

The approximation ratio in Theorem 2 is tight in the following sense: Consider the unit disk graph in Fig. 12, every disk has weight 1. Then the optimum solution to WDS is the set of solid points, and the optimum solution to WCDS is the set of solid points plus the middle one. But by the 'spanning tree' strategy in Algorithm 3, the middle point is used exactly four times to connect different clusters.

Combining Theorems 1 and 2, we have

Theorem 3 *Algorithms 1, 2, 3 together give an approximation algorithm to MWCDS with performance ratio $10 + \varepsilon$.*

4 Conclusion

In our paper we give a $(10 + \epsilon)$-approximation algorithm for Minimum Weight Connected Dominating Set in Unit Disk Graph, which greatly improves the 89-approximation algorithm given in (Ambühl et al. 2006). Our main strategy is to partition the whole plane into squares, and form them into blocks. We compute MWDS for each block firstly, and then combine them together. After that, by shifting strategy, we avoid boundary influences and yield a $(6 + \varepsilon)$-approximation to MDS. Next, we use an algorithm to add some disks into existing dominating set, such that they can be connected. The approximation ratio of this step is 4, which is tight for using 'spanning tree' strategy. Hence, to reduce the whole approximation ratio, it may be more promising to improve on the first part.

References

Ambühl C, Erlebach T, Mihalák M, Nunkesser M (2006) Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In: Proceedings of the 9th international workshop on approximation algorithms for combinatorial optimization problems (APPROX 2006). LNCS, vol 4110. Springer, Berlin, pp 3–14

Baker BS (1994) Approximation algorithms for NP-complete problems on planar graphs. J Assoc Comput Mach 41(1):153–180

Bar-Yehuda R, Moran S (1984) On approximation problems related to the independent set and vertex cover problem. Discrete Appl Math 9:1–10

Clark BN, Colbourn CJ, Johnson DS (1990) Unit disk graphs. Discrete Math 86:165–177

Dai WF, Gao M, Stojmenovic I (2002) On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. J Commun Netw 4(1):59–70

Feige U (1996) A Threshold of *lnn* for approximating set cover. In: Proc. 28th ACM symposium on theory of computing. ACM, New York, pp 314–318

Garey MR, Johnson DS (1979) Computers and intractability. In: A guide to the theory of NP completeness. Freeman, New York

Guha S, Khuller S (1999) Improved methods for approximating node weighted Steiner trees and connected dominating sets. Inf Comput 150(1):57–74

Hochbaum DS, Maass W (1985) Approximation schemes for covering and packing problems in image processing and VLSI. J Assoc Comput Mach 32(1):130–136

Hunt HB III, Marathe MV, Radhakrishnan V, Ravi SS, Rosenkrantz DJ, Stearns RE (1998) NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. J Algorithms 26(2):238–274

Lichtenstein D (1982) Planar formulae and their uses. SIAM J Comput 11(2):329–343

Marathe MV, Breu H, Hunt HB III, Ravi SS, Rosenkrantz DJ (1995) Simple heuristics for unit disk graphs. Networks 25:59–68

Vazirani VV (2001) Approximation algorithms. Springer, Berlin

Wang Y, Li XY (2005) Distributed low-cost backbone formation for wireless ad hoc networks. In: Proceedings of the 6th ACM international symposium on mobile ad hoc networking and computing (MobiHoc 2005), pp 2–13

Wu J, Li H (1999) On calculating connected dominating set for efficient routing in ad-hoc wireless networks. In: Proc. of the 3rd international workshop on discrete algorithms and methods for mobile computing and commun, pp 7–14