IMPACT OF LOCALITY ON LOCATION AWARE UNIT DISK GRAPHS

ANDREAS WIESE[*,§] AND EVANGELOS KRANAKIS[**,§§]

Abstract. A network algorithm is local if the status of a vertex depends only on the vertices which are at most a constant (independent of the size of the network) number of hops away from it. Due to their importance for studies on wireless networks, recent years have seen a surge of activity on the design of local algorithms for the solution of a variety of network tasks. Nevertheless, there are only a few lower bounds known for approximation factors of local algorithms and none for local algorithms in the setting of location aware nodes. In this paper we investigate the impact of very low locality (i.e., number of hops) on the design of algorithms in location aware UDGs. We prove the first ever lower bounds for local algorithms of a given locality for minimum dominating and connected dominating set, maximum independent set and minimum vertex cover in the location aware setting.

Then we study the prospects of algorithms with very low localities. Despite of this restriction we propose local constant ratio approximation algorithms for solving these problems in Unit Disk Graphs. We compare the bounds obtained by designing even tighter upper bounds on Unit Line Graphs (a special class of UDGs whereby all vertices lie on the same line) and contrast them by proving lower bounds for arbitrary locality on these graphs.

1. Introduction

In networks which are formed by wireless devices we often lack a global entity to organize the network traffic. This is especially the case in ad hoc networks. So in this setting the devices need to form structures (e.g., communication backbones) by passing information from one to the other. As such networks are often much too large to be entirely known by a single node, we are interested in local algorithms. These are algorithms in which the status of a vertex v (i.e., whether or not a vertex is part of the dominating set, independent set, etc.) depends only on the vertices which are a constant number of edges (hops) away from v. We require this constant to be independent of the size of the network.

We model the wireless network with Unit Disk Graphs (UDGs). These are undirected graphs in the plane in which two vertices are connected by an edge if and only if their Euclidean distance is at most one unit. Unit disk graphs are widely

Research conducted while the authors were visiting the School of Computing Science at Simon Fraser University, Vancouver.

[*] Technische Universität Berlin, Institut für Mathematik, Germany.

[§] Research supported by a scholarship from DAAD (German Academic Exchange Service).

[**] School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6.

[§§] Research supported in part by NSERC (Natural Science and Engineering Research Council of Canada). Research supported in part by MITACS (Mathematics of Information Technology and Complex Systems).

used for modeling wireless networks. We assume that each node has knowledge of its geographic position in the plane (location awareness). As positioning systems like GPS become more and more common this assumption seems relevant.

In wireless networks, dominating sets play an important role for topology control. In such networks, nodes are often organized in clusters where one node is declared to be the cluster-head. This vertex is responsible to ensure the communication of the nodes in the cluster with other nodes in the network. So the cluster-heads form a dominating set in the underlying network graph. In order to be able to send messages from one cluster to another, one needs the set of dominators to be connected. This leads to a connected dominating set. For organizing the network as efficiently as possible we are looking for small (connected) dominating sets. In order to give the cluster-heads more control, it can be advantageous if they are not adjacent to each other [4]. So in this case they form a maximal independent set. Another important problem in graphs is the minimum vertex cover problem.

Given that several algorithms have been designed to solve these tasks, very little is known about lower bounds for approximation ratios of local algorithms. Such bounds would allow us to understand the prospects and limits of local algorithms. In particular, we want to comprehend how much restriction we impose on algorithms when we require a certain locality distance. In this paper we address this issue.

1.1. **Related Work.** The first results about local algorithms were given by Linial [8]. He gives the first bounds on locality distances for constructing a maximal independent set and a 3-coloring in an n-cycle. He also proves that at least $\sqrt{d}$ colors are needed for coloring a d-regular tree with radius r when the locality is restricted to $2r/3$. In [10] Naor and Stockmeyer provide a framework for local algorithms for Locally Checkable Labeling Problems (LCL). All problems which we discuss in this paper are LCLs. In his book [13], Peleg gives a locality sensitive perspective of distributed algorithms.

In general graphs all problems which we study - dominating and connected dominating set, independent set and vertex cover - are NP-hard [2]. Apart from vertex cover they do not even admit constant ratio approximation algorithms [14, 6]. For vertex cover there are several 2-approximation algorithms known, e.g., [1]. When restricting the case to unit disk graphs, constant ratio approximations [9] and PTASs [7, 12, 11] are known. However, all these algorithms are global in the sense that in order to determine whether a given vertex is in the computed set, we need knowledge of the entire graph. When looking for local algorithms, Kuhn et al. proposed local approximation schemes for maximum independent set and dominating set for growth-bounded-graphs. This class of graphs includes UDGs. However, in these algorithms the status of a vertex depends on the vertices which are up to $O(\log |V|)$ hops away from it, which is not local in our sense. Gfeller and Vicari presented a distributed PTAS for dominating set with the same locality properties [5].

In [3] Czyzowicz et al. presented the first local constant ratio approximation algorithms for our setting with performance ratios 5 and $7.453+\epsilon$ for dominating and connected dominating set, respectively. The locality distances of the dominating set algorithm is 11, the locality for the connected dominating set algorithm is not given in the paper. In [15] Wiese and Kranakis presented local PTASs for these two problems. However, their locality distances, though constant, can be very large.

Problem	UDG one hop	ULG one hop	Bound for locality k
Dominating Set	$12 \cdot OPT$	$3 \cdot OPT$	$1 + 3/(2k+3)$
Independent Set	$1/1801 \cdot OPT$	$\lfloor 1/2 \cdot OPT \rfloor$	$(1 + 1/k)^{-1}$
Vertex Cover	$12 \cdot OPT$	$2 \cdot OPT$	$1 + 1/k$

Table 1. Results for low localities presented in this paper. (Note that the bound for dominating set depends on the congruence class of k. The bound given in this table represents the worst case.)

Connected Dominating Set	
UDG one hop	no local constant ratio algorithm
UDG two hops	$216 \cdot OPT$
ULG one hop	$6 \cdot OPT$
Bound for locality k	$1 + 1/k$

Table 2. Results for connected dominating set presented in this paper.

1.2. **Results of this paper.** In this paper we try to assess the impact of locality on the algorithmic design of important computational tasks, like dominating set, connected dominating set, vertex cover, and independent set, in wireless networks. For arbitrary locality distances we show the first lower bounds for possible approximation ratios of local algorithms for all problems mentioned above in the setting of location aware nodes.

We also investigate the power of algorithms with very low localities. It turns out that despite the fact we are looking only at locality one we can still design constant ratio approximation algorithms for dominating set, independent set, and vertex cover. We prove that for connected dominating set there is no constant ratio algorithm with locality distance one. But we present such an algorithm with locality distance two.

In the proofs for the lower bounds we mostly employ unit disk graphs on a line (unit line graphs). In order to assess our bounds for each problem we present a local algorithm with locality one in unit line graphs. These algorithms achieve significantly better approximation ratios than the local algorithms for general unit disk graphs with this locality.

All results including all approximation ratios and bounds are presented in Tables 1 and 2.

1.3. **Organization of the paper.** The remainder of this paper is organized as follows: First we give some basic definitions and preliminaries. In the following sections we discuss one problem per section: dominating set, connected dominating set, independent set, and finally vertex cover. For each problem we prove our lower bounds for approximation ratios of local algorithms with arbitrary locality distance. We also give our algorithms for unit disk graphs and unit line graphs. For our dominating set algorithm on unit disk graphs we give an example which shows that the analysis of the approximation ratio is tight.

1.4. **Preliminaries.** A Unit Disk Graph (UDG) is an undirected graph which has an embedding in the plane such that two vertices are connected by an edge if and only if their Euclidean distance is at most one.

Definition 1. A *Unit Line Graph (ULG)* is a unit disk graph in which all vertices have the same y-coordinate.

Let $G = (V, E)$ be an undirected graph. A set $D \subseteq V$ is called a *dominating set* if each vertex in V is either in D or adjacent to a vertex in D. A set $CD \subseteq V$ is a *connected dominating set* if it is a dominating set such that the subgraph induced by CD is connected. We call a set $I \subseteq V$ an *independent set* if it does not contain two adjacent vertices. Finally a set $VC \subseteq V$ is called a *vertex cover* if for every edge $e = (u, v)$ it holds that either $u \in VC$ or $v \in VC$.

Definition 2. For two vertices u and v let $d(u, v)$ be the hop-distance between u and v, that is the number of edges on a shortest path between these two vertices.

The hop-distance is not necessarily the geometric distance between two vertices. Denote by $N^r(v) = \{u \in V \mid d(u, v) \leq r\}$ the r-neighborhood of a vertex v. For ease of notation we define $N^0(v) := \{v\}$, $N(v) := N^1(v)$ and for a set $V' \subseteq V$ we define $N(V') = \bigcup_{v' \in V'} N(v')$. Note that $v \in N(v)$.

Denote by the *locality distance* (or short the *locality*) of an algorithm the minimum k such that the status of a vertex v (e.g. whether or not v is in a dominating or connected dominating set) depends only on the vertices in $N^k(v)$.

2. Dominating Set

We present a local approximation algorithm that computes a factor 12 approximation for dominating set on unit disk graphs. We prove its correctness, its approximation ratio and that its locality is exactly one hop. For proving that our analysis of the approximation factor is the best possible we give a tight lower bound. Then we prove lower bounds for local algorithms with arbitrary locality distance k using unit line graphs. Finally we present a local algorithm for dominating set on unit line graphs with locality one which achieves an approximation factor of 3.

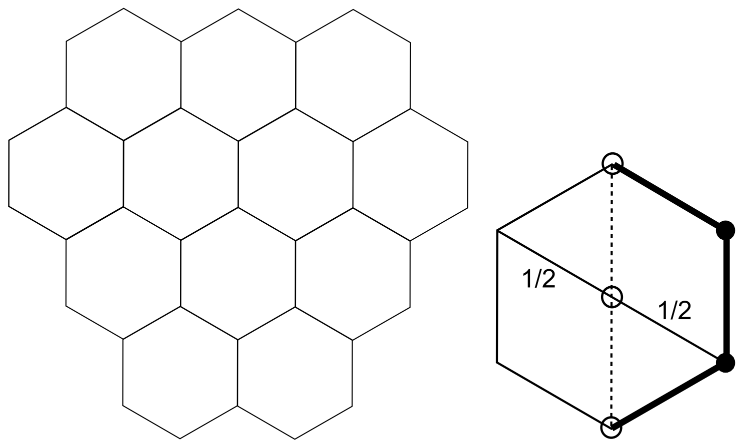
First we introduce a tiling of the plane which we are going to use.

2.1. **Tiling of the Plane.** We tile the plane in hexagons in a way that achieves the following properties:

- Each vertex is contained in exactly one hexagon
- All vertices in a hexagon are connected by an edge

We attain the above properties by tiling the plane in hexagons with a diameter of one. Ambiguities caused by vertices at the border of hexagons are resolved as shown in Figure 1 (b): The right borders excluding the upper and lower apexes belong to a hexagon, the rest of the border does not. We assume that the tiling starts with the coordinates (0,0) being in the center of a tile of class 1.

This is similar to the tiling used in [3] but in contrast to their tiling our hexagons do not have any information that distinguishes them from one another (class numbers etc.).



(a) (b)

FIGURE 1. (a) A part of the tiling of the plane into hexagons. (b) One hexagon of the tiling. The bold lines indicate the parts of its border that belong to this hexagon

2.2. **Algorithm for Unit Disk Graphs.** Now we present a local algorithm for dominating set on unit disk graphs (UDGs) with locality one. Let $G = (V, E)$ be a unit disk graph. The algorithm works as follows: In each hexagon h the vertex v which is closest to the center of h is assigned to the dominating set D. Ambiguities are resolved by e.g. choosing the vertex with the smallest x-coordinate. We refer to this as Algorithm 1.

Algorithm 1: Local algorithm for finding a dominating set in a unit disk graph

```
// Algorithm is executed independently by each node v;
// Denote by V_h all vertices in h;
Find all vertices in N(v) and compute V_h;
if v is the vertex closest to the center of h among all v' ∈ V_h then become
part of D else Do not become part of D
```

We prove the correctness of Algorithm 1 in Theorem 1.

Theorem 1. *Let G be a unit disk graph. Algorithm 1 has the following properties:*

(1) *The computed set D is a dominating set for G.*
(2) *Let D_{OPT} be an optimal dominating set. It holds that $|D| \leq 12 \cdot |D_{OPT}|$.*
(3) *Whether or not a vertex v is in D depends only on the vertices at most one hop away from v, i.e. Algorithm 1 is local.*
(4) *The processing time for a vertex v is linear in the number of vertices adjacent to v.*

Proof. We first prove that D is indeed a dominating set for G. Let h be a non-empty hexagon and denote by V_h all vertices in h. As $V_h \neq \emptyset$ it holds that one vertex $v \in V_h$ is the vertex which is closest to the center of h. So $v \in D$ and v dominates all vertices in h.

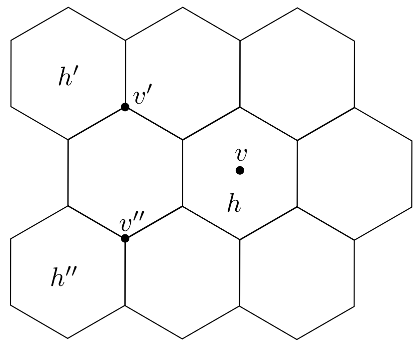


FIGURE 2. A vertex v in the center of hexagon h can only be adjacent to vertices in these 9 hexagons. Note that due to the resolving method for vertices at the border of hexagons, the vertices v' and v'' belong to the hexagons h' and h'' respectively (and are adjacent to v).

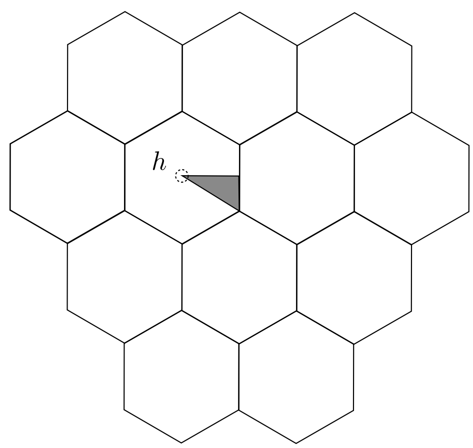


FIGURE 3. A vertex in the gray area excluding the center of h can only be adjacent to vertices in these 12 hexagons.

Now we prove that for an optimal dominating set D_{OPT} it holds that $|D| \leq 12 \cdot |D_{OPT}|$. In the following we prove that for any vertex in D_{OPT} there can exist at most 12 hexagons that have a vertex at (Euclidean) distance at most one from it. (In fact this holds for all vertices but we need the claim only for vertices in D_{OPT}.) Consider a vertex $v \in D_{OPT}$ in a hexagon h. W.l.o.g. we assume that v is in the gray area of h as shown in Figure 3 or in the center of h as shown in Figure 2. In both cases for a vertex $v' \in D$ that is adjacent to v it holds that v' must be in one of the 9 or 12 hexagon in the respective figure (for the case of v being in the center of h check the resolving method for ambiguities at the border of hexagons, Figure 1b). Since at most one vertex per hexagon was assigned to D there are at most 12 vertices in D that are adjacent to v. (Note that in the case where v is in the gray area the resolving method for ambiguities does not come into play.)

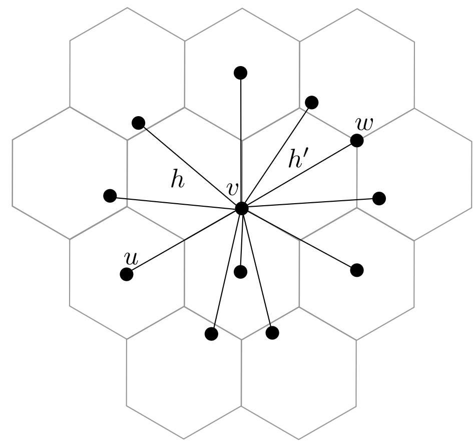


FIGURE 4. Tight example for Algorithm 1. To make the figure clearer the edges which are not adjacent to v are omitted.

So for each vertex $v \in D_{OPT}$ it holds that it is adjacent to at most 12 vertices in D. Since D_{OPT} is a dominating set all vertices in D must be adjacent to at least one vertex in D_{OPT}. This leads to $|D| \leq 12 \cdot |D_{OPT}|$.

Next we determine the locality of Algorithm 1. In the algorithm, a vertex v in a hexagon h only needs to explore all vertices which are at most one hop away from v in order to determine whether v is the vertex closest to the center of h. So whether or not $v \in D$ depends only on the vertices which are at most one hop away from v.

Now we prove the processing time of Algorithm 1. Let v be a vertex and define $n_1(v) := \left|N^1(v)\right|$. In the algorithm v needs to determine for each vertex $v' \in N^1(v)$ whether $v' \in h$ and whether v' is closer to the center of h than v. This can be done in $O\left(n_1(v)\right)$. □

2.3. **Tightness of Approximation Factor.** We give an example which shows that our analysis of Algorithm 1 is tight. Consider the graph shown in Figure 4 and denote it by G (we omitted the edges which are not adjacent to v in order to make the figure clearer). Note that the vertex v is on the crossing of three hexagons but is assigned to the hexagon h. The vertex w is on a crossing of three hexagons as well but by our the resolving method it is assigned to the hexagon h'. Also note that the vertex u is directly in the center of its hexagon. We observe that the vertex v alone is sufficient to dominate all other vertices in G. Since each hexagon contains only one vertex, Algorithm 1 assigns every vertices in the graph to D. So Algorithm 1 achieves a competitive ratio of 12 in G.

We can enlarge this construction to an arbitrary size by putting several copies of G together. This is shown in Figure 5. The copies of G are indeed connected since the length of an edge of a hexagon is 1/2 and the diameter of a hexagon is 1.

2.4. **Lower Bound.** Here we present a construction which enables us to give a lower bound for the approximation factor of a local algorithm for dominating set with a locality distance of k.

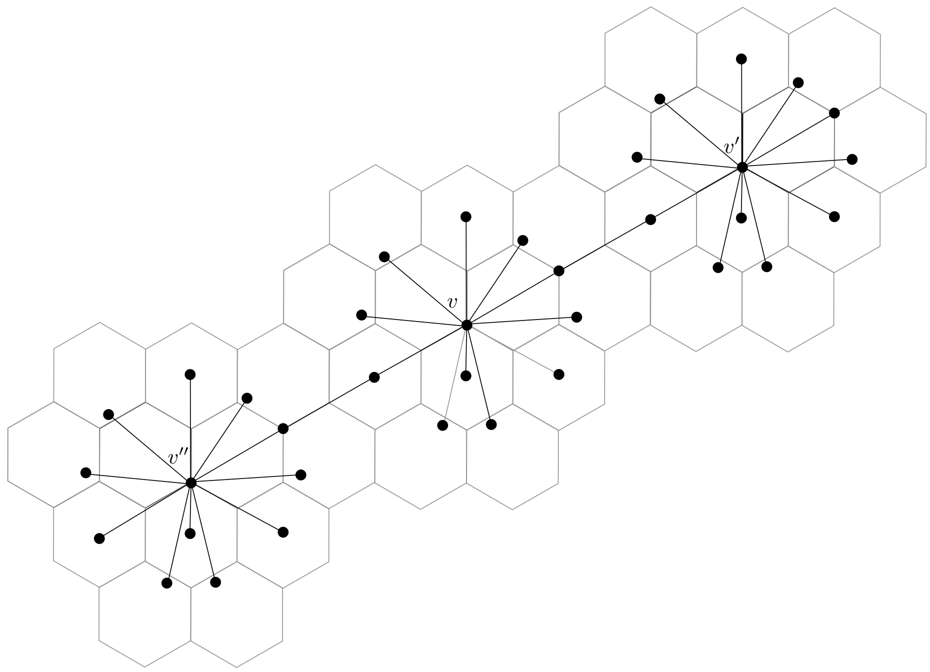


FIGURE 5. The enlargement of our construction of the tight example. It is possible to extend this to an arbitrarily large graph.

Theorem 2. *Let $\mathcal{A}$ be a local algorithm for dominating set in unit line graphs with locality distance k in the setting of location aware nodes. The best performance ratio which can be achieved by $\mathcal{A}$ is*

- $1+\frac{3}{2k+3}$ *if* $k \equiv 0 \mod 3$
- $1+\frac{3}{2k+1}$ *if* $k \equiv 1 \mod 3$
- $1+\frac{3}{2k+2}$ *if* $k \equiv 2 \mod 3$

i.e., for arbitrary k the best performance ratio is $1+\frac{3}{2k+((k-1) \mod 3)+1}$.

Proof. Consider the unit line graph $G=(V,E)$ consisting of $4+2k$ vertices (see Figure 6). Denote by V_M the four vertices in the middle. Let D be the dominating set which is computed by $\mathcal{A}$ for G. We distinguish three cases:

- $k \equiv 0$ mod 3. Then there must be at least one vertex $u \in V_M$ such that $u \in D$ (otherwise D would not be a dominating set). Consider the graph G_0 consisting of u and k vertices on the left and $k+2$ vertices on the right of u (see Figure 6). Let D_0 be the dominating set which is computed by $\mathcal{A}$ for G_0. An optimal dominating set for G_0 has $\frac{2k+3}{3}$ vertices. However, as $u \in D$ and the locality of $\mathcal{A}$ is k it follows that $u \in D_0$. Therefore it follows that $|D_0| \geq \lceil \frac{k-1}{3} \rceil + 1 + \lceil \frac{k+1}{3} \rceil = \frac{2k+6}{3}$. So the approximation ratio of $\mathcal{A}$ is a least $\frac{2k+6}{3} / \frac{2k+3}{3} = \frac{2k+6}{2k+3} = 1 + \frac{3}{2k+3}$.
- $k \equiv 1$ mod 3. We distinguish whether or not there are vertices in V_M which are not in D. If all vertices in V_M are in D then it holds that $|D| \geq \lceil \frac{k-1}{3} \rceil + 4 + \lceil \frac{k-1}{3} \rceil = \frac{2k+10}{3}$. However, an optimal dominating set for

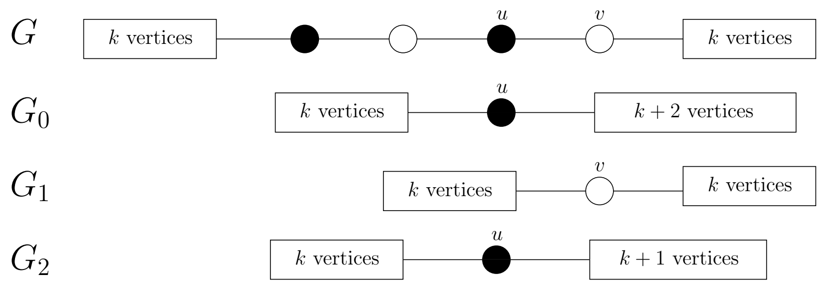


FIGURE 6. A local algorithm $\mathcal{A}$ with locality distance k has to take the same decisions for the vertices u and v in G, G_1, G_2, and G_3. The black vertices show a possible dominating set computed by $\mathcal{A}$. In order to simplify this figure the rectangular boxes represent k, $k+1$, or $k+2$ vertices on the line, respectively.

G has at most $\frac{2k+4}{3}$ vertices. So then the approximation ratio of $\mathcal{A}$ is at least $\frac{2k+10}{3} / \frac{2k+4}{3} = \frac{2k+10}{2k+4} = 1 + \frac{6}{2k+3}$.

If there is a vertex $v \in V_M$ with $v \notin D$ then we consider the graph G_1 which consists of v and k vertices on the left and on the right of v (see Figure 6). Let D_1 be the dominating set which is computed by $\mathcal{A}$ for G_1. An optimal dominating set for G_1 has $\frac{2k+1}{3}$ vertices. However, as $v \notin D$ and the locality of $\mathcal{A}$ is k it follows that $v \notin D_1$. This implies that $|D_1| \geq \lceil \frac{k-1}{3} \rceil + 2 + \lceil \frac{k-1}{3} \rceil = \frac{2k+4}{3}$. So the approximation ratio of $\mathcal{A}$ is a least $\frac{2k+4}{3} / \frac{2k+1}{3} = \frac{2k+4}{2k+1} = 1 + \frac{3}{2k+1}$. As $1 + \frac{3}{2k+1} \leq 1 + \frac{6}{2k+3}$ it follows that if $k \equiv 1 \bmod 3$ then the approximation ratio of $\mathcal{A}$ is a least $1 + \frac{3}{2k+1}$.

- $k \equiv 2 \bmod 3$. Then there must be at least one vertex $u \in V_M$ such that $u \in D$ (otherwise D would not be a dominating set). Consider the graph G_2 consisting of u and k vertices on the left and $k+1$ vertices on the right of u (see Figure 6). Let D_2 be the dominating set which is computed by $\mathcal{A}$ for G_2. An optimal dominating set for G_2 has $\frac{2k+2}{3}$ vertices. However, as $u \in D$ and the locality of $\mathcal{A}$ is k it follows that $u \in D_0$. Therefore it follows that $|D_0| \geq \lceil \frac{k-1}{3} \rceil + 1 + \lceil \frac{k}{3} \rceil = \frac{2k+5}{3}$. So the approximation ratio of $\mathcal{A}$ is a least $\frac{2k+5}{3} / \frac{2k+2}{3} = \frac{2k+5}{2k+2} = 1 + \frac{3}{2k+2}$.

□

Corollary 1. *Let $\mathcal{A}$ be a local algorithm for dominating set in unit disk graphs (UDGs) with a locality distance of k hops in the setting of location aware nodes. The best performance ratio which can be achieved by $\mathcal{A}$ is*

- $1 + \frac{3}{2k+3}$ *if* $k \equiv 0$ *mod* 3
- $1 + \frac{3}{2k+1}$ *if* $k \equiv 1$ *mod* 3
- $1 + \frac{3}{2k+2}$ *if* $k \equiv 2$ *mod* 3

2.5. **Algorithm for Unit Line Graphs.** We present a local algorithm for dominating set on unit line graphs (ULGs) with locality one. It achieves an approximation ratio of 3. The idea is to divide the line into units of width one. Then we assign one vertex from each unit to the dominating set.

Now we present the algorithm in detail. Let $G = (V, E)$ be a unit line graph. For a vertex v we denote by v_x its x-coordinate. For all integers i we define $V_i :=$

$\{v \in V | i \leq v_x < i+1\}$. We define $H := \{i | V_i \neq \emptyset\}$. For all $i \in H$ we define $v[i]$ to be the vertex with the smallest x-coordinate in V_i. We define $D := \{v[i] | i \in H\}$. We output D. We refer to this as Algorithm 2.

Algorithm 2: Local algorithm for finding a dominating set in a unit line graph

```
// Algorithm is executed independently by each node v;
// let i be the integer such that v ∈ V_i;
// let v[i] be vertex with the smallest x-coordinate in V_i;
Find all vertices in N(v) and determine V_i;
if v = v[i] then become part of the dominating set D else Do not become
part of D
```

We prove the correctness of Algorithm 2 in Theorem 3.

Theorem 3. *Let G be a unit line graph. Algorithm 2 has the following properties:*

(1) *The computed set D is a dominating set for G.*
(2) *Let D_{OPT} be an optimal dominating set. It holds that $|D| \leq 3 \cdot |D_{OPT}|$.*
(3) *Whether or not a vertex v is in D depends only on the vertices at most one hop away from v, i.e. Algorithm 2 is local.*
(4) *The processing time for a vertex v is linear in the number of vertices adjacent to v.*

Proof. We first prove that D is indeed a dominating set for G. Let $i \in H$. As $V_i \neq \emptyset$ it holds that $v[i] \in V_i \cap D$. So $v[i]$ dominates all vertices in V_i.

Now we prove that for an optimal dominating set D_{OPT} it holds that $|D| \leq 3 \cdot |D_{OPT}|$. Consider a vertex $v \in D_{OPT}$ in a set V_i. Since all sets V_j have width one it holds that v is adjacent to at most 3 vertices in D (at most one each in V_{i-1}, V_i, and V_{i+1}). Since D_{OPT} is a dominating set all vertices in D must be adjacent to at least one vertex in D_{OPT}. This leads to $|D| \leq 3 \cdot |D_{OPT}|$.

Next we determine the locality of Algorithm 2. In the algorithm, a vertex v in a set V_i only needs to explore all other vertices in V_i. (This is sufficient to determine whether v is the vertex with the smallest x-coordinate among the vertices in V_i.) So whether or not $v \in D$ depends only on the vertices which are at most one hop away from v.

Now we prove the processing time of Algorithm 2. Let v be a vertex and define $n_1(v) := \left|N^1(v)\right|$. In the algorithm v needs to determine for each vertex $v' \in N^1(v)$ whether $v' \in V_i$ and whether v' has a smaller x-coordinate than v. This can be done in time $O(n_1(v))$. □

3. Connected Dominating Set

In this section we present a local approximation algorithm that computes a factor 216 approximation for connected dominating set. We prove its correctness, its approximation ratio and its locality distance of two hops. We prove that in contrast to the other problems discussed in this paper there is no constant ratio approximation algorithm with locality one. We give lower bounds for the approximation ratio of local algorithms with arbitrary locality distance k. Finally we give an approximation algorithm for connected dominating set for unit line graphs which achieves an approximation factor of 6.

3.1. **Algorithm for Unit Disk Graphs.** We present our approximation algorithm for connected dominating set in unit disk graphs with locality two. Let $G = (V, E)$ be a connected unit disk graph. We use the same tiling of the plane as introduced in Section 2. The algorithm works as follows: Consider one hexagon h. We consider each hexagon h' such that there are pairs of adjacent vertices v and v' with $v \in h$, $v' \in h'$. We compute the two adjacent vertices $v \in h$, $v' \in h'$ which are closest to each other (ties are resolved by some some resolving method, e.g. by choosing the pair where the vertex in h has the smallest x-coordinate etc.). We assign v and v' to the connected dominating set CD. Do this for all hexagons h' with the above property.

If there are no such hexagons h' we assign the vertex $v \in h$ to CD which is closest to the center of h (ties are broken like in Algorithm 1). The above description is presented in Algorithm 3.

Algorithm 3: Local algorithm for finding a connected dominating set in a unit disk graph

```
// Algorithm is executed independently by each node v;
dominator:=false;
Find all vertices in N^2(v);
if there are hexagons h' such that there are adjacent vertices v_1 ∈ h and
v_2 ∈ h' then
    forall hexagons h' such that there are adjacent vertices v_1 ∈ h and v_2 ∈ h'
    do
        Compute pair v_1 ∈ h, v_2 ∈ h' with smallest distance between v_1 and v_2;
        if v_1 = v then
            dominator:=true;
        end
    end
end
else
    // Denote by V_h all vertices in h;
    Find all vertices in N(v) and compute V_h;
    if v is the vertex closest to the center of h among all v' ∈ V_h then
    dominator:=true
end
if dominator=true then become part of CD else Do not become part of CD
```

We prove the correctness and the other properties of Algorithm 3 in Theorem 4.

Theorem 4. *Let G be a unit disk graph. Algorithm 3 has the following properties:*

(1) *The computed set CD is a connected dominating set for G.*
(2) *Let CD_{OPT} be an optimal connected dominating set. It holds that $|CD| \leq 216 \cdot |CD_{OPT}|$.*
(3) *Whether or not a vertex v is in CD depends only on the vertices at most two hops away from v, i.e. Algorithm 3 is local.*
(4) *The processing time for a vertex v is quadratic in the number of vertices adjacent to v.*

Proof. First we prove that the set CD is indeed a connected dominating set for G. From the construction it follows that in each hexagon which contains vertices of G at least one vertex is assigned to CD. So CD is a dominating set for G. If in G a vertex $\bar{v}$ in a hexagon h is adjacent to a vertex $\bar{v}'$ in a hexagon h', then the algorithm ensures that for two adjacent vertices v and v' with $v \in h$ and $v' \in h'$ it holds that $v \in CD$ and $v' \in CD$. So in G there is an edge between two vertices in two different hexagons if and only if in G restricted to CD there is an edge between these hexagons. It follows that CD is connected since G is connected.

Now we want to prove the approximation ratio of Algorithm 3. In Algorithm 1 at most one vertex per hexagon is assigned to the connected dominating set. We showed in Theorem 1 that this ensures an approximation ratio of 12 in comparison with an optimal dominating set. Now for one hexagon h there are at most 18 hexagons h' such that there are edges between vertices in h and h'. So at most 18 vertices from h are assigned to CD. Let D_{OPT} be an optimal dominating set, let CD_{OPT} be an optimal connected dominating set and let D be the dominating set computed by Algorithm 1. It follows that

$$\begin{aligned} |CD| &\leq 18 \cdot |D| \\ &\leq 18 \cdot 12 \cdot |D_{OPT}| \\ &\leq 18 \cdot 12 \cdot |CD_{OPT}| \\ &= 216 \cdot |CD_{OPT}| \end{aligned}$$

Now we prove the locality of Algorithm 3. Consider a vertex v in a hexagon h. In order to find out whether or not v is assigned to CD we need to find out whether v is adjacent to a vertex v' in a hexagon h' with $h \neq h'$ and what other vertices $\bar{v} \in h$ are adjacent to vertices $\bar{v}' \in h'$. In order to do this, we need to explore all vertices which are at most two hops away from v. To check whether there are generally vertices in h which are adjacent to vertices in other hexagons (as if not the vertex closest to the center of h is assigned to CD), we need to explore the vertices which are at most two hops away from v.

Finally we prove the processing time of Algorithm 3. Let again v be a vertex in a hexagon h. We define $n_2(v) := \left|N^2(v)\right|$. For each vertex $v' \in N^1(v)$ the vertex v needs to determine whether $v' \in h$ and to what vertices $\bar{v}$ in other hexagons h' the vertex v' is adjacent to. For each hexagon h' such that there are adjacent vertices in h and h' we need to compute the pair of vertices $v \in h$ and $v' \in h'$ which are closest to each other and need to compute the resolving of the ties. This can be done in $O\left(n_2(v)^2\right)$. □

3.2. **No Constant Ratio Approximation Algorithm with Locality One.** We prove that there is no local algorithm for connected dominating set with locality one which achieves a constant approximation ratio.

Theorem 5. *Let $\mathcal{A}$ be a local algorithm with locality one for connected dominating set in the setting of location aware nodes. For every constant c there is a graph G_c such that $\mathcal{A}$ achieves an approximation ratio worse than c when computing a connected dominating set for G_c.*

Proof. W.l.o.g. we assume that c is an integer. Consider three circles of diameters one, three and five respectively which are placed inside of each other (see Figure 7).

Consider the unit disk graph G_c with $21 \cdot c$ vertices distributed on the inner ring. Denote these vertices by V_I. On the middle ring there are another $21 \cdot c$ vertices such that each of them is opposite to one vertex in V_I. We denote by V_M the vertices on the middle ring. As G_c is a unit disk graph an edge between two vertices exists if and only if the Euclidean distance between the vertices is at most one. For each vertex $v \in V_M$ we define the unit disk graph G_v as follows: it consists of the vertices V_I, the vertex v and a vertex v' opposite of v on the outer ring. Denote by v'' the vertex opposite of v on the inner ring. Again, an edge between two vertices exists if and only if the Euclidean distance between two vertices is at most one. Figure 7 shows this construction for $c = 1$. When $\mathcal{A}$ computes a connected dominating set CD_v for G_v it holds that $v \in CD_v$ and $v'' \in CD_v$ (as these vertices must be included in any connected dominating set for G_v). When $\mathcal{A}$ computes a connected dominating set CD for G_c it must take the same decision about assigning v'' to CD as in G_v (since the locality of $\mathcal{A}$ is one). So $v'' \in CD$. With the same argument we can show that for every vertex $u \in V_I$ it holds that $u \in CD$. So $V_I \subseteq CD$ and therefore $|CD| \geq 21 \cdot c$. But since the length of the middle ring is 3π it holds that the size of a maximal independent set is at most $\lfloor 3\pi \rfloor = 10$ and therefore the size of a connected dominating set for G_c is at most 20 (we can construct this by taking a maximal independent subset I_M of V_M and add all vertices in V_I which are opposite of vertices in I_M). Denote by CD_{OPT} an optimal dominating set for G_c. So for the approximation ratio $|CD|/\,|CD_{OPT}|$ it holds that

$$\frac{|CD|}{|CD_{OPT}|} \geq \frac{21 \cdot c}{20} > c$$

□

Corollary 2. *Let $\mathcal{A}$ be a local algorithm with locality one for connected dominating set in the setting of location aware nodes. The approximation ratio of $\mathcal{A}$ is bounded from below by $n/40$, i.e., it is in $\Theta(n)$.*

Proof. We apply the construction in the proof of Theorem 5 to obtain the family of graphs G_c. For each graph G_c the following holds: Let CD be the connected dominating set for G_c which is computed by $\mathcal{A}$. All vertices in the inner ring are assigned to CD by $\mathcal{A}$ (see the proof of Theorem 5). As half of the vertices of G_c are on the inner ring, it holds that $|CD| \geq n/2$. However, an optimal connected dominating set CD_{OPT} for G_c has at most 20 vertices, so $|CD_{OPT}| \leq 20$. So for the approximation ratio $|CD| \,/\, |CD_{OPT}|$ it holds that

$$\frac{|CD|}{|CD_{OPT}|} \geq \frac{n}{2} \cdot \frac{1}{20} = \frac{n}{40}$$

□

3.3. **Lower Bound.** We prove a lower bound for the approximation factor of local algorithms for connected dominating set in the setting of location aware nodes depending on the locality distance of the algorithm.

Theorem 6. *Let $\mathcal{A}$ be a local algorithm for connected dominating set in the setting of location aware nodes. Let k be the locality distance of $\mathcal{A}$. Then the approximation ratio which $\mathcal{A}$ achieves is bounded from below by $1 + 1/k$.*

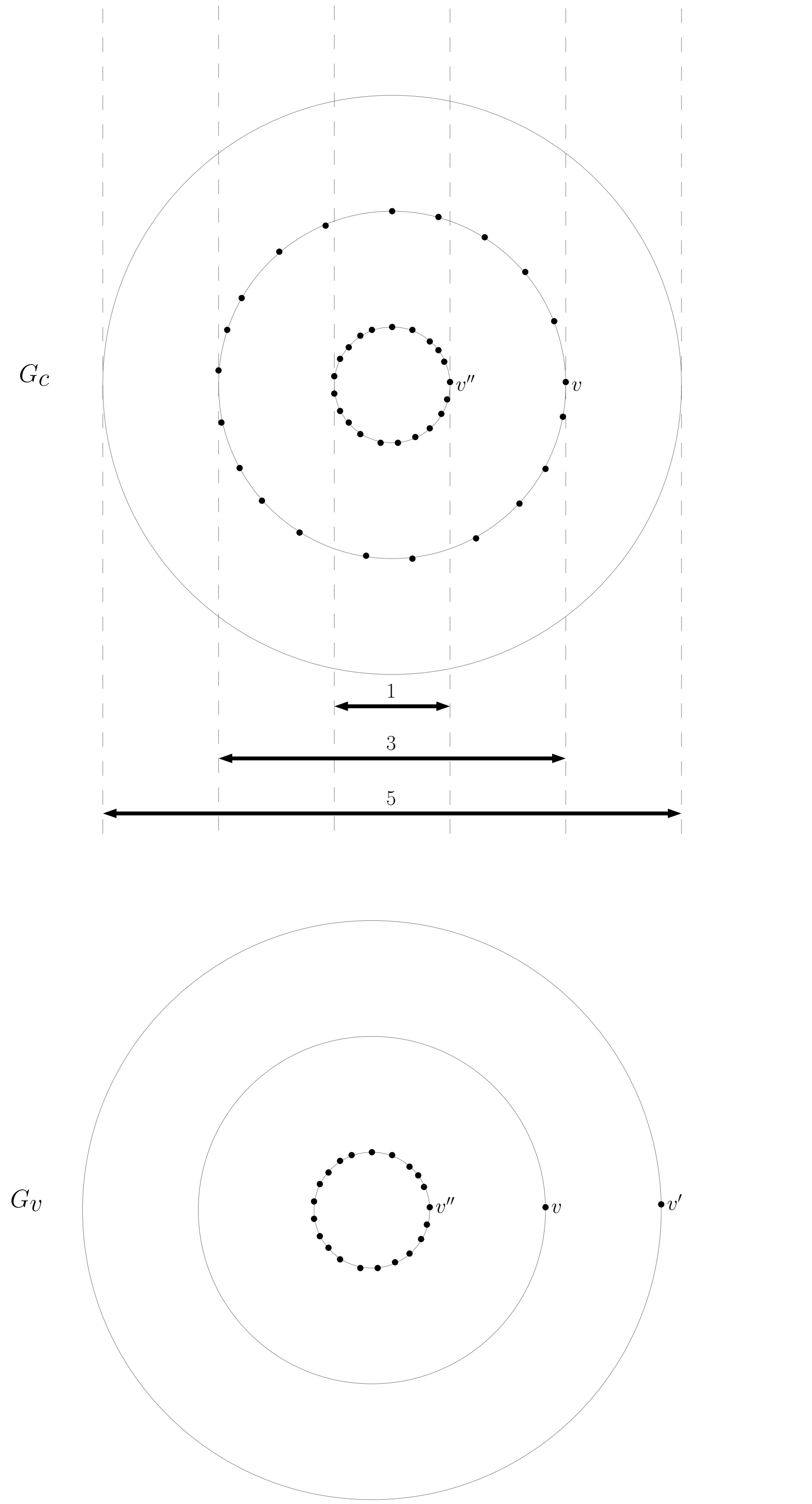


FIGURE 7. The graph G_c for $c = 1$ and the graph G_v for the vertex v. In order to make the figures clearer we omit the edges of the graphs.

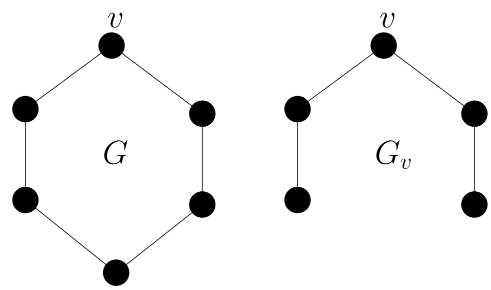


FIGURE 8. Construction for the lower bound for a local algorithm for connected dominating set for $k = 2$.

Proof. Similar to the proof of Theorem 2 we construct graphs in which $\mathcal{A}$ has to take the same decisions in a certain vertex v. Our construction consists of a cycle of vertices with $2k+2$ vertices as shown in Figure 8. Denote this cycle by G. Note that G is indeed a unit disk graph. Denote by CD the connected dominating set which $\mathcal{A}$ computes for G. Now let v be a vertex. Denote by G_v the graph which we obtain by removing the opposite vertex of v from G (see Figure 8). As $\mathcal{A}$ is a local algorithm with locality distance k it must take the same decision about assigning v to the connected dominating set in G and G_v. Denote by CD_v the connected dominating set which is computed for G_v by $\mathcal{A}$. As not assigning v to CD_v would result in CD_v being disconnected, we conclude that $v \in CD_v$ and $v \in CD$. With the same reasoning we can show that all vertices in G must be in CD. So $|CD| = 2k+2$. But an optimal connected dominating set for G has only $2k$ vertices. So the performance ratio of $\mathcal{A}$ is bounded from below by $\frac{2k+2}{2k} = 1 + \frac{1}{k}$. □

Corollary 3. *Let $\mathcal{A}$ be a local algorithm for connected dominating set in the setting of location aware nodes with locality distance* 2. *Then the approximation ratio which $\mathcal{A}$ achieves is bounded from below by* 1.5.

3.4. **Algorithm for Unit Line Graphs.** We present a local algorithm for connected dominating set on unit line graphs with locality distance one. It is an extension of the algorithm for dominating set presented in this paper. The approximation ratio of this algorithm is 6.

Now we present the algorithm in detail. Let $G = (V, E)$ be a unit line graph. For a vertex v we denote by v_x its x-coordinate. As in Algorithm 2 for all integers i we define $V_i := \{v \in V | i \leq v_x < i+1\}$ and $H := \{i | V_i \neq \emptyset\}$. For all $i \in H$ we define $v[i]$ and $v'[i]$ to be the vertices with the smallest and largest x-coordinates in V_i. We define $CD := \{v[i], v'[i] | i \in H\}$. We output CD. We refer to this as Algorithm 4.

We prove the correctness of Algorithm 1 in Theorem 1.

Theorem 7. *Let G be a unit line graph. Algorithm 4 has the following properties:*

(1) *The computed set CD is a connected dominating set for G.*
(2) *Let CD_{OPT} be an optimal connected dominating set. It holds that $|D| \leq 6 \cdot |D_{OPT}|$.*
(3) *Whether or not a vertex v is in CD depends only on the vertices at most one hop away from v, i.e. Algorithm 4 is local.*

Algorithm 4: Local algorithm for finding a connected dominating set in a unit line graph

// Algorithm is executed independently by each node v;
// let i be the integer such that $v \in V_i$;
Find all vertices in $N(v)$ and determine V_i;
if $v = v[i]$ *or* $v = v'[i]$ **then** become part of the dominating set CD **else** Do not become part of CD

(4) *The processing time for a vertex v is linear in the number of vertices adjacent to v.*

Proof. We first prove that CD is indeed a connected dominating set for G. Let $i \in H$. Let D be the dominating set which Algorithm 2 would have computed for G. As $D \subseteq CD$ it remains to prove that CD is connected. For all i it holds that $v[i]$ and $v'[i]$ are connected by an edge (assuming $v[i] \neq v'[i]$). Let i be an integer such that $i \in H$ and $i+1 \in H$. Then there are edges connecting $v'[i]$ and $v[i+1]$ since $v'[i]$ is the vertex on the very right of V_i and $v[i+1]$ is the vertex on the very left of V_{i+1} and G is a connected unit line graph. This proves that CD is connected.

Now we prove that for an optimal connecting dominating set CD_{OPT} it holds that $|CD| \leq 6 \cdot |CD_{OPT}|$. Let D again be the dominating set which Algorithm 2 would have computed for G. From our construction it follows that $|CD| \leq 2 \cdot |D|$. As $|D_{OPT}| \leq |CD_{OPT}|$ it follows that

$$\begin{aligned} |CD| &\leq 2 \cdot |D| \\ &\leq 2 \cdot 3 \cdot |D_{OPT}| \\ &\leq 6 \cdot |CD_{OPT}| \end{aligned}$$

Next we determine the locality of Algorithm 4. In the algorithm, a vertex v in a set V_i only needs to explore all other vertices in V_i. (This is sufficient to determine whether v is the vertex with the smallest x-coordinate or with the largest x-coordinate among the vertices in V_i.) So whether or not $v \in CD$ depends only on the vertices which are at most one hop away from v.

Now we prove the processing time of Algorithm 4. Let v be a vertex and define $n_1(v) := \left|N^1(v)\right|$. In the algorithm v needs to determine for each vertex $v' \in N^1(v)$ whether $v' \in V_i$ and whether v' has a smaller or a larger x-coordinate than v. This can be done in time $O(n_1(v))$. □

4. Independent Set

In this section we present our local approximation algorithm for independent set in unit disk graphs with locality one. This algorithm achieves an approximation ratio of 1/1801. Then we give a lower bound for the approximation ratio of local algorithms with arbitrary locality distance k. For this proof we employ a construction consisting of unit line graphs. Finally we give a local algorithm for unit line graphs with locality one. Its approximation ratio is almost 1/2.

First we introduce another tiling of the plane which is a bit more complex than the tiling presented in Section 2.

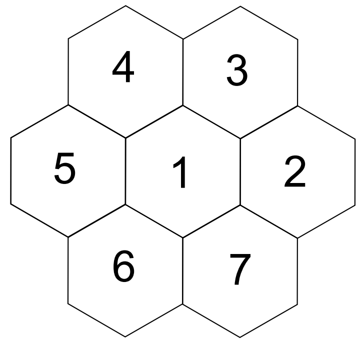


FIGURE 9. One tile consisting of 7 hexagons

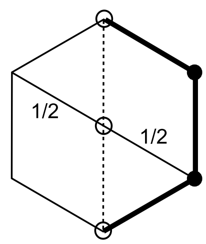


FIGURE 10. Resolving ambiguities at the border of hexagons

4.1. **Tiling of the Plane.** We present the tiling of the plane which our local algorithm for independent set in UDGs will use. Let $G = (V, E)$ be a connected unit disk graph. We tile the plane with tiles where each tile consists of seven hexagons arranged as in Figure 9. The diameter of each hexagon is 1. Each hexagon is assigned a class number between 1 and 7. This assignment is done identically in each tile. For a hexagon h we denote by $class(h)$ its class number. Each vertex is assigned to a hexagon, ambiguities caused by vertices at the border of hexagons are resolved as shown in Figure 10 (this is the same resolving method which was used in Section **??**). The plainer is tiled with the tiles mentioned above as shown in Figure 11. We conclude with the following proposition.

Proposition 1. *Two different hexagons with the same class number have an Euclidean distance of strictly more than 1.*

4.2. **Algorithm for Unit Disk Graphs.** We present our local approximation algorithm for independent set in unit disk graphs. Let $G = (V, E)$ be a unit disk graph. The main idea of the algorithm is that since the vertices in one hexagon form a clique at most one vertex per hexagon can be in an independent set. The algorithm works as follows: In each hexagon h we define the vertex which is closest to the center to be the head vertex v_h (ambiguities are resolved by choosing the vertex with the smallest x-coordinate among the vertices with the same distance to the center and if this is ambiguous we choose the vertex with the smallest y-coordinate). If v_h is *not* adjacent to a vertex in a hexagon with a lower class number we assign v_h to the independent set I.

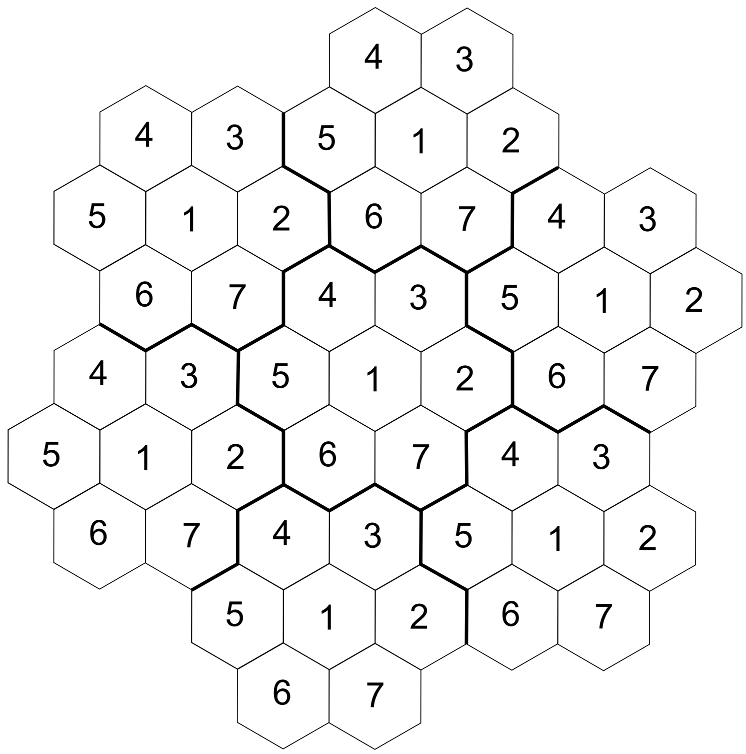


FIGURE 11. Several tiles arranged together

Algorithm 5: Local algorithm for finding an independent set in a unit disk graph

---

// Algorithm is executed independently by each node v;
Find all vertices in $N(v)$;
Determine the hexagon h of v and $class(h)$;
if *v is the head vertex of h* **then**
    if *v is NOT adjacent to a vertex in a hexagon h' with $class(h') < class(h)$* **then**
        Assign v to the independent set I;
    end
end

We prove the correctness of Algorithm 5, its approximation factor, its locality and its processing time in Theorem 8.

Theorem 8. *Let G be a unit disk graph. Algorithm 5 has the following properties:*

(1) *The computed set I is an independent set for G.*
(2) *Let I_{OPT} be an optimal independent set. It holds that $|I| \geq \frac{1}{1801} \cdot |I_{OPT}|$.*
(3) *Whether or not a vertex v is in I depends only on the vertices at most one hop away from v, i.e. Algorithm 5 is local.*

(4) *The processing time for a vertex v is linear in the number of vertices adjacent to v.*

Proof. First we want to show that I is indeed an independent set for G. Assume on the contrary that there are two adjacent vertices $v \in I$ and $v' \in I$. Since at most one vertex per hexagon is assigned to I it follows that v and v' are in different hexagons h and h' respectively. From Proposition 1 it follows that $class(h) \neq class(h')$, so w.l.o.g. we can assume that $class(h) > class(h')$. But from the construction it follows that a vertex v in a hexagon h is not assigned to I if it is adjacent to a vertex v' in a hexagon h' with $class(h) > class(h')$. This is a contradiction. So the set I is an independent set for G.

Now we want to prove the approximation ratio of Algorithm 5. Let I_{OPT} be an optimal independent set for G. Denote by H the set of hexagons which contain vertices from G. Since the vertices in a hexagon form a clique, at most one vertex per hexagon can be part of an independent set. So $|H|$ is an upper bound for $|I_{OPT}|$. For a hexagon h we denote by $B(h)$ the union of h and all hexagons h' such that $class(h) < class(h')$ and h' contains a vertex which is adjacent to a vertex in h. For a set of hexagons H' we define $B(H') := \bigcup_{h' \in H'} B(h')$. We define recursively $B^0(h) := \{h\}$ and $B^i(h) := B(B^{i-1}(h))$. As there are only seven different classes of hexagons, it follows that $B^{7-i}(h) = B^{7-i+k}(h)$ for hexagons of class i and for all $k \geq 0$. We define $\tilde{B}(h) := B^{7-class(h)}(h)$. Intuitively we can say that the hexagon h blocks all hexagons in $\tilde{B}(h)$.

Consider a graph $G' = (V', E')$ with one vertex per hexagon in H and with an edge between two vertices if and only if their respective hexagons are adjacent. For two hexagons h,h' we denote by $d(h, h')$ the distance between their respective vertices in G'. For a hexagon h we get that for all hexagons $h' \in B(h)$ it holds that $d(h, h') \leq 2$. So for all hexagons $h' \in \tilde{B}(h)$ it holds that $d(h, h') \leq 2 \cdot (7 - class(h)) \leq 12$. As for a hexagon h there are at most $3(2k)^2 + 3(2k) + 1$ hexagons h' such that $d(h, h') \leq k$ (see the proof of Lemma **??**) we conclude that $\left|\tilde{B}(h)\right| \leq 3 \cdot 24^2 + 3 \cdot 24 + 1 = 1801$. It follows that $1801|I| \geq \bigcup_{v' \in I} \left|\tilde{B}(h(v'))\right|$.

For a vertex v let $h(v)$ be the hexagon which contains v. From the definition of the algorithm we get the following: Let v be a head vertex. If for all vertices $v' \in I$ it holds that $h(v) \notin \tilde{B}(h(v'))$ then it holds that $v \in I$. So $\bigcup_{v' \in I} \tilde{B}(h(v')) = H$. We conclude that

$$\begin{aligned}
|I| &\geq \frac{1}{1801} \bigcup_{v' \in I} \left|\tilde{B}(h(v'))\right| \\
&\geq \frac{1}{1801} \left|\bigcup_{v' \in I} \tilde{B}(h(v'))\right| \\
&= \frac{1}{1801} |H| \\
&\geq \frac{1}{1801} |I_{OPT}|
\end{aligned}$$

Now we want to prove the locality of Algorithm 5. Since a vertex only needs to determine whether it is the head vertex of its hexagon and whether it is adjacent to a vertex in a hexagon with lower class number, the locality distance of the algorithm is one hop.

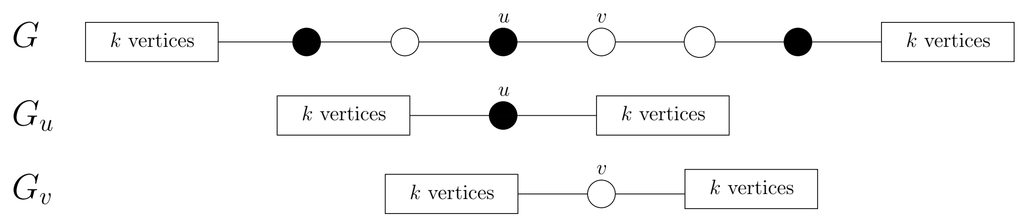


FIGURE 12. A local algorithm $\mathcal{A}$ with locality distance k has to take the same decisions for the vertices u and v in G_1, G_u and G_v. The black vertices show a possible independent set computed by $\mathcal{A}$. In order to simplify this figure the rectangular boxes represent k vertices on the line.

For the proof of the processing time let v be a vertex in a hexagon h. For each vertex $v' \in N(v)$ the algorithm needs to compute the class number of the hexagon containing v' and its distance to the center of h. This can be done in $O(|N(v)|)$. □

4.3. **Lower Bound.** Similarly to the lower bound presented in Section 2.4 we present a construction which enables us to give a lower bound for the approximation factor of local algorithms for independent set with a locality distance of k hops.

Theorem 9. *Let $\mathcal{A}$ be a local algorithm for independent set in unit line graphs with a locality distance of k hops in the setting of location aware nodes. The best performance ratio which can be achieved by $\mathcal{A}$ is $\frac{k}{k+1}$.*

Proof. Consider the unit line graph $G_1 = (V, E)$ consisting of $6 + 2k$ vertices (see Figure 12). Let I_1 be the independent set which is computed for G_1 by $\mathcal{A}$. Note that an optimal independent set for G_1 has $k + 3$ vertices. Let V_1 be the set of vertices in G_1 which are at most $k-1$ hops away from the very right or the very left vertex (in Figure 12 these are the vertices in the box with the label "k vertices"). If $k = 0$ we define $V_1 := \emptyset$. Let $V_2 := V \setminus V_1$ (thus $|V_2| = 6$). We distinguish three cases:

Case 1: There is no vertex $u \in V_2$ such that $u \in I_1$. If $k = 0$ then $|I_1| = 0$ and thus the performance ratio of $\mathcal{A}$ is 0. If $k \geq 1$ then $|I_1| \leq k + 1$ and so the performance ratio of $\mathcal{A}$ is at most $\frac{k+1}{k+3} \leq \frac{k}{k+1}$.

Case 2: There is no vertex $v \in V_2$ such that $v \notin I_1$. Then I_1 is not an independent set which is a contradiction.

Case 3: There is a vertex $u \in V_2$ with $u \in I_1$ and a vertex $v \in V_2$ with $v \notin I_1$. Now consider the graphs G_u consisting of u and v respectively and k nodes each to their left and their right (see Figure 12). Let I_u and I_v be the independent sets computed for G_u and G_v by $\mathcal{A}$. As the locality distance of $\mathcal{A}$ is k and $u \in I_1$ and $v \notin I_1$, it holds that $u \in I_u$ and $v \notin I_v$. If k is even then $|I_v| \leq k$ whereas the optimal independent set for I_v has $k+1$ vertices. So the performance ratio of $\mathcal{A}$ is at most $\frac{k}{k+1}$. If k is odd then $|I_u| \leq k$ whereas the optimal independent set for I_u has $k+1$ vertices. So the performance ratio of $\mathcal{A}$ is at most $\frac{k}{k+1}$. □

Corollary 4. *Let $\mathcal{A}$ be a local algorithm for independent set in unit disk graphs (UDGs) with a locality distance of k hops in the setting of location aware nodes. The best performance ratio which $\mathcal{A}$ can achieve is $\frac{k}{k+1}$.*

4.4. **Algorithm for Unit Line Graphs.** We present a local algorithm for independent set with locality one on unit line graphs (ULGs). For the set I computed by this algorithm it holds that $|I| \geq \left\lfloor \frac{1}{2} \cdot |I_{OPT}| \right\rfloor$ and $|I| \geq 1$ (with I_{OPT} being an optimal independent set). Like in Algorithm 2 we divide the line into units of width one.

Now we present the algorithm in detail. Let $G = (V, E)$ be a unit line graph. For a vertex v we denote by v_x its x-coordinate. For all integers i we define $V_i := \{v \in V | i \leq v_x < i + 1\}$. We define $H := \{i | V_i \neq \emptyset\}$. For all $i \in H$ we define $v[i]$ to be the vertex with the smallest x-coordinate in V_i. Now we partition the vertices $v[i]$ into two sets V_1 and V_2. Let $V_1 := \{v[i] | i \in H \wedge i \text{ is odd}\}$ and $V_2 := \{v[i] | i \in H \wedge i \text{ is even}\}$. Denote by $V_2' \subseteq V_2$ the subset of V_2 consisting of the vertices $v[i]$ which are not adjacent to vertices v' with $v' \in V_{i-1}$ or $v' \in V_{i+1}$ (so for such a vertex v' it would hold that $v'_x < i$ or $v'_x \geq 1$). We define the independent set I as $I := V_1 \cup V_2'$. The output is the set I. We refer to this as Algorithm 6.

Algorithm 6: Local algorithm for finding an independent set in a unit line graph

---

// Algorithm is executed independently by each node v;
// let i be the integer such that $v \in V_i$;
// let $v[i]$ be vertex with the smallest x-coordinate in V_i;
Find all vertices in $N(v)$ and determine V_i;
if $v = v[i]$ **then**
    if *i is odd* **then** become part of the independent set I **else if** *i is even and v is not adjacent to a vertex v' with $v'_x < i$ or $v'_x \geq i + 1$* **then**
        become part of I
    end
    else Do not become part of I
end
else Do not become part of I

We prove the correctness of Algorithm 6, its approximation factor, its locality and its processing time in Theorem 10.

Theorem 10. *Let G be a unit line graph. Algorithm 6 has the following properties:*

(1) *The computed set I is an independent set for G.*
(2) *Let I_{OPT} be an optimal independent set. It holds that $|I| \geq \left\lfloor \frac{1}{2} \cdot |I_{OPT}| \right\rfloor$ and $|I| \geq 1$.*
(3) *Whether or not a vertex v is in I depends only on the vertices which are at most k hops away from v, i.e. Algorithm 6 is local.*
(4) *The processing time for a vertex v is constant.*

Proof. It holds that the Euclidean distance between two vertices in V_1 is strictly greater than one. So V_1 is an independent set. Since all vertices in V_2' are not adjacent to vertices v' with $v'_x < i$ or $v'_x \geq i + 1$ it holds that a vertex in V_2' cannot be adjacent to a vertex in V_1. Like for V_1 it holds that the Euclidean distance between two vertices in V_2 is strictly greater than one. So no two vertices in V_2' are adjacent. All this implies that $I = V_1 \cup V_2'$ is in independent set.

Now we want to prove the approximation ratio of Algorithm 6. Let i be an integer. The vertices in V_i form a clique. Therefore at most one vertex in V_i can be in I. So $|H|$ is an upper bound for $|I_{OPT}|$. If $|H|$ is even then $|V_1| = \frac{|H|}{2}$. If $|H|$ is odd then either $|V_1| = \frac{|H|+1}{2}$ or $|V_1| = \frac{|H|-1}{2}$. So in either case it holds that $|I| \geq |V_1| \geq \lfloor \frac{1}{2} \cdot |I_{OPT}| \rfloor$. If $|V_1| \geq 1$ it clearly holds that $|I| \geq 1$. If $|V_1| = 0$ there must be one set V_i with $V_i \neq \emptyset$ and i being even (since otherwise the graph would be empty and for the empty graph the theorem is trivially true). It also holds that $V_{i-1} = \emptyset$ and $V_{i+1} = \emptyset$. So $v[i]$ is not adjacent to any vertices in V_{i-1} or V_{i+1} and therefore $v[i] \in I$. This implies that $|I| \geq 1$.

Let $v \in V_i$ be a vertex. In order to determine whether $v \in I$ we need to explore the vertices in V_i and we need to determine whether v is adjacent to a vertex v' with $v' \in V_{i-1}$ or $v' \in V_{i+1}$. All this requires only the knowledge of the vertices in $N(v)$. So the locality distance of Algorithm 6 is one. Computing whether $v = v[i]$ and whether v is adjacent to a vertex v' with $v' \in V_{i-1}$ or $v' \in V_{i+1}$ can be done in processing time linear in $|N(v)|$. □

5. Vertex Cover

In this section we prove that if we take all vertices in a unit disk graph this forms a factor 12 approximation for the minimum vertex cover problem. This leads to a local approximation algorithm with a locality of one hop. Then we present lower bounds for the approximation ratio of local algorithms for vertex cover with arbitrary locality distance k. Finally we give a factor 2 approximation algorithm for vertex cover on unit line graphs.

5.1. Factor 12 Upper Bound.

We consider a connected unit disk graph with at least two vertices. We prove an upper bound of 12 for the number of all vertices in comparison with the number of vertices in a minimum vertex cover.

Theorem 11. *Let $G = (V, E)$ be a connected unit disk graph with $|V| \geq 2$ and let VC_{OPT} be a minimum vertex cover. It holds that*

$$|V| \leq 12 \cdot |VC_{OPT}|$$

Proof. We partition the vertices V into two sets V_1 and V_2. We prove that $|V_1| \leq 2 \cdot |V_{OPT}|$ and $|V_2| \leq 10 \cdot |VC_{OPT}|$. As $|V| = |V_1| + |V_2|$ it follows then that $|V| \leq 12 \cdot |VC_{OPT}|$.

First we define the set V_1. Let $M \subseteq E$ be a maximal matching (i.e. a matching which cannot be extended by adding another edge to M). We define $V_1 := \{u, v | (u, v) \in M\}$. As M is a matching it follows that $|V_1| \leq 2 \cdot |VC_{OPT}|$.

Now we define $V_2 := V \setminus V_1$. Since M is a maximal matching it follows that V_2 does not contain any adjacent vertices. Since G is a unit disk graph and therefore does not contain a $K_{1,6}$ it follows that every vertex $v \in V_1$ is adjacent to at most 5 vertices in V_2. Since $|V| \geq 2$ it follows that $|V_1| \geq 1$ and so $|V_2| \leq 5 \cdot |V_1| \leq 10 \cdot |VC_{OPT}|$. So we conclude with $|V| = |V_1| + |V_2| \leq 2 \cdot |VC_{OPT}| + 10 \cdot |VC_{OPT}| \leq 12 \cdot |VC_{OPT}|$. □

Corollary 5. *Let $G = (V, E)$ be a graph such that there is no set of vertices V' such that the subgraph induced by V' is a $K_{1,m}$. Then it holds that*

$$|V| \leq 2m \cdot |VC_{OPT}|$$

5.2. **Algorithm for Unit Disk Graphs.** We present a local approximation algorithm for minimum vertex cover for unit disk graphs. Let $G = (V, E)$ be a connected unit disk graph. The algorithm works as follows: First we check whether $|V| \geq 2$. If $|V| = 1$ then there are no edges in G that need to be covered so we define $VC := \emptyset$. If $|V| \geq 2$ we assign all vertices to the set VC. We output VC. We present the above description as Algorithm 7.

Algorithm 7: Local algorithm for computing a vertex cover for a unit disk graph $G = (V, E)$

```
// Algorithm is executed independently by each node v;
if |V| ≥ 2 then assign v to VC;
else Do not assign v to VC;
```

We prove the correctness of Algorithm 7 in Theorem 12.

Theorem 12. *Let G be a unit disk graph. Algorithm 7 has the following properties:*

(1) *The computed set VC is a vertex cover for G.*
(2) *Let VC_{OPT} be an optimal vertex cover. It holds that $|VC| \leq 12 \cdot |VC_{OPT}|$.*
(3) *Whether or not a vertex v is in VC depends only on the vertices at most one hop away from v, i.e. Algorithm 1 is local.*
(4) *The processing time for a vertex v is constant.*

Proof. If $|V| < 2$ this implies that G contains no edges and $VC = \emptyset$ and $VC_{OPT} = \emptyset$. So in this case VC is indeed a vertex cover for G and $|VC| \leq 12 \cdot |VC_{OPT}|$. If $|V| \geq 2$ then $VC = V$. So VC is indeed a valid vertex cover for G. The approximation factor follows from Theorem 11.

In order to check whether $|V| \geq 2$ we only need to examine if there are vertices which are at most one hop away from a given vertex v. So the locality of Algorithm 7 is one hop.

Checking whether there are vertices which are adjacent to v and assigning v to VC can be done in constant time, so the processing time of Algorithm 7 is constant. □

5.3. **Lower Bound.** We prove a lower bound for the approximation ratio of a local algorithm for vertex cover with locality distance k. The proof is very similar to the proof given for independent set in Section 4.

Theorem 13. *Let k be an integer with $k \geq 1$. Let $\mathcal{A}$ be a local algorithm for vertex cover in unit line graphs with a locality distance of k hops in the setting of location aware nodes. The best performance ratio which can be achieved by $\mathcal{A}$ is $\frac{k+1}{k}$.*

Proof. Consider the unit line graph $G_1 = (V, E)$ consisting of $6 + 2k$ vertices (see Figure 13). Let VC_1 be the vertex cover which is computed for G_1 by $\mathcal{A}$. Note that an optimal vertex cover for G_1 has $k + 3$ vertices. Let V_1 be the set of vertices in G_1 which are at most $k - 1$ hops away from the very right or the very left vertex (in Figure 13 these are the vertices in the box with the label "k vertices"). If $k = 0$ we define $V_1 := \emptyset$. Let $V_2 := V \setminus V_1$ (thus $|V_2| = 6$). We distinguish three cases:

Case 1: There is no vertex $u \in V_2$ such that $u \notin VC_1$. So $|VC| \geq 6 + k - 1$ and thus the performance ratio of $\mathcal{A}$ is at most $\frac{k+5}{k+3} \geq \frac{k+1}{k}$.

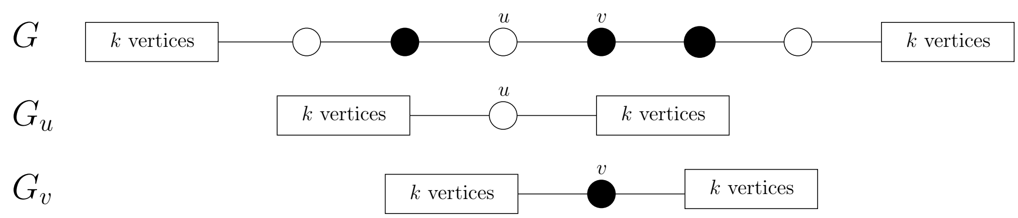


FIGURE 13. A local algorithm $\mathcal{A}$ with locality distance k has to take the same decisions for the vertices u and v in G_1, G_u and G_v. The black vertices show a possible vertex cover computed by $\mathcal{A}$. In order to simplify the figure the rectangular boxes represent k vertices on the line.

Case 2: There is no vertex $v \in V_2$ such that $v \in VC_1$. Then VC_1 is not a vertex cover which is a contradiction.

Case 3: There is a vertex $u \in V_2$ with $u \notin VC_1$ and a vertex $v \in V_2$ with $v \in VC_1$. Now consider the graphs G_u consisting of u and v respectively and k nodes each to their left and their right (see Figure 13). Let VC_u and VC_v be the vertex covers computed for G_u and G_v by $\mathcal{A}$. As the locality distance of $\mathcal{A}$ is k and $u \notin VC_1$ and $v \in VC_1$, it holds that $u \notin VC_u$ and $v \in VC_v$. If k is even then $|VC_v| \geq k+1$ whereas the optimal vertex cover for VC_v has only k vertices. So the performance ratio of $\mathcal{A}$ is at most $\frac{k+1}{k}$. If k is odd then $|VC_u| \geq k+1$ whereas the optimal independent set for VC_u has k vertices. So the performance ratio of $\mathcal{A}$ is at most $\frac{k+1}{k}$. □

Corollary 6. *Let k be an integer with $k \geq 1$. Let $\mathcal{A}$ be a local algorithm for vertex cover in unit disk graphs (UDGs) with a locality distance of k hops in the setting of location aware nodes. The best performance ratio which $\mathcal{A}$ can achieve is $\frac{k+1}{k}$.*

5.4. **Algorithm for Unit Line Graphs.** We present a local algorithm with locality one for vertex cover on unit line graphs (ULGs). It achieves an approximation ratio of 2. The algorithm works as follows: Let $G = (V, E)$ be a unit line graph. Let v_L be the leftmost vertex in V. Define $VC := V \setminus \{v_L\}$. We output VC. We refer to this as Algorithm 8.

Algorithm 8: Local algorithm for computing a vertex cover in a unit line graph

---

// Algorithm is executed independently by each node v;
// let v_L be the leftmost vertex of G;
Explore the vertices in $N(v)$;
if $v = v_L$ **then** become part of the vertex cover VC **else** do *not* become part of VC

We prove the correctness of Algorithm 8 in Theorem 14.

Theorem 14. *Let G be a unit line graph. Algorithm 8 has the following properties:*

(1) *The computed set VC is a vertex cover for G.*
(2) *Let VC_{OPT} be an optimal dominating set. It holds that $|VC| \leq 2 \cdot |VC_{OPT}|$.*

(3) *Whether or not a vertex v is in VC depends only on the vertices which are at most one hop away from v, i.e. Algorithm 8 is local.*
(4) *The processing time for a vertex v is linear in the number of vertices adjacent to v.*

Proof. Since $\{v_L\}$ is an independent set for G it holds that $VC = V \setminus \{v_L\}$ is a valid vertex cover for G. For proving the approximation ratio of Algorithm 8 we distinguish between $|V|$ being even or odd. If $|V|$ is even then there is a maximal matching for G such that every vertex is adjacent to a matching edge: Denote by $v_1, v_2, ..., v_n$ the vertices V ordered by their x-coordinate. The matching $M := \{(v_{2k-1}, v_{2k}) | 1 \leq k \leq |V|/2\}$ is such a maximal matching. So it holds that $|V| \leq 2 \cdot |VC_{OPT}|$ and therefore $|VC| \leq 2 \cdot |VC_{OPT}|$. If $|V|$ is odd then there is a maximal matching such that every vertex but v_L is adjacent to a matching edge: the matching $M' := \left\{(v_{2k}, v_{2k+1}) | 1 \leq k \leq \frac{|V|-1}{2}\right\}$ is such a maximal matching. So it follows that $|VC| \leq 2 \cdot |VC_{OPT}|$. If a vertex v is in VC depends only on the vertices which are one hop away from v (the set $N(v)$) since it is sufficient for v to explore $N(v)$ in order to determine whether $v = v_L$. During the computation for one vertex v we only need to check for every vertex $v' \in N(v)$ whether $v_x \leq v'_x$. So the processing time for a vertex v is linear in the number of vertices in $N(v)$. □

6. Conclusion

We studied the impact of locality in algorithms for dominating and connected dominating set, independent set and vertex cover. We proved the first ever lower bounds for local approximation algorithms for these problems in the setting of location aware unit disk graphs. The bounds depend on the locality distance of an algorithm. We investigated the computational power of local algorithms with very low localities. We showed that for dominating set, independent set and vertex cover a locality distance of one hop is sufficient to guarantee a constant approximation ratio. For connected dominating set we proved that there is no constant ratio approximation algorithm with locality one. However, we gave such an algorithm with locality two.

The gaps between our lower bounds and our algorithms (for locality distances one and two respectively) are significant. It remains an open problem to tighten these bounds. A first step towards this aim would be to find tight bounds for the special case of unit line graphs. In our proofs of the lower bounds we mostly used constructions with unit line graphs. Employing constructions of unit disk graphs in two dimensions one might be able to prove better lower bounds of algorithms for (general) unit disk graphs. This remains an open problem. Possibly one can get more insights about this issue by studying the problems on a grid with integer coordinates and vertices with an arbitrary transmission radius r. This setting approximates unit disk graphs while all lower bounds for it hold for general unit disk graphs as well.

Also of interest would be other local algorithms with small locality distances $k > 1$. Maybe one can improve our approximation ratios considerably by enlarging the locality by only a little. A vast open problem is to find tight bounds for the approximation ratios of local algorithms with arbitrary locality distances k.

References

[1] R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–45, 1985.

[2] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.

[3] J. Czyzowicz, S. Dobrev, T. Fevens, H. González-Aguilar, E. Kranakis, J. Opatrny, and J. Urrutia. Local algorithms for dominating and connected dominating sets of unit disc graphs with location aware nodes. In *Proceedings of LATIN 2008*. LNCS, 2008. to appear.

[4] S. O. Dulman, L. v. Hoesel, T. Nieberg, and P. Havinga. Collaborative communication protocols for wireless sensor networks, 2003.

[5] B. Gfeller and E. Vicari. A faster distributed approximation scheme for the connected dominating set problem for growth-bounded graphs. In E. Kranakis and J. Opatrny, editors, *ADHOC-NOW*, volume 4686 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2007.

[6] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(38), 1997.

[7] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998.

[8] N. Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992, February.

[9] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(1):59–68, 1995.

[10] M. Naor and L. Stockmeyer. What can be computed locally? In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 184–193, New York, NY, USA, 1993. ACM Press.

[11] T. Nieberg and J. L. Hurink. A PTAS for the minimum dominating set problem in unit disk graphs. In T. Erlebach and G. Persiano, editors, *3rd International Workshop on Approximation and Online Algorithms, WAOA 2005, Palma de Mallorca, Spain*, volume 3879 of *Lecture Notes in Computer Science*, pages 296–306, Heidelberg, Germany, 2006. Springer-Verlag.

[12] T. Nieberg, J. L. Hurink, and W. Kern. A robust PTAS for maximum weight independent sets in unit disk graphs. In J. Hromkovič, M. Nagel, and B. Westfechtel, editors, *Graph-Theoretic Concepts in Computer Science: 30th International Workshop, WG 2004, Bad Honnef, Germany*, volume 3353 of *Lecture Notes in Computer Science*, pages 214–221, Berlin, 2004. Springer-Verlag.

[13] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[14] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 475–484, New York, May 1997. Association for Computing Machinery.

[15] A. Wiese and E. Kranakis. Local PTAS for Dominating and Connected Dominating Set in Location Aware UDGs. *to appear*, 2007.