Approximating Fault-Tolerant Domination in General Graphs

Klaus-Tycho Foerster*

Abstract

In this paper we study the NP-complete problem of finding small k-dominating sets in general graphs, which allow $k-1$ nodes to fail and still dominate the graph. The classic minimum dominating set problem is a special case with $k = 1$. We show that the approach of having at least k dominating nodes in the neighborhood of every node is not optimal. For each $\alpha > 1$ it can give solutions $\frac{k}{\alpha}$ times larger than a minimum k-dominating set. We also study lower bounds on possible approximation ratios. We show that it is NP-hard to approximate the minimum k-dominating set problem with a factor better than $(0.2267/k)\ln(n/k)$. Furthermore, a result for special finite sums allows us to use a greedy approach for k-domination with an approximation ratio of $\ln(\Delta + k) + 1 < \ln(\Delta) + 1.7$, with Δ being the maximum node-degree. We also achieve an approximation ratio of $\ln(n) + 1.7$ for h-step k-domination, where nodes do not need to be direct neighbors of dominating nodes, but can be h steps away.

1 Introduction

Imagine you are tasked to manually upgrade a set of nodes as cluster-heads in a wireless sensor network, i.e., every node is either a cluster-head or has one as a neighbor. An example could be increased storage or an enhanced battery. Upgrading every node would be expensive, so you just choose a set as small as possible. But due to environmental conditions, sometimes a node is no longer available to its neighborhood. You could counter this by picking at least k cluster-heads in the neighborhood of each location. Of course this costs more, but now $k-1$ nodes could be cut off from their surroundings and still every node is covered. However this somehow feels ineffective. If a cluster-head is cut off, then why should it need $k-1$ additional cluster-heads in its neighborhood? We show that this observation saves up to a multiplicative factor of k/α in the number of needed cluster-heads, for every $\alpha > 1$.

Similar domination problems and its various applications have been extensively studied in the literature for over 60 years [13, 14], with a more recent focus on distributed computing [21, 23]. In the subsequent rise of wireless sensor networks, fault-tolerance has become a desirable property [18], especially for dominating sets [22, 28]. The failure of a single node should not be able to destroy the domination property and a fault-tolerant dominating set should be as small as possible. Many results have been published using unit disk graphs as a model, where two nodes are connected iff their distance is at most one. However, it is already NP-hard to just decide if a graph $G = (V, E)$ is a unit disk graph [5]. Even adding further information like edge lengths does not help, we refer to [20, pp. 30–33] for an overview. Unit disk graphs also cannot cope well with restrictions from the real world, like obstacles [24]. On the other hand, unit disk graphs allow for strong theoretical results, e.g., [2]. In some sense, we choose the other end of the spectrum, by studying general graphs.

More classical applications arise in operations research: Choose a minimum number of service facilities so that each region is served by its own facility or has at least k neighboring regions with such a facility [15]. This problem is known as the k-dominating set problem [10]: Given a graph $G = (V, E)$ and a parameter $k \in \mathbb{N}$, find a small set of nodes D_k, so that even after the removal of any $k-1$ nodes from G, every node is in D_k or has a neighbor in D_k. A set D_k of minimum cardinality is called a minimum k-dominating set. This problem is NP-complete [16], so finding an optimal solution seems out of the question, unless P=NP.

We shortly give some central notations: A minimization problem has an approximation threshold of a_t, if no polynomial time algorithm exists, that can always guarantee a solution of at most a_t times the size of an optimal solution, unless P=NP (see for example [29, Chapter 13]). An approximation algorithm has an approximation ratio of a_r for a minimization problem, if it can always compute in polynomial time a solution that is at most a_r times the size of an optimal solution. We will also assume that graphs $G = (V, E)$ are connected, undirected and have $|V| = n$ nodes with a maximum node-degree of Δ.

We show that a greedy approach (Section 5) for the minimum k-dominating set problem leads to an approximation ratio of $\ln(\Delta + k) + 1 < \ln(\Delta) + 1.7$ (Section 7). To obtain this result we prove a result for special finite sums in Section 6. We also show

[*] Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland, k-t.foerster@tik.ee.ethz.ch

that this upper bound is almost tight, by giving an approximation threshold of $(0.2267/k)\ln(n/k)$ (Section 4). We extend this upper bound of $\ln(n) + 1.7$ for fault-tolerant dominating sets for the related problem of domination with a higher range than just the 1-step neighborhood (see Section 8).

Following the notation of Hage and Harary, this range-extension is called h-step domination [12]. Going back to the example of cluster-heads, if a node is allowed to be one step away, then why not extend this to h steps, as long it is still reachable even if $k-1$ nodes fail? More practical applications of this concept arise in distributed data structures, sparse routing tables and server placement in networks [3, 25].

Combining fault-tolerance and extended range gives the possibility of computing a good solution for the specifications of a real-world network: How large should the domination range be? How many nodes are allowed to fail? An algorithm designer can now parameterize these domination properties and still achieve almost tight approximation bounds of $\ln(n) + 1.7$ for the desired solution.

2 Related Work

A linear time algorithm exists for the minimum k-dominating set problem on block graphs [15]. For the special case of unit disk graphs, various approximation results exist, see for example [8, 22, 31]. In this specific graph model, a unit disk can contain at most five independent nodes, which leads to a constant approximation ratio for the minimum dominating set problem [27]. The LP-based algorithm in [22] for approximating minimum k-dominating sets can be extended to general graphs, giving an approximation ratio of $(e^2 + e)\ln(\Delta) > 10\ln(\Delta)$.

A related problem is the minimum k-tuple dominating set problem in graphs with minimum degree $k-1$, which demands that all nodes are dominated by at least k nodes. This requires that all nodes in the k-tuple dominating set are connected to at least $k-1$ other nodes in the k-tuple dominating set. This problem is NP-complete as well [26]. Approximation algorithms and hardness results for k-tuple domination for general graphs have not been known before the paper of Klasing and Laforest [19], even though many other results are known for k-(tuple)-domination, see [6]. In [19] the authors approximate the k-tuple dominating set problem with an approximation ratio of $\ln(n) + 1$ with an algorithm for a version of set cover, where each set has to be covered k times. This specific technique cannot be applied to the k-dominating set problem, as nodes in the k-dominating set only require to be "covered" once. But since every k-tuple dominating set is also a k-dominating set, one could search for small k-dominating sets in many cases by searching for small k-tuple dominating sets. In Section 3 we show that this approach is not optimal. For each $\alpha > 1$ it can give solutions $\frac{k}{\alpha}$ times larger than a minimum k-dominating set.

We also note that in some works the notion of k-domination is used for different problems, like k-tuple domination. In this paper we follow the notation of Haynes, Hedetniemi and Slater from [14, Chapter 7.1]. In [25] the notion of k-domination is used for h-step domination: Kutten and Peleg give in [25] a distributed approximation algorithm for h-step domination with nearly optimal runtime. Furthermore, a greedy centralized algorithm gives an approximation ratio of $\ln(n) + 1$ for h-step 1-domination, e.g. again in [25].

Another different but related problem is the B-domination problem, where each node must have at least B facilities in its neighborhood – but it is allowed to place multiple facilities on the same node [30].

3 Comparing the Size of k-Dominating Sets and k-Tuple Dominating Sets

From the definition it follows that for each graph G with minimum degree $k-1$ the size of a minimum k-dominating set is at most the size of a minimum k-tuple dominating set. We prove the possiblity that it also can be nearly k times smaller:

Theorem 3.1. *Let $k \in \mathbb{N}$ and let $\alpha \in \mathbb{R}$ satisfying $1 < \alpha < k$. Then for every $n \in \mathbb{N}$ with $n \geq k-1+\frac{(k+1)^2}{\alpha-1}$, there exists a graph G with n nodes for which the size of its minimum k-tuple dominating set is at least $\frac{k}{\alpha}$ times as large as the size of its minimum k-dominating set.*

Proof. Let $k, m \in \mathbb{N}$ with $k \leq m$, we construct a tripartite undirected graph $G_{\left(K,M,\frac{M}{K}\right)}$ with $n = k + m + \left\lceil \frac{m}{k} \right\rceil \geq 2k+1$ nodes as follows (see Figure 1 for an example):

- we construct three independent sets $K, M, \frac{M}{K}$ with $|K| = k$, $|M| = m$ and $\left|\frac{M}{K}\right| = \left\lceil \frac{m}{k} \right\rceil$
- we connect every node from K with every node from M
- we connect $\left\lceil \frac{m}{k} \right\rceil - 1$ nodes from $\frac{M}{K}$ with $\left(\left\lceil \frac{m}{k} \right\rceil - 1\right)k$ nodes from M, so that no two nodes from M share a common neighbor in $\frac{M}{K}$
- if $\left\lceil \frac{m}{k} \right\rceil = \frac{m}{k}$, then connect the last remaining unconnected node from $\frac{M}{K}$ with the last k nodes from M that are not yet connected to nodes from $\frac{M}{K}$

- else connect the last remaining unconnected node from $\frac{M}{K}$ with the $m - (\lceil \frac{m}{k} \rceil - 1)k$ nodes that are not yet connected to any nodes from $\frac{M}{K}$ and also connect it with $k - \big(m - (\lceil \frac{m}{k} \rceil - 1)k\big)$ other random nodes from $\frac{M}{K}$, so that it is connected to exactly k nodes from $\frac{M}{K}$.

In $G_{\left(K,M,\frac{M}{K}\right)}$ the nodes from K and $\frac{M}{K}$ form a k-dominating set. For a k-tuple dominating set, one needs at least m nodes to k-dominate the nodes in $\frac{M}{K}$. It follows that the quotient between the size of a minimum k-tuple dominating set and a minimum k-dominating set for a graph $G_{\left(K,M,\frac{M}{K}\right)}$ is at least $\frac{m}{k+\lceil \frac{m}{k} \rceil} \geq \frac{m}{k+\frac{m}{k}+1} = \frac{m}{\frac{k^2+m+k}{k}} = \frac{mk}{k^2+k+m}$. For a given $\alpha > 1$, we want that quotient to be $\geq \frac{k}{\alpha}$, i.e., $\frac{mk}{k^2+k+m} \geq \frac{k}{\alpha} \Leftrightarrow \frac{k^2+k+m}{m} \leq \alpha \Leftrightarrow k^2 + k + m \leq m + (\alpha - 1)m \Leftrightarrow \frac{k^2+k}{\alpha-1} \leq m$. Now we can calculate for given α, k a number of nodes $n_{\alpha,k} \in \mathbb{N}$, so that for every $n \geq n_{\alpha,k}, n \in \mathbb{N}$ there exists a graph $G_{\left(K,M,\frac{M}{K}\right)}$ with n nodes where the minimum k-tuple-dominating set is at least $\frac{k}{\alpha}$ times as large as the minimum k-dominating set: The number of nodes in $G_{\left(K,M,\frac{M}{K}\right)}$ is $n = m + k + \lceil \frac{m}{k} \rceil$. It follows that $n + 1 \geq m + k + \frac{m}{k}$. To calculate $n_{\alpha,k}$, we can now substitute m with $\frac{k^2+k}{\alpha-1}$ since $\frac{k^2+k}{\alpha-1} \leq m$ and get $n_{\alpha,k} + 1 \geq \frac{k^2+k}{\alpha-1} + k + \frac{\frac{k^2+k}{\alpha-1}}{k} = \frac{k^2+k}{\alpha-1} + k + \frac{k+1}{\alpha-1} = \frac{k^2+2k+1}{\alpha-1} + k \Leftrightarrow n_{\alpha,k} \geq \frac{k^2+2k+1}{\alpha-1} + k - 1 = k - 1 + \frac{(k+1)^2}{\alpha-1}$. □

Furthermore, Theorem 3.1 is sharp in the sense that the lower bound $\alpha > 1$ cannot be improved:

Lemma 3.1. *Let the graph G have a k-dominating set D and let each node of G have at least degree $k - 1$. Then there exists a k-tuple dominating set D^T with size $|D^T| \leq k|D|$.*

Proof. We start with an empty set D^T and add all nodes from D to D^T. Then we take each node $v \in D^T$ and add neighbors of v in V to D^T until v has k neighbors in the D^T. One needs at most $k - 1$ nodes for each node from the k-dominating set D to do so, since a node is also its own neighbor. The nodes of G originally not in D are already dominated k times by the definition of a k-dominating set. The resulting k-tuple dominating set D^T therefore is at most k times as large as the original k-dominating set D. □

Using Lemma 3.1 we immediately obtain a bound for the approximability of the minimum k-dominating set problem in relation to the approximability of the minimum k-tuple dominating set problem:

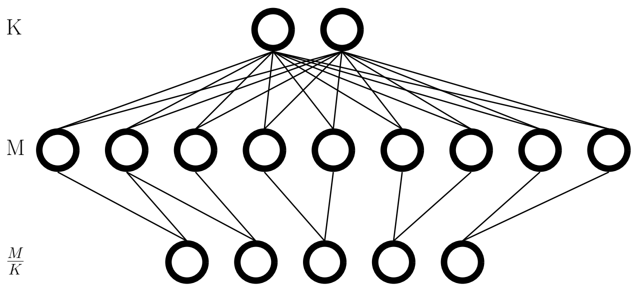

Figure 1: An example for the Graph $G_{\left(K,M,\frac{M}{K}\right)}$ with $|K| = k = 2$, $|M| = m = 9$ and $\left|\frac{M}{K}\right| = \lceil \frac{m}{k} \rceil = \lceil \frac{9}{2} \rceil = 5$, which results in $n = 16$ nodes in total. Each node from K is connected to each node from M and each node from $\frac{M}{K}$ is connected to $k = 2$ different nodes from M, except for the most left one which shares one neighbor in M with the node next to it. In this example, K and $\frac{M}{K}$ form a k-dominating set with $k + \lceil \frac{m}{k} \rceil = 2 + 5 = 7$ nodes. On the other hand, every k-tuple dominating set needs to dominate each node from $\frac{M}{K}$ at least k times. This means a minimum k-tuple dominating set contains at least $|M| = m = 9$ nodes. Similar graphs can be constructed for fixed k by chosing an arbitrarily large m.

Theorem 3.2. *Let $k \in \mathbb{N}$ and let $G = (V, E)$ with $|V| = n$. If $s(n)$ is an approximation threshold for the minimum k-tuple dominating set problem, then $\frac{1}{k}s(n)$ is an approximation threshold for the minimum k-dominating set problem.*

Proof. Consider the family of graphs with minimum degree $k - 1$. If there is an approximation threshold of $\frac{1}{k}s(n)$ for the minimum k-dominating set problem on this family of graphs, then it also applies to general graphs. Let us assume there exists a polynomial approximation algorithm for the minimum k-dominating set problem with approximation threshold of $\frac{1}{k}s(n)$ for this family of graphs. Then we could construct a k-tuple dominating set from it with at most k times its size. But this is a contradiction to the existence of an approximation threshold of $s(n)$ for the minimum k-tuple dominating set problem. □

4 Lower Bounds for Approximating the Minimum k-Dominating Set Problem

At first we prove a theorem that allows us to transfer approximation thresholds from the minimum dominating set problem to the minimum k-dominating set problem. We can then use this Theorem 4.1 to apply the best threshold known to us for the minimum dominating set problem to the minimum k-dominating set problem in Corollary 4.1:

Theorem 4.1. *For $k \in \mathbb{N}$ and arbitrary $G = (V, E)$ with $|V| = n$: If there is an approximation threshold of $s(n)$ for the minimum dominating set problem for G, then there is also an approximation threshold of $\frac{1}{k} s\left(\frac{n}{k}\right)$ for the minimum k-dominating set problem for G.*

Alon et. al. proved in [1] that it is NP-hard to approximate the minimum set cover problem within an approximation ratio of $0.2267 \ln(n)$. Also, the minimum set cover problem is equivalent to minimum dominating set problem under L-reduction, which means here that each instance of each problem can be turned in polynomial time into an equivalent instance of the other problem, see [4][17, pp. 108–109]. Now we can directly derive the following new lower bound:

Corollary 4.1. *For $k \in \mathbb{N}$ and arbitrary $G = (V, E)$ with $|V| = n$: it is NP-hard to approximate the minimum k-dominating set problem within*

$$\frac{0.2267}{k} \ln\left(\frac{n}{k}\right) \quad . \tag{4.1}$$

For ease of notation in the proof of Theorem 4.1, we first introduce the concept of a k-multiplication graph G^k, see Figure 2 for an example. We will use this graph to transfer the approximation thresholds:

Definition 4.1. *Let $k \in \mathbb{N}$, $G = (V, E)$. Let $K_k = (V_k, E_k)$ be the complete graph with k nodes. The k-multiplication graph $G^k = (V^k, E^k)$ is defined as follows:*

1. *V^k is the cartesian product $V \times V_k$*
2. *Two nodes (u_1, u_2) and (v_1, v_2) are adjacent iff*
   (a) *$u_1 \in V$ and $v_1 \in V$ are adjacent in G*
   (b) *or $u_1 = v_1$ with $u_1 \in V$ and $v_1 \in V$*

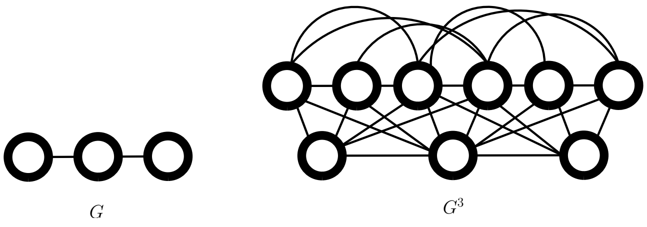

Figure 2: An example for the k-multiplication Graph G^k with $k = 3$.

In other words G^k is being constructed by replacing every node of V with a clique of k nodes (see *1.* and *2.(b)* in Definition 4.1) and generating an edge between nodes if and only if their "original" nodes in G were neighbors (see *2.(a)* in Definition 4.1). We can now prove Theorem 4.1:

Proof. Let D be a dominating set for $G = (V, E)$. Then the set of nodes $D \times V_k$ is a k-dominating set for G^k. We also note the property of $|V^k| = k|V|$. Let D^* be an optimal solution of the minimum dominating set problem for G and let D^*_k be an optimal solution of the minimum k-dominating set problem for G^k. Then $|D^*_k| \leq k|D^*|$ holds.
Let us assume that there exists a polynomial approximation algorithm A for the minimum k-dominating set problem with an approximation ratio of $\frac{1}{k} s\left(\frac{n}{k}\right)$. Then it follows from contradiction that the following algorithm solves the minimum dominating set problem with an approximation ratio of $s(n)$:

1. Construct G^k
2. Generate a k-dominating set D_k of G^k with algorithm A
3. Generate the set of nodes $D \subseteq V$ as follows:
   $v \in D$ if u exists with $(v, u) \in D_k$

G^k has nk nodes, therefore the following inequality holds:

$$|D_k| \leq \frac{1}{k} s\left(\frac{nk}{k}\right) |D^*_k| = \frac{1}{k} s(n) |D^*_k| \quad . \tag{4.2}$$

Furthermore it follows that D is a dominating set for G with $|D| \leq |D_k|$. But now we get a contradiction by applying (4.2): $|D| \leq s(n)|D^*|$. □

5 A Greedy Approach to k-Domination

Before giving an approximation algorithm, we first need two definitions that will play a central role in the algorithm. They will define an appropriate measure how to select the next node in a greedy fashion:

Definition 5.1. *Let $G = (V, E)$ be a graph, $D \subseteq V$ be a set of nodes and $k \in \mathbb{N}$. The degree of k-domination of a node $v \in V$ by D is defined as:*

$$d_k(v, D) = \begin{cases} \min(k, |N(v) \cap D|) \ , & \text{if } v \notin D \\ k \ , & \text{if } v \in D \end{cases} \tag{5.1}$$

This means that d_k is k if v is in D or the amount of neighbors in D otherwise, but limited to k. If the relation to D or v is not clear from the context, we will use the notation of $d_k(v, D)$ or $d_k(v)$.

Definition 5.2. *Let $G = (V, E)$ be a graph, $D \subseteq V$ be a set of nodes and $k \in \mathbb{N}$. The remaining cost $a(G, D)$ for k-domination is defined as:*

$$a(G, D) = nk - \sum_{v \in V} d_k(v, D) \quad . \tag{5.2}$$

The remaining cost therefore is the difference between the sum of all $d_k(v, D_k)$ for all nodes $v \in V$ for a k-dominating set D_k and the sum of all $d_k(v, D)$ for all nodes $v \in V$ for any set of nodes $D \subseteq V$:

$$a\left(G, D\right) = \sum_{v \in V} d_k\left(v, D_k\right) - \sum_{v \in V} d_k\left(v, D\right) \ . \tag{5.3}$$

The following new algorithm generates a finite series of sets of nodes D_k^i with strictly monotone decreasing remaining cost $a(G, D_k^i)$. The last element of the series is a k-dominating set:

Algorithm 1. ***Greedy approximation for a k-dominating set***
Input: *$G = (V, E)$ and $k \in \mathbb{N}$.*
Output: *k-dominating Set D_k.*

```
Initialize $D_k^0 = \emptyset$ and $i = 0$.
Give all nodes their current degree of $k$-domination:
$d_k(v) = 0$ for all $v \in V$.
while there are nodes $v \in V$ with $d_k(v) < k$ do
    Choose a node $g^i \in V$, $g^i \notin D_k^i$ that reduces the
    remaining cost $a(G, D_k^i \bigcup \{g^i\})$ in comparison to
    $a(G, D_k^i)$ the most.
    $d_k(g^i) = k$ and $D_k^{i+1} = D_k^i \bigcup \{g^i\}$.
    Raise for all $v' \in N(g^i)$ with $d_k(v') < k$ the degree
    of $k$-domination $d_k(v')$ by one, i.e., $d_k(v') =$
    $d_k(v') + 1$.
    $i = i + 1$.
end while
$D_k = D_k^i$.
return $D_k$
```

The algorithm terminates correctly, since it adds nodes to the set until all nodes are k-dominated. In the worst case it will add all n nodes. The runtime is dominated by the while loop, which is executed at most n times. All steps in the while loop can be executed in $O(n+|E|)$ time, which results in a total runtime of $O(n(n + |E|))$. We will show in Section 7 that the Algorithm 1 has an approximation ratio of $\ln(\Delta + k) + 1 < \ln(\Delta) + 1.7$, with Δ being the maximum node-degree. For the convenience of the reader, we split the proof and first show a theorem for special finite sums in Section 6, which will be used as a central argument in the proof.

6 A Helpful Lemma for Special Finite Sums

In order to prove upper bounds for the approximation ratio of greedy algorithms for domination problems, several authors (e.g., Guha and Khuller in [11]) have used results for special finite sums for their investigations. In this paper we will use the following estimation, which we need for the proof of our main results in Section 7:

Lemma 6.1. *Let $r \in \mathbb{N}$ with $r > 1$ and let $a_1, a_2, \ldots, a_s \in \mathbb{N}$ with*

$$a_\nu \geq \frac{1}{r}\left(A_s - A_{\nu-1}\right) \quad \textit{for } \nu = 1, 2, \ldots, s \ , \tag{6.1}$$

where the numbers $A_0, A_1, \ldots, A_s$ are defined by

$$A_\nu = \sum_{\mu=1}^{\nu} a_\mu \ . \tag{6.2}$$

Then it follows that $A_\lambda > A_s - r$ for

$$\lambda > \frac{\ln\left(\frac{A_s}{r}\right)}{\ln\left(\frac{r}{r-1}\right)} \ . \tag{6.3}$$

Remark 6.1. *In Lemma 6.1 note that the following rough estimation holds:*

$$\frac{1}{\ln\left(\frac{r}{r-1}\right)} \leq r\left(1 - \frac{1}{2r}\right) \ . \tag{6.4}$$

For the proof consider for real-valued $r > 1$ the continuous function f defined by

$$f(r) = \ln\left(\frac{r}{r-1}\right) - \frac{1}{\left(r - \frac{1}{2}\right)} \ . \tag{6.5}$$

The value of $f(r)$ tends to zero as r tends to infinity. Since the first order derivative of f is smaller than zero for $r > 1$, f is monotonically decreasing and therefore f is positive in $(1, \infty)$.

We now prove Lemma 6.1:

Proof. Let $b_1, b_2, \ldots, b_s$ be the real numbers defined by

$$b_\nu = \frac{1}{r}\left(A_s - B_{\nu-1}\right) \quad \text{for } \nu = 1, 2, \ldots, s \ , \tag{6.6}$$

where $B_0, B_1, \ldots, B_s$ are given by

$$B_\nu = \sum_{\mu=1}^{\nu} b_\mu \ . \tag{6.7}$$

At first we prove by induction that

$$A_\nu \geq B_\nu \quad \text{for } \nu = 0, 1, \ldots, s \ . \tag{6.8}$$

For $\nu = 0$ we have $A_0 = B_0 = 0$. We assume that (6.8) holds for a fixed $\nu < s$:

$$A_\nu = B_\nu + c_\nu \quad , \ c_\nu \geq 0 \ . \tag{6.9}$$

From (6.2), (6.7) and (6.9) it follows that
(6.10)

$$\begin{aligned} A_{\nu+1} - B_{\nu+1} &= \left(A_\nu - B_\nu\right) + \left(a_{\nu+1} - b_{\nu+1}\right) \\ &\geq c_\nu + \tfrac{1}{r}\left(A_s - A_\nu\right) + \tfrac{1}{r}\left(A_s - B_\nu\right) \\ &= c_\nu - \tfrac{1}{r}\left(A_\nu - B_\nu\right) = c_\nu\left(1 - \tfrac{1}{r}\right) \geq 0 \ . \end{aligned}$$

Now we prove by induction that for each $\lambda \geq 1$

$$B_\lambda = A_s \left(1 - \left(1 - \frac{1}{r}\right)^\lambda\right) \ . \tag{6.11}$$

(6.11) is valid for $\lambda = 1$ since by (6.6) we have

$$B_1 = b_1 = \frac{1}{r} A_s = A_s \left(1 - \left(1 - \frac{1}{r}\right)\right) \ . \tag{6.12}$$

Assuming the correctness of (6.11) for a fixed $\lambda \geq 1$ we obtain

(6.13)
$$\begin{aligned} B_{\lambda+1} &= B_\lambda + b_{\lambda+1} = B_\lambda + \tfrac{1}{r}\left(A_s - B_\lambda\right) \\ &= \left(1 - \tfrac{1}{r}\right) B_\lambda + \tfrac{1}{r} A_s \\ &= \left(1 - \tfrac{1}{r}\right) A_s \left(1 - \left(1 - \tfrac{1}{r}\right)^\lambda\right) + \tfrac{1}{r} A_s \\ &= A_s \left(1 - \left(1 - \tfrac{1}{r}\right)^{\lambda+1}\right) \ , \end{aligned}$$

which proves (6.11) for each $\lambda \geq 1$. Now with (6.8) and (6.11) we consider the inequalities

$$A_\lambda \geq B_\lambda = A_s \left(1 - \left(1 - \frac{1}{r}\right)^\lambda\right) > A_s - r \tag{6.14}$$

where the second inequality is equivalent to the inequality

$$\left(\frac{r}{r-1}\right)^\lambda > \frac{A_s}{r} \ , \tag{6.15}$$

which holds iff

$$\lambda > \left(\ln\left(\frac{A_s}{r}\right)\right) / \left(\ln\left(\frac{r}{r-1}\right)\right) \ . \tag{6.16}$$

This proves Inequality (6.3). □

7 Approximation Ratios for k-Domination

In this section we will prove the announced approximation ratio of $\ln(\Delta + k) + 1 < \ln(\Delta) + 1.7 < \ln(n) + 1.7$ for Algorithm 1. We first state an even stronger result in Theorem 7.1 and then the above mentioned results in Corollary 7.1 and 7.2. After that we prove Theorem 7.1 with the help of Lemma 6.1 from Section 6.

Theorem 7.1. *Let $k \in \mathbb{N}$ and $G = (V, E)$ with $|V| = n$. Let $r > 1$ be the size of an optimal solution for the minimum k-dominating set problem for G. Then Algorithm 1 generates a k-dominating set with approximation ratio*

$$\frac{\ln\left(\frac{nk}{r}\right)}{r \ln\left(\frac{r}{r-1}\right)} + 1 \ . \tag{7.1}$$

Remark 7.1. *If we apply Equation (6.4) from Remark 6.1, then a rough estimation results in*

$$\frac{\ln\left(\frac{nk}{r}\right)}{r \ln\left(\frac{r}{r-1}\right)} + 1 \quad \leq \quad \left(1 - \frac{1}{2r}\right) \ln\left(\frac{nk}{r}\right) + 1 \ . \tag{7.2}$$

One node can at most k-dominate itself and 1-dominate all of its at most Δ neighbors. Therefore $\frac{nk}{\Delta+k} \leq r \Leftrightarrow \frac{nk}{r} \leq \Delta + k$ holds. With $\left(1 - \frac{1}{2r}\right) < 1$ we obtain:

Corollary 7.1. *Algorithm 1 has an approximation ratio of less than $\ln(\Delta + k) + 1$.*

If we would set $k > \Delta$, then each node of the graph would need to be a part of the dominating set and a greedy algorithm would find an optimal solution. Therefore we can use the bound $k \leq \Delta$ for our upper bound, which results in $\ln(\Delta + \Delta) + 1 = \ln(\Delta) + \ln(2) + 1 < \ln(\Delta) + 1.7$. Since $\Delta \leq n - 1$, the following corollary holds:

Corollary 7.2. *Algorithm 1 has an approximation ratio of less than $\ln(\Delta) + 1.7 < \ln(n) + 1.7$.*

We now prove Theorem 7.1:

Proof. Let us fix an arbitrary optimal solution of the minimum k-dominating set problem for the input by calling it D_k^*. If we would set $D_k^i = D_k^{i-1} \cup D_k^*$, then D_k^i would be a k-dominating set with remaining cost of domination of $a(G, D_k^i) = 0$. It follows from the pigeonhole principle that in the set of nodes $D_k^* \backslash D_k^{i-1}$, there is at least one node that reduces the remaining cost a by a factor of $\frac{1}{|D_k^*|}$. Since the algorithm chooses the node g^i in a greedy fashion, it holds that

$$\begin{aligned} a(G, D_k^i) =& a(G, D_k^{i-1} \cup \{g^i\}) \\ \leq& a(G, D_k^{i-1}) \left(1 - \frac{1}{|D_k^*|}\right) \ . \end{aligned} \tag{7.3}$$

We can now use Lemma 6.1 with $r = |D_k^*|$ and $A_s = nk$ to estimate after how many steps of the greedy algorithm the remaining cost of domination is less than $|D_k^*|$. Since the cost is a natural number, the remaining cost is then at most $|D_k^*| - 1$. In the worst case, the greedy algorithm will need to add $|D_k^*| - 1$ more nodes to achieve a remaining cost of 0. We can now combine these arguments: If the Algorithm 1 has ended after s steps, then we can bound s by

$$\begin{aligned} s \leq & \left\{ \left\lfloor \left(\ln\left(\frac{nk}{|D_k^*|}\right)\right) / \left(\ln\left(\frac{|D_k^*|}{|D_k^*| - 1}\right)\right) + 1 \right\rfloor \right\} \\ & + (|D_k^*| - 1) \ . \end{aligned} \tag{7.4}$$

Since we are looking for the approximation ratio, we divide s by $|D_k^*|$, which concludes the proof:
(7.5)

$$\frac{s}{|D_k^*|} \leq \left\{ \left(\ln\left(\frac{nk}{|D_k^*|}\right)\right) / \left(|D_k^*| \ln\left(\frac{|D_k^*|}{|D_k^*|-1}\right)\right)\right\} + 1 \ .$$

□

8 Domination with Extended Range

With the help of our Lemma 6.1 we can also apply the general greedy approach from Algorithm 1 to domination with a range greater than one. All we need to change are two things: The degree of domination d for a single node from Definition 5.1 and the remaining cost a for a domination from Definition 5.2. If the modified greedy algorithm still runs in polynomial time, then we still can achieve an approximation ratio of $\ln(n) + 1.7$.

A node v is h-step dominated by a set D^h, if there is a node in D^h at most h steps away from v. A h-step k-dominating set D_k^h is a set of nodes, that remains a h-step dominating set even if any $k-1$ nodes get deleted from the graph.

However now k dominating nodes in the h-neighborhood of v no longer guarantee that $v \notin D_k^h$ will be h-step dominated if any other $k-1$ nodes fail. v could have just one neighbor v', who has k more neighbors in the dominating set. If v' fails, then v is no longer 2-step dominated. If there were k node disjoint paths from $v \notin D_k^h$ to D_k^h (disjoint except for v), then any $k-1$ nodes could fail and v would still be h-step dominated. This condition is not only sufficient, but also necessary: If there were only $k-1$ node disjoint paths, then there would be $k-1$ nodes whose removal would violate the h-step domination property for v.

The checking for such k node disjoint paths of length at most h to a set of nodes D_k^h can be performed in a runtime of $O(k|E|)$, see for example [7, 9]. For ease of notation, we first define the number of node disjoint paths from v to D_k^h:

Definition 8.1. *Let $G = (V, E)$ be a graph, $D_k^h \subseteq V$ be a set of nodes and $h, k \in \mathbb{N}$. The set of node disjoint paths (disjoint except for the start-node) of length at most h from v to nodes in D_k^h in G is defined as $p(v, h, D_k^h, G)$.*

We can now define the degree of domination d_k^h and the remaining cost a_k^h in a similar way as in the Definitions 5.1 and 5.2:

Definition 8.2. *Let $G = (V, E)$ be a graph, $D_k^h \subseteq V$ be a set of nodes and $h, k \in \mathbb{N}$. The degree of h-step k-domination of a node $v \in V$ by D_k^h is defined as:*
(8.1)

$$d_k^h\left(v, D_k^h\right) = \left\{ \begin{array}{c} \min\left(k, \left|p(v, h, D_k^h, G)\right|\right) \ , \quad \text{if } v \notin D_k^h \\ \\ k \ , \quad \text{if } v \in D_k^h \end{array} \right.$$

This means that if a node fails that is not v and $v \in D_k^h$ holds, then v still dominates itself. Should any $k-1$ nodes fail and $v \notin D_k^h$ holds, then there is at least one path left from v to D_k^h. This degree of domination is limited by k. We can now define the remaining cost:

Definition 8.3. *Let $G = (V, E)$ be a graph, $D \subseteq V$ be a set of nodes and $h, k \in \mathbb{N}$. The remaining cost $a_k^h(G, D)$ for h-step k-domination is defined as:*

$$a_k^h\left(G, D\right) = nk - \sum_{v \in V} d_k^h\left(v, D\right) \ . \tag{8.2}$$

Similar as before, the remaining cost a_k^h is the difference between the sum of all $d_k^h(v, D_k^h)$ for all nodes $v \in V$ for a h-step k-dominating set D_k^h and the sum of all $d_k^h(v, D)$ for all nodes $v \in V$ for any set of nodes $D \subseteq V$:

$$a_k^h\left(G, D\right) = \sum_{v \in V} d_k^h\left(v, D_k^h\right) - \sum_{v \in V} d_k^h\left(v, D\right) \ . \tag{8.3}$$

We can now apply this to our greedy algorithm presented earlier:

Theorem 8.1. *Let $k, h \in \mathbb{N}$ and $G = (V, E)$ with $|V| = n$. There exists a greedy algorithm for the minimum h-step k-dominating set problem on G with approximation ratio of* $\ln(n) + 1.7$.

Proof. If we modify Algorithm 1 with the Definitions 8.2 and 8.3 it still has a polynomial runtime. We use basically the same proof as for Theorem 7.1 and Corollaries 7.1 and 7.2 for 1-step k-domination. However now one node can k-dominate itself and its h-neighborhood, instead of just the 1-neighborhood. If we define Δ_h to be the largest h-neighborhood of any node, then we can bound the size of a optimal solution $|D_k^{h,*}|$ by

$$\frac{nk}{\Delta_h + k} \leq |D_k^{h,*}| \Leftrightarrow \frac{nk}{|D_k^{h,*}|} \leq \Delta_h + k \ . \tag{8.4}$$

This yields an approximation ratio of $\ln(\Delta_h + k) + 1 \leq \ln(n + k) + 1$. We can bound $k \leq n$, since else every node would be a part of the dominating set and the greedy algorithm would find an optimal solution. This yields the desired approximation ratio of $\ln(n+n)+1 \leq \ln(n) + \ln(2) + 1 < \ln(n) + 1.7$. □

9 Acknowledgements

The author would like to thank (in alphabetical order) Jiří Adámek, Jürgen Koslowski and Roger Wattenhofer for helpful discussions and suggestions. The author would also like to thank the anonymous reviewers for their helpful comments.

References

[1] Alon, N., Moshkovitz, D., Safra, S.: Algorithmic construction of sets for k-restrictions. *ACM Transactions on Algorithms* 2:2, pp. 153–177 (2006)

[2] Alzoubi, K.M., Wan, P.-J., Frieder, O.: Message-optimal connected dominating sets in mobile ad hoc networks. *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc '02)*: pp. 157–164 (2002)

[3] Bar-Ilan, J., Kortsarz, G., Peleg, D.: How to Allocate Network Centers. *J. Algorithms* 15(3): pp. 385–415 (1993)

[4] Bar-Yehuda, R., Moran, S.: On approximation problems related to the independent set and vertex cover problems. *Discrete Applied Mathematics* 9:1, pp. 1–10 (1984)

[5] Breu, H., Kirkpatrick, D.G.: Unit disk graph recognition is NP-hard. *Comput. Geom.* 9(1-2): pp. 3–24 (1998)

[6] Chellali, M., Favaron, O., Hansberg, A., Volkmann, L.: k-Domination and k-Independence in Graphs: A Survey. *Graphs and Combinatorics* 28:1, pp. 1–55 (2012)

[7] Cormen, T. H., Leiserson, C. E., Rivest, R.L., Stein, C.: *Introduction to Algorithms* (3. ed.). MIT Press (2009)

[8] Couture, M., Barbeau, M., Bose, P. , Kranakis, E.: Incremental Construction of k-Dominating Sets in Wireless Sensor Networks. *Proceedings of the 10th international conference on Principles of Distributed Systems (OPODIS '06)*: pp. 202–214 (2006)

[9] Dai, F., Wu, J.: On constructing k-connected k-dominating set in wireless ad hoc and sensor networks. *J. Parallel Distrib. Comput.* 66(7): pp. 947–958 (2006)

[10] Fink, J.F., Jacobson, M.S.: n-Domination in graphs. In: *Graph theory with applications to algorithms and computer science.* John Wiley & Sons, Inc., New York, pp. 283–300 (1985)

[11] Guha, S., Khuller, S.: Approximation Algorithms for Connected Dominating Sets. *Algorithmica* 20:4, pp. 374–387 (1998)

[12] Hage, P., Harary, F.: *Island Networks: Communication, Kinship, and Classification Structures in Oceania.* Cambridge University Press (1996)

[13] Haynes, T.W., Hedetniemi, S. and Slater, P.: *Domination in Graphs: Advanced Topics.* Marcel Dekker Inc., New York (1998)

[14] Haynes, T.W., Hedetniemi, S. and Slater, P.: *Fundamentals of Domination in Graphs.* Marcel Dekker, Inc., New York (1998)

[15] Hwang, S.-F., Chang, G.J.: The k-neighbor domination problem. *European Journal of Operational Research* 52:3, pp. 373–377 (1991)

[16] Jacobson, M. S., Peters, K.: Complexity questions for n-domination and related parameters. *Congr. Numer.* 68, pp. 7–22 (1989)

[17] Kann, V.: *On the Approximability of NP-complete Optimization Problems.* PhD thesis, Royal Institute of Technology, Stockholm (1992).

[18] Karl, H., Willig, A.: *Protocols and Architectures for Wireless Sensor Networks.* John Wiley & Sons, Ltd, West Sussex (2006)

[19] Klasing, R., Laforest, C.: Hardness results and approximation algorithms of k-tuple domination in graphs. *Inform. Process. Lett.* 89, pp. 75–83 (2004)

[20] Kröller, A.: *Algorithms for Topology-Aware Sensor Networks.* PhD thesis, Braunschweig Institute of Technology, Braunschweig (2008).

[21] Kuhn, F., Moscibroda, T., Wattenhofer, R.: What cannot be computed locally!. *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing (PODC '04)*: pp. 300–309 (2004)

[22] Kuhn, F., Moscibroda, T., Wattenhofer, R.: Fault-Tolerant Clustering in Ad Hoc and Sensor Networks. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS '06)* (2006)

[23] Kuhn, F., Wattenhofer, R.: Constant-time distributed dominating set approximation. *Springer Journal for Distributed Computing* 17:4, pp. 303–310 (2005)

[24] Kuhn, F., Wattenhofer, R., Zollinger, A.: Ad hoc networks beyond unit disk graphs. *Wireless Networks* 14(5): pp. 715–729 (2008)

[25] Kutten, S., Peleg, D.: Fast Distributed Construction of Small k-Dominating Sets and Applications. *Journal of Algorithms* 28:1, pp. 40–66 (1998)

[26] Liao, C.-S., Chang, G.J.: k-tuple domination in graphs. *Inform. Process. Lett.* 87, pp. 45–50 (2003)

[27] Marathe, M.V., Breu, H., Hunt III, H.B. , Ravi, S. S., Rosenkrantz, D.J.: Simple Heuristics for Unit Disk Graphs. *Networks* 25, pp. 59–68 (1995)

[28] Moscibroda, T., Wattenhofer, R.: Maximizing the Lifetime of Dominating Sets. *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)* (2005)

[29] Papadimitriou, C.H.: *Computational Complexity.* Addison-Wesley (1995)

[30] Srinivasan, A.: Improved Approximation Guarantees for Packing and Covering Integer Programs. *SIAM J. Comput.* 29:3, pp. 648–680 (1999)

[31] Wu, Y., Li, Y.: Construction algorithms for k-connected m-dominating sets in wireless sensor networks. *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '08)*: pp. 83–90 (2008)