Maximizing the Lifetime of Dominating Sets

Thomas Moscibroda and Roger Wattenhofer
{moscitho, wattenhofer}@tik.ee.ethz.ch
Computer Engineering and Networks Laboratory, ETH Zurich
8092 Zurich, Switzerland

Abstract

We investigate the problem of maximizing the lifetime of wireless ad hoc and sensor networks. Being battery powered, nodes in such networks have to perform their intended task under rigid energy restrictions that forces the designers to impose a judicious power management and scheduling. For the purpose of saving energy, dominating set based clustering has turned out to be a useful and generic concept in such networks. In data gathering applications, for example, only nodes in the dominating set must be active, while all other nodes can remain in the energy-efficient sleep mode. Prolonging the duration of such a dominating set based clustering is a key algorithmic challenge. In this paper, we define the maximum cluster-lifetime problem *which asks for a schedule that maximizes the time the network is clustered by a dominating set. We give approximation algorithms with an approximation ratio of $O(\log n)$ for several variants of the maximum cluster-lifetime problem. Our approach is based on results given in a paper by Feige, Halldórsson, Kortsarz, and Srinivasan on the* domatic partition *problem [5].*

1. Introduction

One of the key characteristics of wireless ad hoc and particularly sensor networks is the scarcity of energy, the most vital resource. In most cases, sensor nodes are battery powered, and they can stay active for only a limited time before the battery resources are depleted. Not surprisingly, improving the network lifetime (i.e., the time interval in which the network is capable of performing its intended task) is a significant part of the system design in sensor networks, and mechanisms that conserve energy resources are highly desirable.

One of the most prominent such mechanisms is to schedule the node activity such as to allow redundant nodes to enter some kind of *sleep* mode as often and for as long as possible. During this sleep mode, nodes can neither receive nor send messages, but they hardly use any energy. The energy consumed in the *active* mode with the CPU operating at full energy is typically orders of magnitudes higher than in the sleep mode.

Consider a typical application such as *data gathering*, where the nodes have to produce relevant information by sensing an extended geographical area that is eventually sent to an information sink for processing. In data gathering, partitioning the network into clusters plays a vital role. Two nodes being within each others transmission range implies a physical proximity of these nodes. This proximity can be exploited for the sake of energy-conservation. In particular, neighboring nodes may be capable of taking over each other's sensing task. In this case, when one node is actively gathering data, all its neighbors may enter the sleep mode and save energy for future use. Thus, at any time, only a dominating set of nodes is required to be active, instead of all network nodes.

Similar to the data gathering application, clustering the network has proven to be one of the most successful strategies when dealing with the complexity of wireless ad hoc and sensor networks. Clustering improves the usage of scarce resources such as bandwidth and energy, and it helps realizing spatial multiplexing in non-overlapping clusters. Depending on the specific network organization problem at hand, different forms of clustering have been proposed. One frequent approach is to choose clusterheads such that every node is either a clusterhead or has a least one clusterhead in its neighborhood. When modelling the network as a graph $G = (V, E)$, this form of clustering maps to the well-known *dominating set* problem. A dominating set of G is a subset $S \subseteq V$ such that, each node $v \in V$ is either in S or has a neighbor in S. The *minimum dominating set* problem asks for a dominating set of minimal cardinality.

An important characteristic of wireless ad-hoc and sensor networks is that node failure is an event of non-negligible probability. For applications where fault-tolerance is critical, a simple dominating set may not be a desirable form of clustering. The notion of a k-*dominating* *set* accounts for this additional fault-tolerance by demanding that every node has at least k clusterheads in its neighborhood. Formally, a k-dominating set is a subset $S \subseteq V$ such that, each node $v \in V$ has at least k dominators in its neighborhood (including itself) in S.

Finding a good, low-cardinality dominating set is an important algorithmic challenge. But in view of the scarcity of energy in wireless sensor networks, maximizing the *lifetime* of dominating set based structures is at least as important. What does the best dominating set help if the battery of the dominators are irrevocably depleted after a short period of time?

In this paper, we tackle this problem by finding a large number of *disjoint dominating sets*, as this has a direct impact on conserving energy resources, hence prolonging the network lifetime. The idea is that by using several disjoint dominating sets, we can activate them successively – i.e., only nodes in the currently active set take over the data gathering responsibilities; other nodes are in the low-energy sleep mode. The problem of finding a maximum number of disjoint dominating sets is called the *maximum domatic partition* problem and it is a classic graph theory problem [7, 5]. The maximum number of disjoint dominating sets that can be established in a graph G is called the *domatic number* $D(G)$.

In large-scale distributed (and possibly even mobile) systems such as wireless ad hoc and sensor networks, the usage of centralized algorithms that are based on a global view of the network graph are infeasible. Instead, nodes must be able to come up with a solution based on locally and quickly obtainable information. By turning a distributed domatic partition algorithm into approximation algorithms for the maximum cluster-lifetime problem, we show that a good solution to the domatic partition problem is the key to obtaining an energy-efficient clustering schedule.

Particularly, we propose distributed, randomized algorithms which approximate the optimal solution within a factor of $O(\log n)$ with high probability. As approximating the domatic partition problem within a factor better than $\ln n$ was proven to be impossible unless $NP \subseteq DTIME(n^{O(\log\log n)})$ [5], these approximation algorithms are likely to be asymptotically tight. We also give approximation algorithms for the non-uniform case where each node's initial battery supply may differ, as well as the fault-tolerant case. Note that all our algorithms are completely distributed and require only a constant number of communication rounds. More precisely, communication is only needed to let each node know its 2-hop neighborhood.

The remainder of this paper is structured as follows. In Section 2, we describe our model of computation and formally define the *Maximum Cluster-Lifetime* problem. In Section 3, we give an overview over the relevant existing literature. The special case where all nodes have identical initial battery supply is discussed in Section 4. An algorithm for the general case is subsequently given in Section 5. In Section 6, we also consider the fault-tolerant uniform case. Finally, Section 7 concludes the paper and indicates directions for future research.

2. Problem Statement and Model

In this section, we describe the model and introduce the notation that is being used throughout the paper.

We model the network as a graph $G = (V, E)$. Each network node is represented by a vertex $v \in V$ and there is an edge $\{u, v\} \in E$ between two nodes if and only if u and v are within each others communication range. Note that we do not make the assumption that the underlying network graph forms a *Unit Disk Graph* or any other specific graph. Our algorithms work even in completely arbitrary graphs. However, in order to prevent already basic communication between neighboring nodes from becoming unacceptably cumbersome [19], we assume that a message sent over a link can be acknowledged by sending a corresponding message over the same link in the opposite direction. In other words, we consider only undirected edges. Let $n = |V|$ denote the number of nodes in the network. We assume that this value (or an upper bound thereof) is known to all nodes in the network. The set of neighbors of a node v is written as N_v, and $N_v^+ := N_v \cup \{v\}$. The *degree* δ_v of a node is the number of its neighbors, i.e., $\delta_v = |N_v|$. By $\Delta := \max_{v\in V} \delta_v$ and $\delta := \min_{v\in V} \delta_v$, we denote the maximum and minimum degree (number of neighbors) in the network, respectively.

Further, let b_v be the maximum time node v can be in a dominating set. Clearly b_v depends on the initial battery supply, as well as the energy consumption per time-unit. Note that the value b_v does not necessarily have to be the total available battery power at node v. On the contrary, in most applications, b_v will be set to a value strictly smaller than the total available energy. The idea is to keep some remaining energy at each node in order to guarantee that, for instance, the gathered data can be sent to the sink (for example by collectively constructing a data aggregation tree), where the data can subsequently be processed.

We now formally define the *Maximum Clustering-Lifetime* problem. A *schedule* $\mathcal{S}$ is a set of pairs $(D_1, t_1), \ldots, (D_k, t_k)$, where D_i is a dominating set and t_i is the time during which D_i is active. More precisely, the set D_1 is active in the interval $[0, \ldots, t_1]$ and generally, D_i is active in the interval $[\sum_{j=0}^{i-1} t_j, \ldots, \sum_{j=0}^{i} t_j]$. The *lifetime* of a schedule is defined as $L(\mathcal{S}) := \sum_{i=1}^{k} t_i$. The *Maximum Clustering-Lifetime Problem* asks for the schedule $\mathcal{S}$ with maximum length $L(\mathcal{S})$, such that $\sum_{i:v\in D_i} t_i \le b_v$ holds for all nodes $v \in V$. The last con-

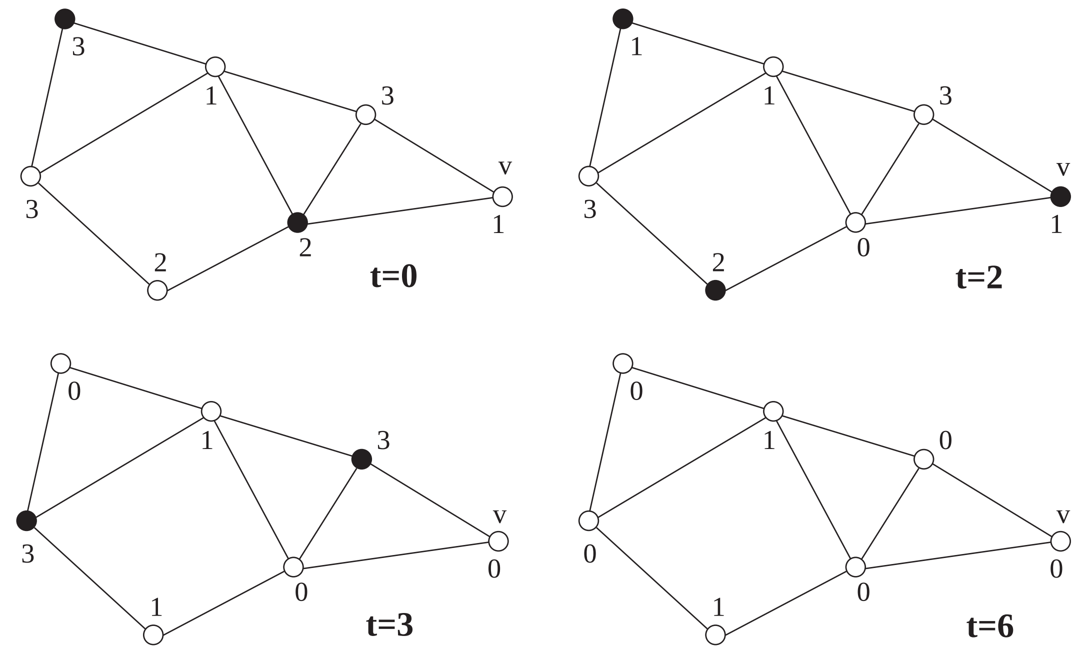

Figure 1. An optimal schedule $\mathcal{S}$ with length $L(\mathcal{S}) = 6$. After the last step, node v cannot be covered anymore.

dition states that the total time each node v can take part in a dominating set is bounded by b_v. We assume that $b_v \in \mathbb{N}$ and intuitively, $b_v = 1$ means that node v can stay in a dominating set for exactly one time-slot.

Figure 1 shows an optimal schedule $\mathcal{S}$ with length $L(\mathcal{S}) = 6$ for a sample graph with 7 nodes. The number next to each node denotes its currently remaining battery supply. During the first two time-slots, a dominating set consisting of two nodes is active. By time 3, the energy-level of these two nodes has decreased by two units each. For the interval $[2, \ldots, 3]$, a dominating set consisting of three nodes is selected. This set is replaced in the interval $[3, \ldots, 6]$ by another two-node dominating set. Finally, at time 6, all nodes in the neighborhood of v have used up their energy for joining dominating sets and hence, v cannot be covered any longer. Note that the optimal solution is not unique. There are other schedules having lifetime 6.

As mentioned in the introduction, many applications require a certain degree of fault-tolerance. The failure of a single node at the wrong time should not destabilize the entire system. In the *Maximum k-tolerant Clustering-Lifetime Problem* we demand that at any time t, every node $v \in V$ has at least k dominators in its neighborhood. In contrast to the regular *Maximum Clustering-Lifetime Problem*, we want to find a schedule $(D_1, t_1), \ldots, (D_k, t_k)$, such that each D_i is a k-dominating set. In order for the *Maximum k-tolerant Clustering-Lifetime Problem* to have a feasible solution, we consider only graphs in which $\delta + 1 \geq k$.

Throughout the paper, we write S_{OPT} to denote the optimal schedule. Further, we write $L_{OPT} := L(S_{OPT})$. Finally, the indicator variable $\mathcal{S}_v(s_1 : s_2)$ denotes whether node v is active in $\mathcal{S}$ during the time interval $[s_1, \ldots, s_2]$. That is, $\mathcal{S}_v(s_1 : s_2) = 1$ if $v \in D_i$ and D_i is active in the interval $[s_1, \ldots, s_2]$. If only a given time-slot t is considered, we also write $\mathcal{S}_v(t)$.

We conclude the section with a well-known fact we will use for the analysis. The proof can be found in standard mathematical textbooks.

Fact 2.1. *For all n, t, such that $n \geq 1$ and $|t| \leq n$,*

$$e^t \left(1 - \frac{t^2}{n}\right) \leq \left(1 + \frac{t}{n}\right)^n \leq e^t.$$

3. Related Work

The importance of clustering in wireless ad hoc and sensor networks has lead to a plethora of papers on dominating sets. From a theoretical point of view, the minimum dominating set problem has been shown to be NP-hard in [9, 12]. Moreover, unless the problems in NP can be solved by deterministic $n^{O(\log \log n)}$ algorithms, no algorithm can approximate the minimum dominating set problem better than $\ln \Delta$, where Δ is the highest degree in the network [18, 4].

One of the first distributed approximation algorithms for the minimum dominating set problem was given in [16]. This was subsequently improved in [11]. The work of [15] explores the trade-off between approximation and communication. In particular, [15] gives an algorithm that achieves an approximation ratio of $O(k\Delta^{2/k} \log n)$ in $O(k^2)$ communication rounds only. Hence, it is shown that even with a constant number of communication rounds, a non-trivial approximation ratio can be achieved. On the other hand, it was shown in [14] that the above upper bound is not too far away from being tight. In particular, it is proven in [14] that (even in messages are unbounded), in k rounds of communication, no algorithm can approximate the minimum dominating set problem better than $\Omega(n^{c/k^2}/k)$ or $\Omega(\Delta^{1/k}/k)$, for some constant c. It follows that every, possibly even randomized algorithm requires more than $\Omega(\sqrt{\log n/\log\log n})$ or $\Omega(\log \Delta/\log\log \Delta)$ communication rounds in order to obtain a constant or polylogarithmic approximation ratio.

The communication in wireless ad-hoc and sensor networks is often modelled as a so-called unit disk graph UDG [3]. In a UDG G, there is an edge between two nodes if their Euclidean distance is at most 1. The dominating set problem remains NP-hard in this case, but constant approximations become possible. Particularly, every maximal independent set is a $4 \cdot OPT + 1$ approximation to an optimal dominating set [21]. Hence, the elegant randomized algorithm by Luby [17] allows to find a constant approximation to the minimum dominating set problem in time $O(\log n)$. If nodes are capable of sensing the distance to its neighbors, a constant approximation ratio can be achieved in time $O(\log\log n)$ [8].

Often, dominating set based clusterings are used for establishing an efficient MAC-layer. Clearly, obtaining a dominating set for that particular purpose demands a protocol that is not based on the assumption that an existing MAC-layer is available. Moreover, such algorithms must be capable of working even if nodes can wake up asynchronously without having access to a global clock or without knowing their neighbors. For this scenario, an algorithm achieving a constant approximation ratio in roughly $O(\log^2 n)$ was given in [13].

All of the above algorithms focus on finding *one* good dominating set. None of them is concerned with the *duration* a dominating set based structure can be maintained in the network.

The single most important related work is therefore the seminal paper by Feige, Halldórsson, Kortsarz, and Srinivasan on domatic partitions [5]. In this paper, the authors prove that every graph with maximum degree Δ and minimum degree δ contains a domatic partition of size $(1 - o(1))(\delta + 1)/\ln \Delta$. They turn this proof into a polynomial time, centralized algorithm that produces a domatic partition of $\Omega(\delta/\ln \Delta)$ sets. Furthermore, they prove that approximating the maximum domatic partition problem within a factor of $(1 + o(1)) \ln n$ is impossible, unless $NP \subseteq DTIME(n^{O(\log\log n)})$. Finally, they analyze the performance of the natural greedy algorithm that repeatedly picks good dominating sets, showing that the approximation ratio of the greedy domatic partition algorithm is bounded by $O(\sqrt{n \log n})$. Fujita [6] gives examples where the performance of this greedy algorithm is as bad as $\Omega(\sqrt{n})$. The authors of [5] investigate the domatic partition problem in such a thoroughness that most related questions appear to be satisfactorily solved. In this paper, we show that the methods introduced in [5] have an important application in wireless ad hoc and sensor networks, helping to increase network lifetime.

Maximizing the number of disjoint dominating sets for the purpose of increasing the lifetime of network organization was later considered in [20, 2]. The papers consider the problem of covering a geometrical region by network nodes for as long a period of time as possible. Both papers propose heuristics for finding the maximum number of disjoint dominating sets. They do *not* provide any worst-case analysis and stringent bounds. Moreover, the heuristics proposed are centralized and hence inappropriate from a practical point of view.

4. The Uniform Case

In this section, we consider the special case where all nodes have the same initial battery level, i.e., $b_v = b$ for all $v \in V$. We begin the analysis with a straightforward observation which lower bounds the optimal value L_{OPT}.

Lemma 4.1. *The lifetime of the optimal schedule S_{OPT} is at most $L_{OPT} \leq b \cdot (\delta + 1)$.*

Proof. Let v be a node with degree $\delta_v = \delta$. Since S_{OPT} is a correct schedule, v must be covered by either itself or by a neighbor during the entirety of the schedule. Because each node can be part of a dominating set at most b time-units, v can be covered at most for $b \cdot (\delta + 1)$ time-units. □

Using this lower bound, we can show that a probabilistic argument given in [5] can be turned into an efficient approximation algorithm with an approximation ratio of $O(\log n)$. The algorithm is randomized; each node v randomly chooses a color in the range from $[1, \ldots, \delta_v^{(2)}/(3 \log n)]$, where $\delta_v^{(2)}$ denotes the minimum degree of a node in N_v^+. The idea is to interpret the different color classes as a *domatic partition* of the network graph. The schedule S then simply follows by activating each of the color classes one after another.

We begin the proof by showing that with high probability, each individual color class forms a valid dominating set.

Algorithm 1 Uniform Algorithm

1: Send δ_v to all neighbors;
2: Receive δ_u from all $u \in N_v$;
3: $\delta_v^{(2)} := \min_{u \in N_v^+} \delta_u$
4: Choose randomly a color c_v from the range $[0, \ldots, \delta_v^{(2)}/(3 \ln n)]$
5: $\mathcal{S}_v(bc_v : b(c_v + 1) - 1) := 1$;

For that purpose, let C_i be the set of nodes that have randomly chosen $c_v = i$ in line 3 of the algorithm. Notice that in the following, we can assume that $\delta_v + 1 \geq 3 \ln n$ for all $v \in V$ because otherwise, a $O(\log n)$ approximation follows directly from Lemma 4.1, even if all nodes choose the same color.

Lemma 4.2. *The set C_i forms a dominating set in G for all $i \in [0, \ldots, \delta/(3 \ln n)]$ with probability $1 - o(n^{-1})$.*

Proof. Following the reasoning in [5], we begin by computing the probability that an individual color class C_i is a dominating set. Subsequently, we show that with high probability, the claim holds for *all* color classes.

Let $A_{v,c}$ be the event that there is no node in N_v^+ which has chosen color c in line 4. If this event is true for some color c and an arbitrary node v, it means that the color class C_c does not form a dominating set.

For an arbitrary node v and a color c, it holds that

$$\begin{aligned} P[A_{v,c}] &= \prod_{u \in N_v^+} \left(1 - \frac{3 \ln n}{\delta_u^{(2)}}\right) \\ &\leq \left(1 - \frac{1}{\delta_v/(3 \ln n)}\right)^{\delta_v + 1} \\ &\underset{\text{Fact 2.1}}{\leq} e^{-(\delta_v+1)(3 \ln n)/\delta_v} \\ &< e^{-3 \ln n} = n^{-3}, \end{aligned}$$

where the first inequality follows from $\delta_u^{(2)} \leq \delta_v$ for all $u \in N_v^+$.

Let B_v be the event that node v does not have all colors $c \in [0, \ldots, \delta_v/(3 \ln n)]$ in its neighborhood. Formally, B_v is true if there exists an $i \in [0, \ldots, \delta_v/(3 \ln n)]$, such that, $C_i \cap N_v^+ = \emptyset$. Using the fact that the probability of the union of events is no more than the sum of their probabilities, we can bound B_v as

$$P[B_v] = P\left[\bigcup_{c=0}^{\delta_v/(3 \ln n)} A_{v,c}\right] \leq \frac{\delta_v}{3 \ln n} n^{-3}.$$

Assume B_v is false for some v. Because of $\delta_v \geq \delta$, this implies that for all $i \in [0, \ldots, \delta/(3 \ln n)]$, $C_i \cap N_v^+ \neq \emptyset$. Hence, $P[B_v]$ is an upper bound on the probability that v is not properly covered by $\delta/(3 \ln n)$ disjoint dominating sets. We now bound the probability P_1 that there is at least one node v which does not have all colors $[0, \ldots, \delta/(3 \ln n)]$ in its neighborhood. Using $\delta_v \leq n$ for all v, we obtain

$$\begin{aligned} P_1 &\leq P\left[\bigcup_{v \in V} B_v\right] \leq \sum_{v \in V} \frac{\delta_v}{3n^3 \ln n} \\ &\leq \frac{1}{3} \sum_{v \in V} \frac{1}{n^2 \ln n} \in O\left(\frac{1}{n \log n}\right). \end{aligned}$$

Hence, with high probability, the color classes C_i for $i \in [0, \ldots, \delta/(3 \ln n)]$ as computed by Algorithm 1 form $\delta_v/(3 \ln n) + 1$ many disjoint dominating sets. □

It only remains to be shown that the schedule $\mathcal{S}$ computed by the algorithm does indeed give the claimed approximation ratio.

Theorem 4.3. *The schedule $\mathcal{S}$ computed by Algorithm 1 is a $O(\log n)$ approximation to the Maximum Cluster-Lifetime problem with probability at least $1 - o(n^{-1})$.*

Proof. Let $\mathcal{S}_{ALG}$ be the schedule computed by Algorithm 1. $\mathcal{S}_{ALG}$ assigns the time interval $[ib, \ldots, (i+1)b]$ to nodes $v \in C_i$. By Lemma 4.2, we know that with probability $1 - o(n)$, the color classes from 0 to $\delta/(3 \ln n)$ form correct dominating sets. It follows that

$$L(\mathcal{S}_{ALG}) \geq (\delta/(3 \ln n) + 1)b$$

holds with high probability, too. The approximation ratio of $O(\log n)$ now follows directly from the bound on L_{OPT} in Lemma 4.1. □

5. The General Case

We now move on to the problem in its full generality by allowing each node v to have a different initial battery supply b_v. In analogy to the uniform case, we start with a lemma bounding the optimal value L_{OPT} from above.

Lemma 5.1. *The lifetime of the optimal schedule S_{OPT} is at most*

$$L_{OPT} \leq \min_{u \in V} \sum_{v \in N_u^+} b_v.$$

Proof. Let u be a node that minimizes $\sum_{v \in N_u^+} b_v$. u must be covered by a dominator in N_u^+ for the entire duration of L_{OPT}. Let $E_u(t)$ be the total remaining energy in N_u^+ at time t. Clearly, $E_u(0) = \sum_{v \in N_u^+} b_v$. In each subsequent time-slot, at least one neighbor must be in the dominating set, and therefore $E_u(t+1) \leq E_u(t) - 1$. It follows that u can be covered only for $E_u(0)$ time-slots. □

Our algorithm for the general case is also randomized. Instead of choosing only one color c_v, each node v randomly and independently chooses b_v many colors in a certain range (recall that $b_v \in \mathbb{N}$). Again, the idea is that by restricting the color range appropriately, we can guarantee that each node has many different color classes in its neighborhood. As in the uniform case, the schedule is then produced by invoking the different color classes consecutively.

Algorithm 2 General Algorithm

1: Send b_v to all neighbors;
2: $\hat{b}_v := \max_{u \in N_v^+} b_u$
3: $\tau_v := \sum_{u \in N_v^+} b_u$;
4: Send $(\hat{b}_v, \tau_v)$ to all neighbors;
5: $\hat{b}_v^{(2)} := \max_{u \in N_v^+} \hat{b}_u$
6: $\tau_v^{(2)} := \min_{u \in N_v^+} \tau_u$;
7: $\mathcal{C} := \emptyset$;
8: **for** $j := 0$ to b_v by 1 **do**
9: Choose randomly a color $c_v^{(j)}$ from the range $[0, \ldots, \tau_v^{(2)}/(3\ln(\hat{b}_v^{(2)} n))]$;
10: $\mathcal{C} := \mathcal{C} \cup \{c_v^{(j)}\}$;
11: **end for**
12: $\mathcal{S}_v(t) := \begin{cases} 1 & , t \in \mathcal{C} \\ 0 & , t \notin \mathcal{C} \end{cases}$

The proof is similar to the uniform one, the difference being that instead of the minimum degree δ, we consider the *minimum energy coverage* $\min_{u \in V} \sum_{v \in N_u^+} b_v$ of any node in the network.

Lemma 5.2. *Let $\tau := \min_{u \in V} \sum_{v \in N_u^+} b_v$ be the minimum energy coverage of the network G, and let $b_{max} := \max_{v \in V} b_v$. The set C_i forms a dominating set in G for all $i \in [0, \ldots, \tau/(3\ln(b_{max} n))]$ with probability $1 - o(n^{-1})$.*

Proof. Again, we first show that the probability of C_i not being a dominating set is small. Let the event $A_{v,c}$ be defined as in Section 4.

After line 6 of Algorithm 2, it holds that $\tau_u^{(2)} \leq \tau_v$ for all $v \in V$ and $u \in N_v^+$. Consequently, for an arbitrary node v and a color c, we have

$$\begin{aligned} P[A_{v,c}] &= \prod_{u \in N_v^+} \prod_{i=1}^{b_u} \left(1 - \frac{3\ln(\hat{b}_u^{(2)} n)}{\tau_u^{(2)}}\right) \\ &\underset{(\tau_u^{(2)} \leq \tau_v)}{\leq} \prod_{u \in N_v^+} \left(1 - \frac{3\ln(\hat{b}_u^{(2)} n)}{\tau_v}\right)^{b_u} \\ &\underset{(\hat{b}_u^{(2)} \geq \hat{b}_v)}{\leq} \prod_{u \in N_v^+} \left(1 - \frac{3\ln(\hat{b}_v n)}{\tau_v}\right)^{b_u} \\ &= \left(1 - \frac{3\ln(\hat{b}_v n)}{\sum_{u \in N_v^+} b_u}\right)^{\sum_{u \in N_v^+} b_u} \\ &\underset{\text{Fact 2.1}}{\leq} e^{-3\ln(\hat{b}_v n)} = \frac{1}{(\hat{b}_v n)^3}. \end{aligned}$$

Like in the uniform case, we now compute the probability of the event B_v that node v does *not* have all colors $c \in [0, \ldots, \tau_v^{(2)}/(3\ln(\hat{b}_v^{(2)} n))]$ in its neighborhood. For that purpose, recall the following relationship between $\tau_v^{(2)}$ and $\hat{b}_v$,

$$\tau_v^{(2)} \leq \tau_v = \sum_{u \in N_v^+} b_u \leq (\delta_v + 1) \cdot \hat{b}_v. \tag{1}$$

With the above inequality, $P[B_v]$ can be upper bounded by

$$\begin{aligned} P[B_v] &\leq \frac{\tau_v^{(2)}}{3\ln(\hat{b}_v^{(2)} n)} \cdot \frac{1}{(\hat{b}_v n)^3} \\ &\underset{\text{Eq. (1)}}{\leq} \frac{\delta_v + 1}{3\ln(\hat{b}_v^{(2)} n)} \cdot \frac{1}{\hat{b}_v^2 n^3}. \end{aligned}$$

Since there are n nodes in the network, the probability that there exists a node $v \in V$ for which the event B_v is true is at most

$$\begin{aligned} P[\bigcup_{v \in V} B_v] &\leq \frac{\delta_v + 1}{2\ln(\hat{b}_v^{(2)} n)} \cdot \frac{1}{(\hat{b}_v n)^2} \\ &\in O\left(\frac{1}{n \log n}\right) \end{aligned}$$

due to $(\delta_v + 1) \leq n$ for all $v \in V$.

With high probability, each node v has all color classes C_i for $i \in [0, \ldots, \tau_v^{(2)}/(3\ln(\hat{b}_v^{(2)} n))]$ in its neighborhood. The lemma now follows from the fact that $\tau_v^{(2)} \geq \tau$ and $\hat{b}_v^{(2)} \leq b_{max}$ holds for all nodes $v \in V$. In other words, at least the color classes in the range $[0, \ldots, \tau/(3\ln(b_{max} n))]$ form correct dominating sets with high probability. □

Combining Lemmas 5.1 and 5.2 yields the following theorem.

Theorem 5.3. *The schedule $\mathcal{S}$ computed by Algorithm 2 is a $O(\log(b_{max} n))$ approximation to the general Maximum Cluster-Lifetime problem with probability at least $1 - o(n^{-1})$.*

Proof. Let $\mathcal{S}_{ALG}$ be the schedule computed by Algorithm 2. By Lemma 5.2, we know that with probability $1 - o(n)$, the color classes from 0 to $\tau/(3\log(b_{max} n))$ form correct dominating sets and thus,

$$\begin{aligned} L(\mathcal{S}_{ALG}) &\geq \frac{\min_{u \in V} \sum_{v \in N_u^+} b_v}{3\log(b_{max} n)} + 1 \\ &\geq \frac{L_{OPT}}{3\log(b_{max} n)} + 1. \end{aligned}$$

Therefore, the approximation ratio α is in $O(\log(b_{max} n))$. For b_{max} polynomial in n, this reduces to $O(\log n)$. □

6. Fault-Tolerance

In this section, we adjust our algorithms to provide the desired degree of fault-tolerance. We begin by giving an algorithm for the uniform k-tolerant version of the problem, i.e., every node must constantly be covered by at least k nodes in its neighborhood.

Algorithm 3 for the fault-tolerant uniform case is an extension of the uniform case as discussed in Section 4. The main idea is that we can merge k consecutive color classes to one color class. Since each individual color class forms a dominating set with high probability and because every node is in exactly one such color class, the merged color class forms a valid k-dominating set. In order to guarantee the $O(\log n)$ approximation ratio, we need to distinguish the two cases $\delta/3\ln n < k$ and $\delta/3\ln n \geq k$. In the second case, merging k consecutive color classes to a single one will be sufficient. In the first case, however, there are fewer than k color classes. Without adjusting the algorithm, this would lead to a schedule of length $L(\mathcal{S}) = 0$. Clearly, in order to come up with an approximation algorithm, we have to take take care of this special case. We do so by letting every node be active for a constant fraction of time b, say $b/2$. Then, the algorithm continues as in Algorithm 1.

Algorithm 3 Fault-Tolerant Uniform Algorithm

1: Send δ_v to all neighbors;
2: Receive δ_u from all $u \in N_v$;
3: $\delta_v^{(2)} := \min_{u \in N_v^+} \delta_u$
4: Choose randomly a color c_v from the range $[0, \ldots, \delta_v^{(2)}/(3\ln n)]$
5: $\mathcal{S}_v(1 : \frac{b}{2}) := 1$;
6: $\mathcal{S}_v(\frac{b}{2} + \frac{b}{2} \cdot \lfloor \frac{c_v}{k} \rfloor : \frac{b}{2} + \frac{b}{2} \cdot (\lfloor \frac{c_v}{k} \rfloor + 1)) := 1$;

In analogy to Sections 4 and 5, we first give a simple lemma bounding the optimal value L_{OPT} from below.

Lemma 6.1. *The lifetime of the optimal schedule S_{OPT} is at most $L_{OPT} \leq b \cdot (\delta + 1)/k$.*

Proof. The proof follows along the lines of the corresponding proofs in Sections 4 and 5. Let v be a node with minimum degree, i.e., $\delta_v = \delta$. Since S_{OPT} is a correct schedule, v must have at least k dominators in N_v^+ at all times. The lemma follows because each node can be part of a dominating set at most b time-units, v can be covered at most for $b \cdot (\delta + 1)/k$ time-units. □

For the upper bound of the algorithm, we distinguish the two cases in line 4 of the algorithm. We obtain the following result.

Theorem 6.2. *The schedule $\mathcal{S}$ computed by Algorithm 3 is a $O(\log n)$ approximation to the uniform Maximum k-tolerant Cluster-Lifetime problem with probability at least $1 - o(n^{-1})$.*

Proof. From the construction of the algorithm, the length of the schedule $\mathcal{S}$ is at least $b/2$. Afterwards, all nodes execute lines 5 and 6 of the algorithm. By Lemma 4.2, we know that the color classes C_i form correct dominating sets for $i \in [0, \ldots, \delta_v/(3\ln n)]$. Together with the above observation, it follows that the schedule $\mathcal{S}$ has length at least

$$L(\mathcal{S}) \geq \frac{b}{2} + \left\lfloor \frac{\delta}{(3k\ln n)} \right\rfloor \cdot \frac{b}{2}.$$

We now distinguish two cases. If $\delta/3\ln n \geq k$, then the approximation ratio α is

$$\begin{aligned}\alpha &\leq \frac{b \cdot (\delta + 1)/k}{\lfloor \delta/(3k\ln n) \rfloor \cdot b/2 + b/2} \\ &\leq \frac{2(\delta + 1)/k}{\delta/(3k\ln n) - 1} \in O(\log n).\end{aligned}$$

In the second case $\delta/3\ln n < k$, α is bounded as

$$\begin{aligned}\alpha &\leq \frac{b \cdot (\delta + 1)/k}{\lfloor \delta/(3k\ln n) \rfloor \cdot b/2 + b/2} \\ &\leq \frac{2(\delta + 1)}{k} \in O(\log n).\end{aligned}$$

This concludes the proof of Theorem 6.2. □

7. Conclusions and Open Problems

In this paper, we have given approximation algorithms for the problem of maximizing the lifetime of dominating set based clusterings. One technical open question is to come up with an approximation algorithm for the general k-tolerant case. Furthermore, there appear to be numerous unanswered questions going beyond this specific open problem.

There are applications in wireless ad hoc or sensor networks for which mere dominating sets are not the desired structure. In particular, it is often required that the dominating set fulfil the additional requirement that it is connected. The first and foremost such application is routing. It has been shown that the efficiency of routing can be enhanced by using methods that are based on building a *connected dominating set* in the network graph, e.g. [10, 1, 22]. As with regular dominating sets, maximizing the lifetime of connected dominating sets is important in practice. It is an intriguing open problem to come up with an approximation algorithm for the *Maximum Lifetime Connected Dominating Set* (or *maximum connected domatic partition*) problem. The difficulty of finding an approximation algorithm for the maximum connected domatic partition problem highlights a fundamental difference between the two problems, dominating set and domatic partition. Constructing a connected dominating set from a dominating set is straightforward and easy. However, extending a given domatic partition to a connected domatic partition appears to be highly non-trivial.

In practice, it may not always be possible to assume that the number of nodes n in the network (or an upper bound thereof) is known to all nodes. Therefore, getting rid of the assumption that n is known is another open and challenging problem.

Considering the practical potential of wireless ad hoc and sensor networks and in light of the need to deal with scarce energy-resources, the search for energy-efficient algorithms (and particularly approximation algorithms with concise worst-case guarantees) appears to be an interesting area for future research.

8. Acknowledgements

We would like to thank Ittai Abraham and Aaron Zollinger for fruitful discussions.

References

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks. In *Proc. of the 3^{rd} ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, pages 157–164, EPFL Lausanne, Switzerland, 2002.

[2] M. Cardei, D. Maccallum, X. Cheng, M. Min, X. Jia, D. Li, D.-Z. Du, and D. Du. Wireless Sensor Networks with Energy Efficient Organization. *Journal of Interconnection Networks*, 3(3-4):213–229, 2002.

[3] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit Disk Graphs. *Discrete Mathematics*, 86:165–177, 1990.

[4] U. Feige. A Threshold of ln n for Approximating Set Cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[5] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM Journal of Computing*, 32(1):172–195, 2003.

[6] S. Fujita. On the Performance of Greedy Algorithms for Finding Maximum r-Configurations. In *Proceedings of the Korea-Japan Joint Workshop on Algorithms and Computation (WAAC)*, 1999.

[7] S. Fujita, M. Yamashita, and T. Kameda. A Study on r-Configurations—A Resource Assignment Problem on Graphs. *SIAM Journal on Discrete Mathematics*, 13(2):227–254, 2000.

[8] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete Mobile Centers. In *Proc. 17^{th} Symposium on Computational Geometry (SCG)*, pages 188–196. ACM Press, 2001.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[10] S. Guha and S. Khuller. Approximation Algorithms for Connected Dominating Sets. In *Proc. of the 4^{th} Annual European Symposium on Algorithms (ESA)*, volume 1136 of *Lecture Notes in Computer Science*, pages 179–193, 1996.

[11] L. Jia, R. Rajaraman, and R. Suel. An Efficient Distributed Algorithm for Constructing Small Dominating Sets. In *Proc. of the 20^{th} ACM Symposium on Principles of Distributed Computing (PODC)*, pages 33–42, 2001.

[12] R. M. Karp. Reducibility Among Combinatorial Problems. In *Proc. of a Symposium on the Complexity of Computer Computations*, pages 85–103, 1972.

[13] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing Newly Deployed Ad Hoc and Sensor Networks. In *Proceedings of the 10^{th} Int. Conf. on Mobile Computing and Networking (MOBICOM)*, 2004.

[14] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What Cannot be Computed Locally! In *Proceedings of the 23^{rd} ACM Symposium on the Principles of Distributed Computing (PODC)*, 2004.

[15] F. Kuhn and R. Wattenhofer. Constant-Time Distributed Dominating Set Approximation. In *Proceedings of 22^{nd} ACM Int. Symposium on the Principles of Distributed Computing (PODC)*, pages 25–32, 2003.

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

[17] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.

[18] C. Lund and M. Yannakakis. On the Hardness of Approximating Minimization Problems. *Journal of the ACM*, 41:960–981, 1994.

[19] R. Prakash. Unidirectional Links Prove Costly in Wireless Ad-hoc Networks. In *Proceedings of the 3^{rd} International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, pages 15–22, 1999.

[20] S. Slijepcevic and M. Potkonjak. Power Efficient Organization of Wireless Sensor Networks. In *Proceedings of IEEE Infocom*, pages 472–276, 2001.

[21] P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *ACM/Baltzer Mobile Networks and Applications Journal*, 9(2):141–149, 2004.

[22] J. Wu and H. Li. On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks. In *Proc. of the 3^{rd} Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, pages 7–14, 1999.