A Generalized Clustering Algorithm for Peer-to-Peer Networks

Stefano Basagni Imrich Chlamtac András Faragó

Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
E-mail: {basagni,chlamtac,farago}@utdallas.edu

Abstract

A greedy approach is investigated for efficiently clustering the nodes of a mobile wireless network that has a "peer-to-peer" (i.e., non-cellular) organization. The considered algorithm is a generalization of different solutions proposed to this problem so far in the literature. We define the worst case performance ratio ρ of the algorithm in a way that reflects how the algorithm performs compared to the theoretical optimum. We show that ρ is nontrivially bounded from below by a compactly expressed network parameter. Moreover, we prove that the algorithm is optimal in the sense that it cannot be outperformed by any polynomial-time algorithm, provided $\mathsf{P} \neq \mathsf{NP}$. Simulation results are also presented to evaluate stability issues related to clustering in peer-to-peer networks.

1 Introduction

Peer-to-peer networks are wireless radio networks that consist of a set of geographically dispersed nodes, stationary or mobile, in which the nodes communicate with each other directly, without a wired network of base stations (cellular organization). These networks play an important role in situations where no fixed infrastructure is available or is not reasonable to build. These include, tactical battlefield communications, mobile computing in areas without infrastructure, ad hoc networking, law enforcement, disaster recovery, as well as administration and control of large events (e.g., sports, entertainment), etc.

One of the most important organizational problems in peer-to-peer networks is the reduction of routing and other control information overheads required for an autonomous organization mechanism in face of node mobility. Recent research in the field shows that a solution to this problem can be obtained via the organization

of nodes in groups, or *clusters* (see, e.g., [11, 12]). Therefore, an important design problem in peer-to-peer networks is *efficient clustering* of the mobile nodes. Since the network topology may change rapidly, it is also important that clusters are formed quickly.

In the present paper we address the clustering problem of peer-to-peer networks. The methods used so far for obtaining physical clustering in these networks all implement some type of *greedy algorithm* for finding a set of nodes that act as *clusterheads*. Once the clusterheads are selected, clusters are defined by associating each non-clusterhead node with, for instance, the closest clusterhead. To achieve easy organization it is desirable in this case that this closest clusterhead is reachable in a single hop, that is, the clusterheads should form a *Dominating Set* in the graph, i.e., each non-clusterhead node must have at least one clusterhead neighbor. On the other hand, it is also desirable that the network be covered by a "well scattered" set of clusterheads, so that they do not group in a small part of the network. This can be captured by the requirement that the clusterheads form an *Independent Set* of nodes in the graph (i.e., we require that no two clusterheads are neighbors).

Existing solutions all implement simple greedy algorithms for finding such dominating independent set of clusterheads. In the "largest degree first" approach of Gerla et al. [5], a node with maximum *degree* (i.e., with the maximum number of neighbors) is selected as a clusterhead, then removed from the graph, together with its neighbors. The same procedure is repeated in the remaining graph, until all nodes are removed. In the "lowest ID first" method, used in Ephremides et al. [1] and in Gerla et al. [3, 4], node IDs are used to choose the clusterheads. One can easily define similar greedy clusterhead selection algorithms based on other parameters of the nodes, as well. Usually, however, these algorithms remain a purely heuristic level solution, and no performance guarantee have been proven.

The common model of all the above algorithms is the greedy search for a *Maximum Weight Independent Set* (MWIS) in a graph, where nonnegative weights are associated with the nodes. These weights are, of course, the degree of a node in the "largest degree first" approach, and the nodes' IDs in the "lowest ID first" approach. In the present paper, after the definition of a natural model for peer-to-peer networks (Section 2), we generalize the existing solutions by associating each node with a generic nonnegative weight (Section 3). For the resulting generalized clustering algorithm we show that it is possible to prove a nontrivial performance guarantee. Such a guarantee, of course, also applies to the special versions mentioned above. Specifically, in Section 4, we define the *quality* of a clustering algorithm as a measure, ρ, of how the algorithm performs compared to the theoretical optimum. We show that ρ has a nontrivial lower bound that depends on a global network parameter. Furthermore, in Section 5, we show that, despite its extreme simplicity, the greedy algorithm here is essentially the best we can do, given that $\mathsf{P} \neq \mathsf{NP}$. More precisely, we show that for any class of graphs for which the Maximum cardinality Independent Set (MIS) problem is NP-hard, it is also NP-hard to *outperform* the

greedy MWIS search in a well defined sense.

Note that by associating generic weights with the nodes of a peer-to-peer network, we obtain more than a simple unification of existing solutions in a common framework that allows us to prove theoretical properties. Indeed, the proposed generalized algorithm makes it possible to express *preferences* through the choice of weights. This mechanism, not available in previously proposed solutions, can be practically used to choose the clusterheads depending on the motion rate of each node, its power, its degree, or a combination of these parameters. As an example of its application, in Section 6 we demonstrate by simulations that, when the weights reflect the speed of the nodes, our proposed algorithm achieves a substantial improvement over the "lowest ID first" approach.

2 Preliminaries

It is natural to model a *peer-to-peer* network by an undirected graph $G = (V, E)$ in which V, $|V| = n$, is the set of (radio) nodes and there is an edge $\{u, v\} \in E$ if and only if u and v can mutually receive each others' transmission. In this case we say that u and v are neighbors. Due to mobility, the graph can change in time.

The set of the neighbors of a node $v \in V$ will be denoted by $\Gamma(v)$ and its cardinality, $\delta(v)$, is said to be the *degree* of v. The degree Δ of the entire network is defined as the maximum among all the degrees in the graph, namely, $\Delta = \max\{\delta(v) : v \in V\}$. We do not consider networks with no edges, so that we always have $\Delta > 0$. Every node v in the network is assigned a unique identifier (ID), denoted by the numbers $1, \ldots, n$. For simplicity, the node is identified with its ID and both are denoted by v. Finally, we consider weighted networks, i.e., a weight w_v (a real number ≥ 0) is assigned to each node $v \in V$ of the network. The weights are collected in the weight vector $w = \langle w_1, \ldots, w_n \rangle$. As an example, the topology of a simple peer-to-peer network is shown in Figure 1.

Given a graph $G = (V, E)$, an *Independent Set* (IS) is a set $V' \subseteq V$ such that no two nodes in V' are joined by an edge in E. It is well known that it is computationally difficult to decide whether a graph has an IS of a given cardinality, namely, the following *decision* problem:

INDEPENDENT SET
Input: A graph $G = (V, E)$, and a positive integer $k \leq |V|$.
Question: Does G contain an independent set of size k or more?

is *NP-complete*. Thus, the search for an IS of *maximum* cardinality (a *Maximum Independent Set*, MIS) is *NP-hard*. (For a clear introduction to the theory of NP-completeness the reader is referred to the classical book of Garey and Johnson, [2].) In the case of (node) weighted graphs, the *Maximum Weight Independent Set*

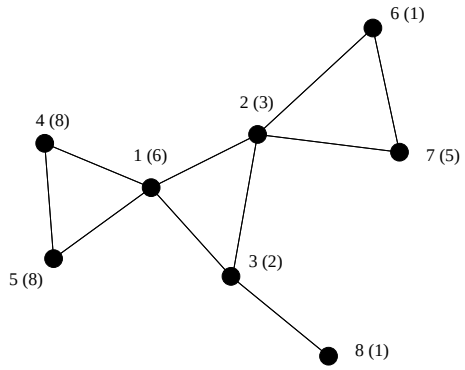

Figure 1: A peer-to-peer network with nodes v and their weights (w_v), $1 \leq v \leq 8$.

(MWIS) problem consists in finding an independent set of nodes, the sum of whose weights is as big as possible. It is clear that, being the MIS problem a special case of the MWIS problem, this latter one is NP-hard as well. Finally, a *Dominating Set* (DS) of nodes of a graph $G = (V, E)$ is a set $V' \subseteq V$ such that for all $u \in V \setminus V'$ there is a v in V' fro which $\{u, v\} \in E$. The search for a DS in general graphs is also an NP-Hard problem.

3 The Clustering Algorithm

As already mentioned in the Introduction, existing practical clusterhead selection algorithms do a greedy search for a *Maximum Weight Independent Set* (MWIS) in the network (see, e.g., [1, 3, 5]). They differ only in terms of the basis used for the greedy choice, such as "largest degree" or "lowest ID," etc. To provide a common analysis for all these algorithms, we examine the following natural generalization. The proposed algorithm is similar to the "larger degree first" and to the "lowest ID first" approaches, but instead of the degree or the node's ID, the greedy choice acts on the basis of arbitrary nonnegative weights associated to the nodes. (Ties are broken arbitrarily, e.g., by using nodes' IDs—we choose to take the node with the lowest ID.) Thus, through the choice of appropriate weights, we can express in a completely general way the *preference* to have a given node as a clusterhead. It is easy to see that both the "largest degree first" and "lowest ID first" approaches are special cases of the algorithm.[1] Being a common generalization of previous clusterhead selection algorithms, we call the proposed algorithm the *Generalized Clustering Algorithm* (GCA), defined as follows.

Procedure GCA
(**input**: $G = (V, E)$: network, w: weights; **output**: $\{C_i\}_{i \in I \subset V}$: clustering);

[1] For each $v \in V$, it is enough to define $w_v = \delta(v)$ and $w_v = 1$, respectively.

```
begin
    i := 0;
    while V ≠ ∅ do
        begin
            i := i + 1;
            (* Pick the node with the lowest ID among those with maximum
            weight: *)
            v := min{u ∈ V : w_u = max{w_z : z ∈ V}};
            C_i := {v} ∪ Γ(v);
            V := V \ C_i
        end
end;
```

It is evident that the set of all the nodes v selected in each iteration of the main loop is a dominating independent set and that, as far as $V \neq \emptyset$, its cardinality is the final value of i. The application of the procedure to the network of Figure 1 is illustrated in Figure 2 (where the squared nodes are the clusterheads).

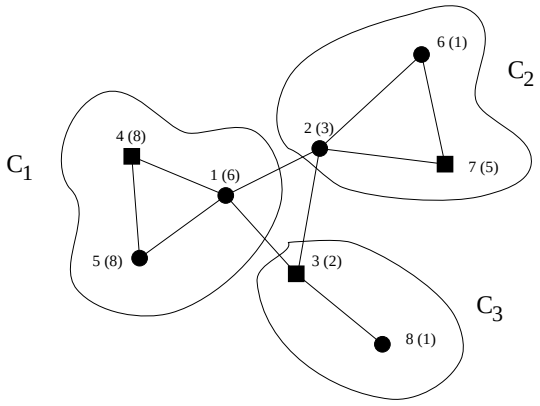

Figure 2: The clustering obtained by applying the GCA to the network of Figure 1.

4 Performance Guarantee

This section investigates the question of whether it is possible to prove a nontrivial performance guarantee about the proposed algorithm. We propose to measure the *quality* of an algorithm in the following way. Let $\mathcal{A}$ be any clusterhead selection algorithm and run it on a network (graph) $G = (V, E)$, $|V| = n$, with node weighting w. (We use the terms *network* and *graph* interchangeably). Denote by $\mathcal{A}(G, w)$ the weight of the obtained independent set. Further, let $\alpha(G, w)$ be the weight of a

MWIS, which would be the ideal solution. The ratio

$$\rho_{\mathcal{A}}(G, w) = \frac{\mathcal{A}(G, w)}{\alpha(G, w)} \tag{1}$$

characterizes how well algorithm $\mathcal{A}$ performs on the network G with weighting w, compared to the ideal optimum. Since the result of $\mathcal{A}$ cannot exceed the maximum, therefore, $\rho_{\mathcal{A}}(G, w)$ cannot exceed 1, and generally the higher, the better.

The ratio (1) for a given network G depends, of course, on the weighting at hand. To provide a *performance guarantee* for a given network, i.e., a value of (1) that is *always* guaranteed when running algorithm $\mathcal{A}$ on G, we define the *worst case performance ratio* $\rho_{\mathcal{A}}(G)$ of algorithm $\mathcal{A}$ on network G by taking the minimum of (1) over all possible choices of the weighting. Since the number of possible weight vectors is infinite, this can be expressed by the infimum operator:

$$\rho_{\mathcal{A}}(G) = \inf_{w \geq 0} \frac{\mathcal{A}(G, w)}{\alpha(G, w)}. \tag{2}$$

This assigns a unique number to the algorithm on each network. If we can bound it from below, then a *performance guarantee* is provided for the algorithm. In the remaining part of this section, we show that it is possible to bound the worst case performance ratio of the GCA from below by a network parameter, thus providing a nontrivial performance guarantee. To do this, we need to introduce some concepts.

If $G = (V, E)$ is a network, and $v \in V$ is a node of G, let us denote by G_v the *neighborhood graph* of v. This is the induced subgraph spanned by v and its neighbors. In other words, G_v is obtained if all nodes are deleted from G, except v and its neighbors (Figure 3).

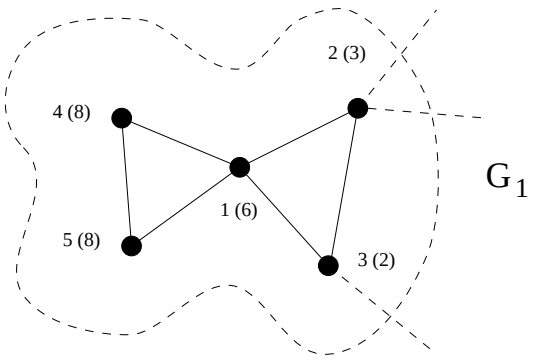

Figure 3: The neighborhood graph G_1 of node 1 in the network of Figure 1.

The independence number (sometimes also called stability number) $\alpha(G)$ of the network G is the maximum size of an independent set (also called stable set) in G.

(It is the special case of $\alpha(G, w)$, obtained by setting each weight to 1.) Let us introduce the *local independence number* of G by

$$\beta(G) = \max_{v \in V} \alpha(G_v).$$

The local independence number tells what is the maximum possible size of an independent set that occurs in the neighborhood of any node. It is obvious that $\beta(G)$ cannot be larger than the size of the entire neighborhood, that is, $\beta(G) \leq \Delta$.

Using the local independence number, it is possible to state a compact result on the worst case performance ratio of our GCA.

Theorem 1 *For any network G and for any nonnegative weighting w, the worst case performance ratio of the GCA is bounded from below as*

$$\rho_{\mathrm{GCA}}(G) \geq \frac{1}{\beta(G)}. \tag{3}$$

Remark. It is worth noticing that, since $\beta(G) \leq \Delta$ always holds, it is obtained as a direct corollary that

$$\rho_{\mathrm{GCA}}(G) \geq \frac{1}{\Delta}. \tag{4}$$

We also notice that both (3) and (4) are nontrivial because the bounds hold for arbitrary weighting, while both $\beta(G)$ and Δ are independent of the weights.

Proof of Theorem 1. We use results on the analysis of general *independence systems* from [7]. An independence system consists of a finite set S (the base set), and a family $\mathcal{F}$ of subsets of S. The family $\mathcal{F}$ is assumed closed under taking subsets. That is, $A \in \mathcal{F}$ and $B \subseteq A$ implies that $B \in \mathcal{F}$. The independent sets of nodes in a graph $G = (V, E)$ clearly satisfy this property (any subset of an independent set is independent), so this is a special independence system with base set V.

The greedy heuristic for finding a "heavy" independent set in a general independence system with nonnegative weights on the elements works as follows:

- Order the elements of the base set S according to nondecreasing weight (ties are broken arbitrarily). Let $s_1, \ldots, s_n$ be such an ordering, with $w(s_1) \geq \ldots \geq w(s_n)$.
- Find a set $F \in \mathcal{F}$ iteratively by the following algorithm:

  $F := \emptyset$;
  for $i = 1$ to n do
  if $F \cup \{s_i\} \in \mathcal{F}$ **then** $F := F \cup \{s_i\}$.

The result we use from [7] can be stated as follows. Let F_g be an independent set found by this greedy algorithm and let F_0 be a maximum weight independent set (the optimum). Then, for any independence system, we have:

$$1 \geq \frac{w(F_g)}{w(F_0)} \geq r(\mathcal{F}), \tag{5}$$

where $w(F)$ indicates the weight of the independent set F and $r(\mathcal{F})$ is the *rank ratio* of the independence system. The rank ratio is defined by using the *upper rank* and *lower rank* of subsets of the base set S. For any non empty $S' \subseteq S$ the upper rank $ur(S')$ is the maximum size of an independent set that occurs in S'. The lower rank $lr(S')$ is the size of the smallest such independent set in S' which already cannot be extended in S'. (In other words, such a set is the smallest dominating independent set in S'.) Then the rank ratio is defined as

$$r(\mathcal{F}) = \min_{S' \subseteq S} \frac{lr(S')}{ur(S')},$$

where the minimum is taken over all nonempty subsets of S.

Let us compute now the rank ratio of the special independence system formed by the family $\mathcal{F}$ of independent sets of nodes in a graph $G = (V, E)$. Let $V' \subseteq V$ be any (non empty) subset of nodes. Let A be a maximum size independent set in V' and let B be a minimum size dominating independent set in V'. Let us define the following function $f : B \to 2^A$:

$$f(v) = \begin{cases} \{v\} & v \in A, \\ \\ \Gamma(v) \cap A & v \notin A. \end{cases}$$

We have:

$$\bigcup_{v \in B} f(v) = (A \cap B) \cup \bigcup_{v \in B \setminus A} (\Gamma(v) \cap A) = (B \cup \bigcup_{v \in B \setminus A} \Gamma(v)) \cap A.$$

Now, for each $v' \in A$, either $v' \in B$ or, if $v' \notin B$, then $v' \in \cup_{v \in B \setminus A} \Gamma(v)$ (otherwise, v' not having any neighbor in B, $B \cup \{v'\}$ would be an independent set, contradicting the maximality of B). Thus:

$$A \subseteq (B \cup \bigcup_{v \in B \setminus A} \Gamma(v)),$$

whence

$$A = \bigcup_{v \in B} f(v).$$

Therefore, we have:

$$|A| = \left|\bigcup_{v\in B} f(v)\right| = \left|\bigcup_{v\in A\cap B} f(v)\right| + \left|\bigcup_{v\in B\setminus A} f(v)\right| \le \sum_{v\in A\cap B} |\{v\}| + \sum_{v\in B\setminus A} |\Gamma(v)\cap A|.$$

Since A is a maximal independent set in V', therefore, for each $v \in B\setminus A$, $|\Gamma(v)\cap A| > 0$ must hold, otherwise A could be extended in V' without violating its independence. Moreover, since $\Gamma(v)\cap A$ is an independent set, we have $|\Gamma(v)\cap A| \le \alpha(G_v)$, yielding

$$\begin{aligned}\sum_{v\in A\cap B} |\{v\}| + \sum_{v\in B\setminus A} |\Gamma(v)\cap A| &\le |A\cap B| + (|B| - |A\cap B|)\max_{v\in B\setminus A}\alpha(G_v)\\ &\le (|A\cap B| + |B| - |A\cap B|)\max_{v\in V'}\alpha(G_v)\\ &\le |B|\,\beta(G).\end{aligned}$$

Thus, by $\beta(G) > 0$, we have:

$$\frac{lr(V')}{ur(V')} = \frac{|B|}{|A|} \ge \frac{1}{\beta(G)},$$

which yields

$$r(\mathcal{F}) = \min_{V'\subseteq V}\frac{lr(V')}{ur(V')} \ge \frac{1}{\beta(G)}.$$

This, using (5), implies

$$\rho_{GCA}(G) \ge \frac{1}{\beta(G)},$$

which proves the theorem. •

5 Optimality

Here we show that our GCA is optimal in a sense defined below, i.e., that it is the best possible heuristics we can use, given that $P \neq NP$. To make this precise, we need to define the method for comparing two algorithms (by algorithm we refer here an algorithm that finds a dominating independent set).

Definition 1 *Let $\mathcal{G}$ be any class of graphs. Algorithm $\mathcal{A}$* outperforms *algorithm $\mathcal{B}$ on the class $\mathcal{G}$ if $\mathcal{A}(G,w) \ge \mathcal{B}(G,w)$ holds for any ($G \in \mathcal{G}$, w) and, furthermore, the inequality is strict whenever $\mathcal{B}(G,w) < \alpha(G,w)$.*

Informally speaking, algorithm $\mathcal{A}$ outperforms $\mathcal{B}$ on $\mathcal{G}$ if the result of $\mathcal{A}$ is never worse than that of $\mathcal{B}$, and $\mathcal{A}$ has a strictly better result whenever $\mathcal{B}$ does not find the optimum. Of course, in cases when $\mathcal{B}$ does find the optimum, the inequality cannot be strict, since in those cases there is already no room for improvement.

Now we can state what it is meant by the optimality of the GCA heuristics.

Theorem 2 *Let $\mathcal{G}$ be any class of graphs in which it is NP-hard to find a Maximum cardinality Independent Set. Then, assuming $\mathsf{P} \neq \mathsf{NP}$, no polynomial-time algorithm can outperform the GCA on $\mathcal{G}$ in the sense of Definition 1.*

Proof. Let us assume that there is a polynomial-time algorithm $\mathcal{A}$ that outperforms the GCA on $\mathcal{G}$. Let us consider now the following algorithm $\mathcal{A}'$ with any input graph $G = (V, E) \in \mathcal{G}$, $|V| = n$. In the definition of $\mathcal{A}'$, we use the notation χ_A for the *characteristic vector* of a set $A \subseteq V$, i.e., the $|A|$ coordinates of χ_A corresponding to the elements of A are set to 1, and the remaining $n - |A|$ are set to 0.

Algorithm $\mathcal{A}'$

Step 1. Set $w = \langle n, \ldots, n \rangle$.

Step 2. Run the GCA on G with weighting w to find a dominating independent set. Let A be the obtained independent set.

Step 3. Run algorithm $\mathcal{A}$ on G with weighting w. Let B be the obtained independent set. If $|B| = |A|$ then *STOP*.

Step 4. Set $A = B$ and $w = \langle n, \ldots, n \rangle + \chi_B$;
Go to *Step 3.*

We claim that algorithm $\mathcal{A}'$ works in polynomial time and when it halts the last value of the set B is a Maximum cardinality Independent Set (MIS) in G.

Let us prove first the second statement, assuming the algorithm halts after a finite number of steps. Let B_0 be the last value of B, and let B_1 be the previous value of B. Then after the last execution of *Step 4* the value of w is $\langle n, \ldots, n \rangle + \chi_{B_1}$ and $A = B_1$. Under the weighting w each element of the dominating independent set B_1 has weight $n + 1$, while all other nodes have weight n. Observe that this weighting directly implies the following special properties:

(i) For any two sets $X, X' \subseteq V$ the inequality $w(X) > w(X')$ can hold if and only if $|X| > |X'|$.

(ii) The GCA on G with this weighting would return B_1 as its result.

Now, in the last execution of *Step 3* algorithm $\mathcal{A}$ returns the set B_0. Since, by assumption, $\mathcal{A}$ outperforms the GCA, therefore, by *(ii)* $w(B_0) \geq w(B_1)$ must hold. If $w(B_0) > w(B_1)$ is the case, then by *(i)* it implies $|B_0| > |B_1|$, which by $A = B_1$ contradicts to the assumption that this is the last execution of *Step 3*. Thus, $w(B_0) = w(B_1)$ must hold. Then B_1 must be a MWIS (otherwise $\mathcal{A}$ would return a set with strictly larger weight) and then by *(i)* it is also a MIS and $\mathcal{A}'$ halts with $|B_0| = |B_1|$, implying that B_0 is a MIS.

It remains to show that $\mathcal{A}'$ halts after a polynomial number of steps. This is a consequence of the fact that, as directly implied by the above considerations, in each execution of *Step 3*, except the last one, we must always obtain an independent set of strictly larger weight than the previous one. Since each node has weight at most $n+1$ and each weight is an integer, therefore, *Step 3* can be executed only polynomially many times. Since, by assumption, algorithm $\mathcal{A}$ runs in polynomial time and the other parts of algorithm $\mathcal{A}'$ clearly run in polynomial time, too, therefore, the whole algorithm $\mathcal{A}'$ halts after a polynomial number of steps.

Thus, we have shown that the assumption that there exists a polynomial-time algorithm $\mathcal{A}$ that outperforms the GCA on $\mathcal{G}$ implies that algorithm $\mathcal{A}'$ can find a MIS in any graph $G \in \mathcal{G}$ in polynomial time. This, however, if $P \neq NP$, contradicts to the condition that finding an MIS is NP-hard in $\mathcal{G}$, which proves the theorem. •

Theorem 2 shows that the GCA (i.e., the greedy heuristics) is the best possible algorithm in the defined sense, whenever we have to deal with a class of graphs for which the MIS problem is NP-hard. Interestingly, for these classes, there is nothing that lies between the trivial greedy heuristics and the non-polynomial algorithms in the sense that, according to Theorem 2, it is impossible to outperform the GCA by a polynomial-time algorithm on these classes. This relies, of course, on the strong definition of outperforming.

Although there are several classes of graphs for which the MWIS problem can be efficiently solved (see, e.g., the survey in [9]), nevertheless, for general networks the problem is not only NP-hard, but even hardly approximable (see, e.g., [6]). Since we typically deal here with a mobile radio network environment, therefore, we cannot realistically assume that the more or less arbitrarily changing network topology can be forced to remain within a class in which the considered NP-hard algorithmic problem is polynomially solvable. This is why the greedy approach plays a central role here, taking into account its simplicity and speed. Moreover, its goodness is highlighted not only by the analysis we have provided, but also by other results in [10] (where several greedy solutions are devised and compared) and in [8].

6 Simulation Results

We conclude this paper with a simulation example that shows the practical importance of node weight setting in a mobile network. We demonstrate that, with properly chosen weights, the GCA can achieve substantial improvement over the "lowest ID first" approach with respect to the *stability* of the network clustering.

We simulate the clustering algorithm by placing $n = 40$ nodes randomly on a grid of size 100×100, and two nodes are neighbors if they are within a fixed distance to each other. We are interested in measuring the stability of the clustering algorithm in a changing network. We count how many nodes move to a new cluster

("reaffiliations") within a time-unit and how many ordinary nodes (i.e., nodes that are not clusterheads) become clusterheads in each time-unit ("elections"), after re-clustering to adjust for random node movements in each tick. At confidence level of 99% , all simulation results are in the confidence interval of less than 1%. Each node moves in a random direction defined by an angle that is independently and uniformly distributed over the interval $(0, 2\pi)$. The speed of a node is a randomly and uniformly chosen integer in the interval $[1, k]$, where k is the *maximum speed* of all nodes, $1 \leq k \leq 100$. The simulation is run for different values of the maximum speed.

We compare two clusterhead selection algorithms in this model with respect to the defined stability parameters. One algorithm is the "lowest ID first" approach presented in [3, 4, 5]. (As in [5] and [4] it is shown that this algorithm is more stable than the "largest degree first" approach, we do not consider the latter one.)

The other algorithm is our GCA with weights that depend on the speed of the node. If $s(v)$ is the (randomly chosen) speed of a node v, we define its weight as $w_v = k+1-s(v)$. That is, for any fixed maximum speed, the nodes that are moving at a lower speed are assigned higher weight, because these "more quiet" nodes are more likely to yield higher stability.

Figures 4 and 5 show the average number of elections (new clusterheads) and reaffiliations per time-unit, respectively, as functions of the maximum speed, for both algorithms. It can be seen from the figures that the GCA with speed dependent weighting outperforms the other algorithm. The percentage gain is in the range $25 \ldots 45\%$, as shown in Figure 6.

7 Conclusion

In this paper we examined the greedy approach for the clustering of peer-to-peer radio networks. We have proven a performance guarantee for the GCA, a generalization of previous algorithms that have been applied for this problem. Moreover, we have also proven the optimality of this greedy approach in the sense that it is NP-hard to outperform it by a polynomial-time algorithm. The practical importance of the general setting is that, as opposed to other solutions, such as "lowest ID first" or "largest degree first," the GCA makes it possible to express preferences through the choice of the weights. For instance, we have demonstrated by simulation that properly chosen speed-dependent weights result in less likely change of clusters, thus reducing the amount of reconfiguration overhead and the time when nodes are not available for communication due to reorganization. To achieve even more refined results, a combined weight of node degree, motion and power can be set to account for multi-parameter optimization, not available in existing practical solutions.

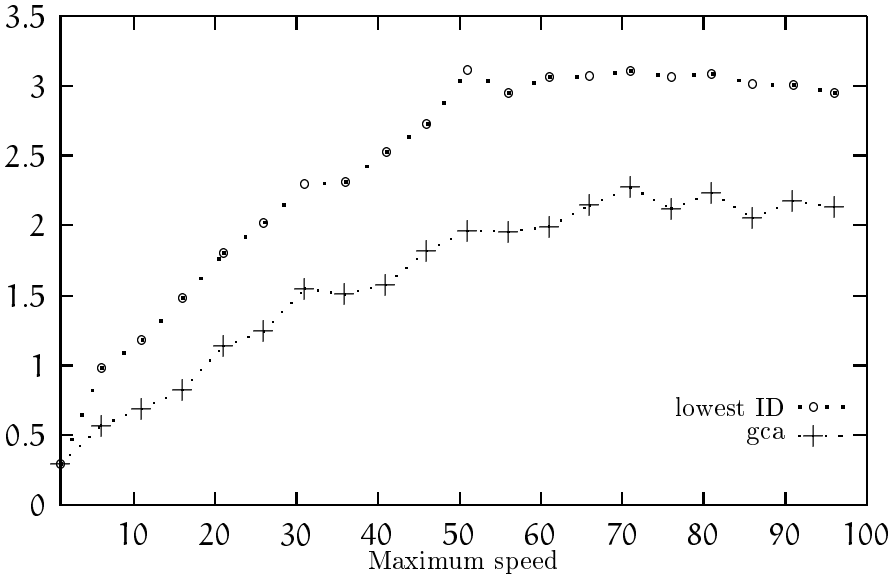

Figure 4: Elections per tick (average).

References

[1] Ephremides, A., Wieselthier, J. E., and Baker, D. J. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE 75*, 1 (January 1987), 56–73.

[2] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, CA, 1979.

[3] Gerla, M., and Lin, C. R. Multimedia transport in multihop dynamic packet radio networks. In *Proceedings of International Conference on Network Protocols* (Tokyo, Japan, 7–10 November 1995), pp. 209–216.

[4] Gerla, M., and Lin, C. R. Adaptive clustering for mobile wireless networks. *Journal on Selected Areas in Communications 15*, 7 (September 1997), 1265–1275.

[5] Gerla, M., and Tsai, J. T.-C. Multicluster, mobile, multimedia radio network. *Wireless Networks 1*, 3 (1995), 255–265.

[6] Håstad, J. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science* (Burlington, VT, 14-16 October 1996), pp. 627–636.

[7] Hausmann, D., and Korte, B. An analysis of the greedy heuristic for indepedence systems. *Annals of Discrete Mathematics 2* (1978), 65–74.

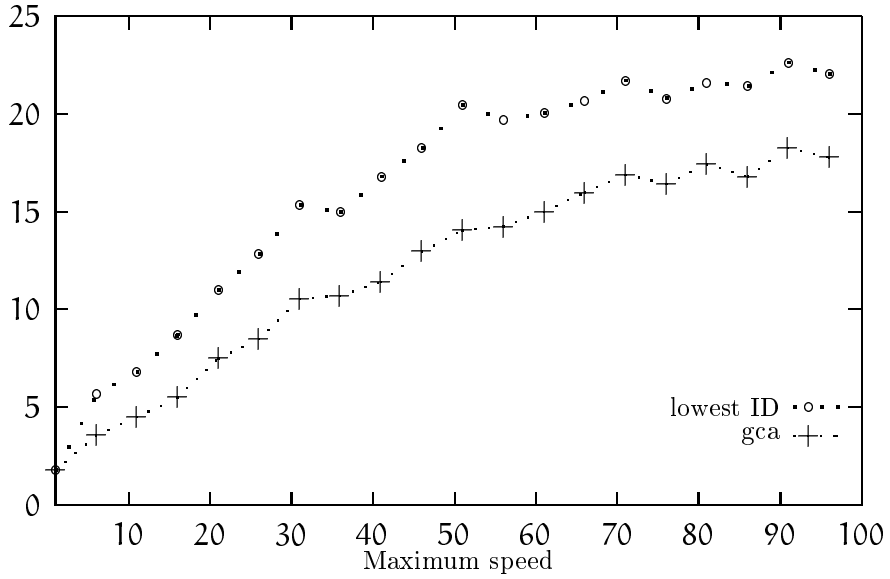

Figure 5: Reaffiliations per tick (average).

[8] Hausmann, D., and Korte, B. k-greedy algorithms for independence systems. *Zeitschrift fur Operations Research, Serie A (Theorie) 22*, 5 (October 1978), 219–228.

[9] Johnson, D. S. The NP-completeness column: an ongoing guide. *Journal of Algorithms 6*, 3 (1985), 434–451.

[10] Kopf, R., and Ruhe, G. A computational study of the weighted independent set problem for general graphs. *Foundations of Control Engineering 12*, 4 (1987), 167–180.

[11] Lauer, G. S. Hierarchical routing design for suran. In *Proceedings of the IEEE International Conference on Communications '86. ICC '86* (Toronto, Ont., Canada, 22-25 June 1986), vol. 1, pp. 93–102.

[12] Lauer, G. S. Packet-radio routing. In *Routing in Communication Networks*, M. E. Steenstrup, Ed. Prentice Hall, Englewood Cliffs, N.J., June 1995.

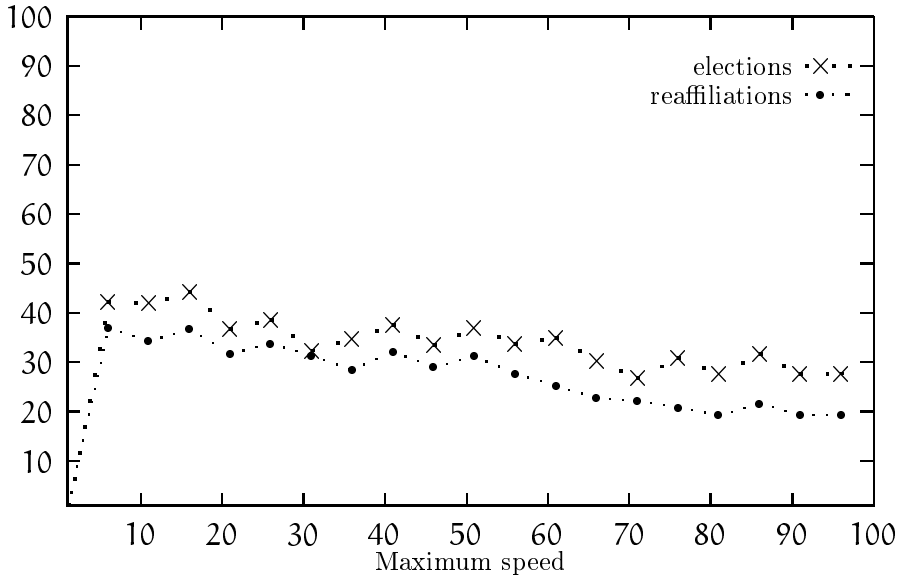

Figure 6: GCA vs. "lowest ID first": percentage gain for elections and reaffiliations.