Proceedings of the IASTED International Conference
Parallel and Distributed Computing and Systems (PDCS '99)
November 3-6, 1999, Cambridge, Massachusetts - USA

A Distributed Algorithm for Finding a Maximal Weighted Independent Set in Wireless Networks*

Stefano Basagni
Center for Advanced Telecommunications Systems and Services (CATSS)
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
E-mail: `basagni@utdallas.edu`

Abstract

This paper introduces *MWIS*, a distributed algorithm for the efficient determination of a Maximum Weighted Independent Set in the topology graph G of a wireless network. Motivated by the observation that the problem of partitioning wireless nodes into *clusters* easily reduces to the problem of finding a maximum weighted independent set of nodes, the proposed algorithm is described by taking into account two main characteristic of wireless networks, namely, the broadcast nature of the wireless medium and the possibility to support nodes mobility. MWIS is executed at each node by means of fast message-triggered procedures that require the sole knowledge of the topology local to the node. Moreover, its time complexity is proven to be bounded by the (topology dependent) *stability number* $\alpha(G)$, rather than by the (invariant) number n of the network nodes. Finally, by using a well known results on the stability number in *random graphs*, we show that the average time complexity of the MWIS is logarithmic in n.

Key Words: maximal weighted independent set, wireless mobile networks, distributed algorithms.

1 Introduction

An *independent set* in a graph $G = (V, E)$ is a set $I \subseteq V$ such that there is no pair of nodes in I linked by an edge in E. The *maximal independent set problem* is to find an independent set that is not properly contained in any other independent set. This problem admits a natural generalization to graphs in which each node is associated with a weight (a real number ≥ 0), and in this case we want to find a maximal independent set by choosing the nodes of the set so to maximize the total weight (*maximal weighted independent set problem*). Of course, a *maximum* weight independent set in G (i.e., the independent set with the biggest weight among all possible independent sets in G) would solve our problem, but this is a well known NP-hard problem [1] for which approximation in polynomial time is NP-hard also (see, e.g., [2]). Thus, given that a polynomial time solution can be found for the maximal, rather than the maximum, weighted independent set problem, several fast and simple algorithms have been proposed to solve the "maximal version" using either sequential or parallel techniques (see, e.g., [3] and [4]). All these algorithms solve the maximal (generally non weighted) independent set problem in general undirected graphs.

In this paper we are interested in solving the maximal weighted independent set problem for that specific class of graphs that represent the topology of *wireless networks*. In particular, we present an algorithm that takes into account two of the main characteristics of these networks:

1. The *broadcast* nature of the wireless channel, which implies the reception of a message transmitted by a node at *all* the neighbors of that node.[1]

2. The fact that wireless networks allow the mobility of the nodes. This makes difficult to consider *centralized* solutions for the maximal weighted independent set problem and imposes the need for a decentralized (i.e., *distributed*) algorithm.

Motivations. Finding a maximal weighted independent set in a wireless network is strictly related to the problem of organizing the nodes of the network in a hierarchical way, a well-known and studied problem of distributed computing.

In the case of *wireless networks*, partitioning the nodes into groups (*clusters*) can be effectively used for controlling the spatial reuse of the shared channel (e.g., in terms of time division or frequency division schemes), for minimizing the amount of data to be exchanged in order to maintain routing and control information in a mobile environment, as well as for building and maintaining cluster-based *virtual* network architectures.

A host of *clustering* algorithms have been proposed for wireless networks since their appearance (see, e.g., [5] and [6] for further references). All these solutions implement some *greedy algorithm* for finding an independent set of nodes that act as *clusterheads*. (The use of greedy heuristic is motivated by the fact that the clustering problem is easily reducible to the computationally hard problem of finding a maximum independent set of nodes in the network graph.) The fact that no two clusterheads can be neighbors (i.e., the fact that they should form an independent set in the network graph) is motivated by the need to cover the network with a "well scattered"

* This work was supported in part by the Army Research Office (DARPA) under contract No. DAAG55-97-1-0312 and in part by the TxTEC Fellowship 1998/1999.

[1] Two nodes are *neighbors* if they can exchange messages, i.e., if they can hear each other transmissions.

set of clusterheads, so that each node in the network has a clusterhead in its neighborhood and it has a direct access to that clusterhead.

In [7] the characteristics of a good clustering for wireless networks are described which include the possibility of choosing a node for the role of clusterhead according to some node specific parameter such as its velocity or the transmission range. To this purpose, as proposed in that paper, a generic *weight* (a real number ≥ 0) is associated to each node to indicate how suitable a node is for the role of clusterhead. (This was not available in previous solutions where the choice was based either on the nodes' unique identifier or on the number of their neighbors.)

Therefore, the algorithm for selecting the clusterheads of a wireless network reduces to the problem of finding a maximal weighted independent set in the topology graph of the network. The nodes in the independent set will be the clusterheads. Here, the maximality guarantees that the clusterheads are spread out in the network so that they can properly coordinate all the other nodes (often called *ordinary nodes*). At the same time, the weight-based choice of the clusterheads insures that the most suitable nodes have been chosen for the role of clusterhead.

Our contribution. In this paper we describe the *MWIS* algorithm, a distributed algorithm for determining a maximal weighted independent set in the network topology graph of a wireless network. MWIS is distributed in the precise sense that it is executed at each node v of the network, and only the knowledge of the identity and of the weight of v's neighbors is required. In order to cooperate in the generation of a maximal weighted independent set, neighboring nodes exchange messages whose reception triggers the local execution of a corresponding procedure. The correctness of MWIS is proven, and for the first time its time complexity is bounded to a parameter that depends on the (possibly changing) topology of the network G, instead of on the invariant number of its nodes n. More specifically, we show that, by associating each node with a positive integer $\leq 2k$ (being k the cardinality of the maximal weighted independent set produced by MWIS), the time complexity of MWIS is always proportional to the graph *stability number* $\alpha(G)$, i.e., the cardinality of the biggest independent set in G. Finally, by using a well known result from the theory of *random graphs* [8], we show that the average time complexity of MWIS is (asymptotically) logarithmic in n. Our algorithm is thus fast, easy to implement and therefore constitutes an effective base for clusterhead selection (clustering) in wireless mobile networks.

The paper is organized as follows. Section 2 introduces the basic definition used throughout the paper. The MWIS distributed algorithm is described in Section 3. In the following Section 4 the correctness of MWIS is formally stated, and bounds are proven on its message and time complexity. Section 5 concludes the paper.

2 Preliminaries

We model a *wireless* network by an undirected graph $G = (V, E)$ in which V, $|V| = n$, is the set of (wireless) nodes and there is an edge $\{u, v\} \in E$ if and only if u and v can mutually receive each others' transmission (this implies that all the links between the nodes are bidirectional). In this case we say that u and v are neighbors. The set of the neighbors of a node $v \in V$ will be denoted by $\Gamma(v)$.

Every node v in the network is assigned a unique identifier (ID). For simplicity, here we identify each node with its ID and we denote both with 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. For the sake of simplicity, in this paper we stipulate that each node has a different weight (ties can be broken arbitrarily, for instance, by using the nodes' unique ID). As an example, the topology graph of a simple wireless network is shown in Figure 1 (a).

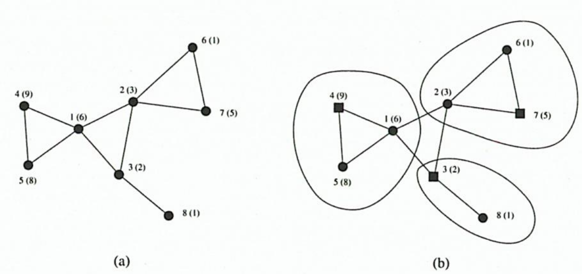

Figure 1: (a) A wireless network G with nodes v and their weights (w_v), $1 \leq v \leq 8$, and (b) a partition of G into clusters. Each "squared" node belongs to the maximal weighted independent set.

An *independent set* in a graph $G = (V, E)$ is a set $I \subseteq V$ such that there is no pair of nodes in I linked by an edge in E. The cardinality of the biggest among all independent sets in G is denoted by $\alpha(G)$ and it is often called the *stability number* of G. As mentioned in the Introduction, we are interested in finding a maximal weighted independent set, i.e., an independent set that is not properly contained in any other independent set of the graph. In selecting the nodes, we are interested in maximizing the sum of the weights.

Given the characteristics of a wireless network as listed in the previous section, in the description of the algorithm:

1. We assume that a message sent by a node is received correctly within a finite time (a *step*) by *all* its neighbors.

2. We assume that every node knows only its ID, its weight and the IDs and the weights of its neighbors (local knowledge).

As it will be clear from the description of the algorithm, when communicating with its neighbors each node transmits a message whose length is logarithmic in the size n of the network (up to a small multiplicative constant). Therefore, the listed assumptions can be easily implemented by using a "data link layer" (or, according to the wireless terminology, a MAC layer) protocol which takes into account the physical characteristics of the wireless channel such as the channel noise, the possibility of *collisions* on the transmission of the message, etc. (For a general introduction to computer networks, their organization in layers, and data link as well as MAC layer protocol, the reader is referred to [9].)

3 Algorithm Description

In this section we describe a distributed algorithm that, given any wireless network, determine a maximal weighted independent set among the nodes in the network. In describing the algorithm we exploit the similarity (and the terminology) between the problem of finding a maximal weighted independent set of nodes and the problem of determining the set of the clusterheads of a suitable clustering for wireless networks (we outlined this similarity in the Introduction). Thus, upon the algorithm termination, the network results partitioned into *clusters*, each of which has one *independent node* as its *clusterhead* and possibly some ordinary nodes.

The algorithm, henceforth referred to as the MWIS algorithm, is executed at each node so that a node v decides its own role (either a *clusterhead* or an *ordinary node*) depending solely on the decision of its neighbors with bigger weights.

Initially, only those nodes with the bigger weight in their neighborhood will broadcast a message to their neighbors stating that they will be a clusterhead (i.e., an independent node). On receiving one or more of this "clusterhead" messages, a node v will decide to join the cluster of the neighboring clusterhead with the biggest weight. If no node with bigger weight has sent such a message (thus indicating that it is going to join some other cluster as an ordinary node), then v will send a clusterhead message. We will show that all the nodes terminate the algorithms being either clusterheads (independent nodes) or ordinary nodes, and that the set of the clusterhead is indeed a MWIS.

Except for the initial routine, the algorithm is message driven: a specific procedure will be executed at a node depending on the reception of the corresponding message. We use two types of messages: CH(v), used by a node v to make its neighbors aware that it is going to be a clusterhead, and JOIN(v, u), with which a node v communicates to its neighbors that it will be part of the cluster whose clusterhead is node $u \in \Gamma(v)$. In the procedures, we use the following notation:

• v, the generic node executing the algorithm (from now on we will assume that v encodes not only the node's ID but also its weight w_v).

• *Ch* $(-)$ and *Join* $(-,-)$, boolean variables. Node v sets *Ch* (u), $u \in \{v\} \cup \Gamma(v)$, to **true** when either it sends a CH(v) message ($v = u$) or it receives a CH(u) message from u ($u \in \Gamma(v)$). The boolean variable *Join* (u, t), $u \in \Gamma(v)$, $t \in V$, is set to **true** by v when it receives a JOIN(u, t) message from u. They are initialized to **false**.

• By executing the command EXIT, a node v quits the execution of the clustering algorithm (this means that it will not listen to any more messages from the network).

Every node starts the execution of the algorithm at the same time, running the procedure *Init*. Only those nodes that have the biggest weight among all the nodes in their neighborhood will send a CH message. Given the nature of the weights (real numbers), there always exists at least a node v that transmits the message CH(v). All the other nodes just wait to receive a message.

```
PROCEDURE Init;
begin
  if for each u ∈ Γ(v) is w_v > w_u
     then begin
             send CH(v);
             Ch (v) := true;
             EXIT
          end
end;
```

Then, we have the following two message-triggered procedures:

• On receiving a CH message from a neighbor u, node v checks if it has received from *all* its neighbors z such that $w_z > w_u$, a JOIN(z, x) message, $x \in V$ (this is recorded in the corresponding boolean variables).[2] In this case, v will not receive a CH message from these zs, and u is the node with the biggest weight in v's neighborhood that has sent a CH message. Thus, v joins u and quits the algorithm execution (it already knows the cluster to which it belongs, i.e., its clusterhead). If there is still at least a node z, $w_z > w_u$, that has not sent a message yet, node v just records in the variable *Ch* (u) that u sent a CH message, and keeps waiting for a message from z.

```
On receiving CH(u);
begin
  Ch (u) := true;
  if for each z ∈ Γ(v) : w_z > w_u
    is Join (z, x) for some x ∈ V
      then begin
              send JOIN(v, u);
              EXIT
           end
end;
```

• On receiving a JOIN(u, t) message, node v checks the nature of the messages received so far from the nodes z such that $w_z > w_v$. If v has received a JOIN message from all its neighbors with bigger weight, this means that all those neighbors have decided to join a cluster as ordinary nodes. This implies that now v is the node with the biggest weight among the nodes (if any) that have still to decide what to do. In this case, v will be a clusterhead, it sends the corresponding message and quits the execution of the algorithm. Alternatively, if v has received at least a CH message from a node z with $w_z > w_v$, then it checks if all the neighbors u with weight bigger than w_z have sent a JOIN message. In

[2] We stipulate that the boolean conditions **for each ...** in the *Init*, On receiving CH(u) and On receiving JOIN(u, t) procedures evaluate to **true** when the sets on which they are based are the empty set.

this case, v can join the cluster of the biggest z that has sent a CH message (this is selected by the means of the operator $\max_{w_z}\{z : Ch(z)\}$), and quits the execution of the MWIS.

```
On receiving JOIN(u, t);
begin
  Join (u, t) := true;
  if for each z ∈ Γ(v) : w_z > w_v
   is Join (z, x) for some x ∈ V
      then begin
             send CH(v);
             Ch (v) := true;
             EXIT
           end
      else if for at least a z ∈ Γ(v)
        is Ch (z) and for each u ∈ Γ(v) : w_u > w_z
         is Join (u, x) for some x ∈ V
            then begin
                   send JOIN(v, max_{w_z}{z : Ch (z)});
                   EXIT
                 end
end;
```

EXAMPLE 1. Let us consider the simple wireless network of Figure 1 (a). At time step 1 (beginning of the algorithm) all the nodes execute the *Init* procedure of the MWIS. Nodes 4 and 7, being the nodes with the bigger weight in their neighborhood, send a CH message, declaring that they will be clusterheads (and they both quit the execution of the MWIS). By the end of the same time step, nodes 1 and 5 receive the message CH(4), nodes 2 and 6 receive the message CH(7) and nodes 3 and 8 neither receive nor send any message. At time step 2, by executing the procedure On receiving CH(4), nodes 1 and 5 sends messages JOIN(1,4) and JOIN(5,4), respectively, and quit the execution of the MWIS. (They are not prevented to do so by any node with weight bigger than the weight of node 4.) By the end of time step 1 nodes 2 and 6, by executing the procedure On receiving CH(7), know that 7 is a neighboring clusterhead. Since there is no node in 6's neighborhood with weight $> w_7$, node 6 sends the message JOIN(6,7) and quits the execution of the MWIS. Node 2 cannot do the same, since it has to wait for a message from node 1 (whose weight is $> w_7$). The JOIN(1,4) message is received at node 2 by the end of the second time step: since node 1 is going to join node 4's cluster, node 2 has to affiliate with node 7. Thus, by executing the procedure On receiving JOIN(1,4), node 2 sends the message JOIN(2,7). Let us consider now node 3. By the end of time step 2, it has received the message JOIN(1,4). At this time, though, node 3 cannot decide what role it is going to assume, since there is still another node (node 2) with a weight bigger than its own weight w_3 that has to decide. At the end of the third time step, node 3 finally receives the message JOIN(2,7) and, by executing the procedure On receiving JOIN(2,7), it realizes that all the nodes with weight bigger than its own weight have decided to join another cluster. Thus, it sends a CH(3) message, declaring that it will be a clusterhead (and terminates the execution of the MWIS). By the end of step 4 this message is received by node 8 (by executing the procedure On receiving CH(3)) that, since $w_3 > w_8$ and no other node with weight $> w_3$ is in its neighborhood, joins node 3's cluster by sending the message JOIN(8,3) and quits the execution of the MWIS. Thus, by the end of the fifth step the MWIS is terminated at all nodes (the clustering induced by executing the MWIS is depicted in Figure 1 (b)). ◇

Once the algorithm is terminated at all nodes, the nodes v whose local variable $Ch(v)$ is set to **true** belong to a maximal weighted independent set.

4 Algorithm Analysis

In this section we prove the correctness and the time and message complexity of the MWIS algorithm. We start by proving that the nodes that sends a CH message form an independent set.

Proposition 1 *A node v of a wireless network sends a* CH *message if and only if all its neighbors z with $w_z > w_v$ have already joined another clusterhead. A node v sends a* JOIN(v,u) *message as soon as at least a neighbor u with $w_u > w_v$ has sent a* CH(u) *message and all the neighbors z such that $w_z > w_u$ have joined a cluster. In this case u is guaranteed to have the biggest weight among all nodes that have sent a* CH *message.*

Proof: Let us consider the set $Z = \{z : z \in \Gamma(v), w_z > w_v\}$. If $Z = \emptyset$ then the thesis follows because of the *Init* procedure (node v sends a CH(v) message). When $Z \neq \emptyset$, then the only case in which v can send a CH(v) message is by executing the **then** branch of the outermost **if** in the JOIN procedure. This can happen if and only if all the nodes $z \in \Gamma(v)$, $w_z > w_v$, have already joined a clusterhead $x \in V$.

A node v can send a JOIN message if and only if it executes either the CH procedure or the **else** branch of the outermost **if** of the JOIN procedure. In both cases it has to have received a CH message from at least a node u such that $w_u > w_v$, and JOIN messages from all the nodes with weights $> w_u$. If the JOIN message has been sent while executing the CH(u) procedure, u is guaranteed to have the biggest weight among all the nodes $z \in \Gamma(v), w_z > w_v$, because all nodes with weight $> w_u$ have already sent a JOIN message (otherwise, v could not execute the **then** branch of the **if** command). If the JOIN message has been sent while executing the **else** branch of the outermost **if** in the JOIN procedure, the thesis is guaranteed by the max operator. •

Corollary 1 *(Independence) No two clusterheads can be neighbors.*

Proof: Let, by the sake of contradiction, v and u be neighboring clusterheads. Suppose, without loss of generality, that $w_u > w_v$. Then, v cannot be a clusterhead, because there exists at least one of its neighbor, namely u, with $w_u > w_v$, that has not joined another clusterhead (u has already sent a CH(u) message, and it has quit the algorithm execution). •

To prove that the set of all clusterheads form a maximal independent set, and also to establish the message and time complexity of the MWIS algorithm we introduce some definitions.

Let us consider the common (centralized) greedy algorithm for finding a maximal weighted independent set in an undirected graph $G = (V, E)$ and let us extend it to generate a partition of the graph G into k ($\leq n$) clusters $C_1, \ldots, C_k$ so that each of the k independent nodes is the clusterhead of a cluster.

```
CLUSTERS(G = (V, E));
begin
  i := 0;
  while V ≠ ∅ do
   begin
     i := i + 1;
     v := min{u ∈ V : w_u = max{w_z : z ∈ V}};
     C_i := {v} ∪ Γ(v);
     V := V \ C_i
   end
end;
```

At each iteration i of the main loop the node v with the biggest weight is selected (here we break possible ties by selecting the node with the lowest ID). Then, the cluster C_i is formed by v as the clusterhead and all its neighbors (if any) as ordinary nodes. It is clear that the set of all the clusterheads forms a maximal weighted independent set (see also, e.g., [7]).

Given the clustering $C_1, \ldots, C_k$ we define the following function $\tau : V \to \{1, \ldots, 2k\}$ that assigns to each of the wireless nodes v an integer number as follows: $\tau(v) = 2i - 1$ if v is C_i's clusterhead; $\tau(v) = 2i$ if v is C_i's ordinary node. (The function τ is well defined since the procedure CLUSTERS do not allow overlapping clusters.) It is immediate from the definition of τ that for each cluster C_i with clusterhead v is $\tau(v) = \tau(u) - 1$, where $u \in C_i \setminus \{v\}$ and $i = 1, \ldots, k$. Another useful property is stated by the following:

Proposition 2 *All the neighbors u (if any) of a clusterhead v such that $w_u > w_v$ are ordinary nodes such that $\tau(u) < \tau(v)$.*

Proof: Let C_i be the cluster of v, $i = 1, \ldots, k$. According to the procedure CLUSTERS, at the ith step of the **while** loop, node v has the bigger weight among all the nodes that have not been assigned to a cluster yet. This implies that all the neighbors u with weight bigger than v's weight, if any, are $a)$ ordinary nodes, and $b)$ already assigned to a cluster C_j with $j < i$. Indeed, $a)$ if a clusterhead u with $w_u > w_v$ is in v's neighborhood, then v could not be a clusterhead, since u would have been picked up before v and v would belong to u's cluster. Moreover, $b)$ if an ordinary node u with $w_u > w_v$ has not been assigned to a cluster yet, then it would be chosen as the clusterhead at the ith iteration of the **while** loop, again preventing v to be C_i's clusterhead. By the definition of τ this implies $\tau(u) < \tau(v)$. •

The following result is fundamental in proving the time and message complexities of the MWIS algorithm as well as its correctness.

Proposition 3 *Each node v of the network sends* exactly *one message within $\tau(v)$ steps.*

Proof: *Each node v sends at least a message within $\tau(v)$ steps.* We proceed by induction on the definition of $\tau(v) = \ell \leq 2k \leq n$, where, as usual, k is the number of clusters returned by the procedure CLUSTERS. When $\tau(v) = 1$ (i.e., when v is the clusterhead of C_1) the claim is obvious: if v is C_1's clusterhead, then it sends the message CH by executing the procedure *Init* (first step). Let us now assume that all the nodes u such that $\tau(u) = \ell$ $(< 2k)$ have sent either a CH or a JOIN message within ℓ steps. Let v be a node such that $\tau(v) = \ell + 1$. We distinguish two cases:

1. *v is a clusterhead.* In this case, by Proposition 2, we know that all the nodes u in v's neighborhood such that $w_u > w_v$ are ordinary nodes such that $\tau_u < \tau_v$. This implies that, by induction hypothesis, they have sent a JOIN message by ℓ steps. Thus, by executing the procedure On receiving JOIN(u, t), where u is the last ordinary node that sends that message among all v's neighbors with $w_u > w_v$, node v sends the message CH(v) by $\ell + 1$ steps.

2. *v is an ordinary node.* In this case, since v's "heaviest" neighboring clusterhead u is such that $\tau(u) = \tau(v) - 1$ and since a node v that receives a CH(u) message sends a JOIN(v, u) message in the following step (Proposition 1), by induction hypothesis u sends a CH(u) message by ℓ steps, allowing v to send the JOIN(v, u) message within $\ell + 1$ steps.

Each node v sends at most one message. Immediate by inspecting the procedures' code: after a **send** command, a node always quits the algorithm execution. •

With the previous proposition we have stated that, for each node v of a wireless network G, $\tau(v)$ indicates the *time* (in number of steps) that v has to *wait* before deciding its role, i.e., before declaring if it is going to be a clusterhead (i.e., a node in the maximal weighted independent set) or an ordinary node. The following results are immediate corollaries of Proposition 3.

Corollary 2 *The MWIS terminates within $2k$ steps, being k the cardinality of the maximal weighted independent set found.*

Proof: As soon as it sends a message, a node always quits the execution of the MWIS. •

On the wireless network of Figure 1 (a) the MWIS terminates (correctly) in $5 < 2k = 6$ time steps (see Example 1).

The following corollary states the message complexity of the MWIS.

Corollary 3 *The message complexity of the MWIS is n.*

Corollary 4 *Each node belongs exactly to a cluster.*

Proof: Each node v that sends a CH(v) message belongs to the cluster of which it is the clusterhead. On the other hand, if a node u sends a JOIN(u, t) message, $t \in \Gamma(u)$, then it belongs to the cluster whose clusterhead is t. •

The previous corollary immediately implies that a node is either a clusterhead or an ordinary node, that an ordinary node joins only one clusterhead (no overlapping clusters) and that an ordinary node is only one hop apart from the clusterhead it joins (namely, the maximal weighted independent set is also a *dominating* set). As a consequence we have:

Corollary 5 *(Maximality) The set of the clusterheads is maximal.*

The previous results are summed up in the following:

Theorem 1 *(Correctness) All the nodes of the wireless network exit the execution of MWIS having been assigned either the role of clusterhead or the role of ordinary node. The set of all the clusterheads forms a maximal weighted independent set.*

We conclude this section with a note on the time complexity of our algorithm. It is clear that the bound presented in this paper depends on the (possibly changing) topology of the wireless network as opposed to an invariant like the number n of its nodes. As immediate consequence, this implies an improvement on the $O(n)$ upper bound presented in [6] in all those cases in which $k \ll n$. Of course, in the worst case, the time complexity of the MWIS remains $O(n)$.

We show now that, "on average," the parameter k—and hence the MWIS time complexity—is (asymptotically) logarithmic in the number n of the nodes of the wireless network. On average, here, is intended with respect to a mobile scenario in the following sense. When the nodes of a wireless network are mobile, the independence of the clusterheads can be lost due to the fact that two or more clusterheads can become neighbors. One way to cope with this problem is to invoke a re-clustering of the network either periodically or every time a new maximal weighted independent set is needed. In this setting, it is interesting to measure the average time complexity of the MWIS as resulting from its application to graphs that are snapshots of the network topology at the time a re-clustering is invoked.

Provided that the nodes do not move during the clustering process, we can "abstract" the network graph $G = (V, E)$, $|V| = n$ and $|E| = m$, as a *random graph*. A *random graph*, as beautifully defined in [8], is one of the element of the probability space that consists of all (labeled) graphs G with n nodes and m edges. The key feature we use from this theory is its power of providing rigorous analysis tools for various properties on large statistical manifolds of graphs. In this way we can avoid the difficulties caused by worst cases, since they usually occur with very small probability. Using the theory of random graphs, in many cases it is possible to rigorously prove that the probability of the worst case to occur tends to zero as the graph size grows. Let us explain this situation with a short example based on a classical NP-complete problem. Let G be a graph with n vertices and m edges. Assume a number h is given and we want to decide whether G contains an independent set of size $\geq h$. As mentioned earlier, this is well known to be an NP-complete problem [1]. We have also already mentioned that the corresponding optimization version (the *maximum independent set* problem) cannot even be approximated by a polynomial time algorithm (assuming $P \neq NP$) within an error factor of $n^{1-\epsilon}$ for any fixed $\epsilon > 0$. Thus, for large values of n no fast algorithm can guarantee even an acceptable approximation for the worst case. On the other hand, in a random graph setting with the same parameters, the maximum size $\alpha(G)$ of an independent set in G can be computed, as it is shown to be $O(log_b n)$, where b is defined as the probability to have an edge between any two nodes in the graph and that depends on n and m [8, Chapter XI]. For a finite graph this holds *almost surely*. This means that the fraction of all graphs that deviate from the formula tends to 0 as n grows. Thus, if we have a graph that is chosen at random from all graphs of n nodes with m edges, we can determine the desired quantity with vanishing error probability, despite its *worst-case* algorithmic intractability.

Since the size k of a maximal weighted independent set is $\leq \alpha(G)$, we conclude this section by stating the following:

Theorem 2 *The average time complexity of finding a maximal weight independent set in a mobile wireless network with n nodes and m edges is bounded by $O(log_b n)$, where $b = b(n, m)$ is the probability of having an edge between any two pair of nodes in the network.*

5 Conclusions

This paper presented MWIS, a distributed algorithm for the efficient determination of a maximum weighted independent set in the topology graph G of a wireless network. It is shown that the algorithm can be effectively used as a base for clustering algorithms for wireless networks. This is a practically important tasks for all those network algorithms/applications that assume an underlying hierarchical organization of the network nodes. The proposed algorithm requires only knowledge of the local topology at each node, it is easy to implement and its time complexity is proven to be bounded by the network stability number $\alpha(G)$ (a network parameter that depends on the possibly changing topology of the wireless network). Finally, by abstracting the topology graph of a wireless network as a random graph, we have shown that the time complexity of the MWIS is, on average, logarithmic in n, the number of the nodes of the network.

References

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

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[3] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, November 1986.

[4] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, December 1986.

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

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

[7] S. Basagni, I. Chlamtac, and A. Faragó. A generalized clustering algorithm for peer-to-peer networks. In *Workshop on Algorithmic Aspects of Communication, satellite workshop of ICALP'97, invited paper*, Bologna, Italy, July 11–12 1997.

[8] B. Bollobás. *Random Graphs*. Academic Press Inc., London, UK, 1985.

[9] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, Englewood Cliffs, N.J., third edition, 1996.