Minimum Total Node Interference in Wireless Sensor Networks

Nhat X. Lam, Trac N. Nguyen, and D.T. Huynh

Department of Computer Science
University of Texas at Dallas, Richardson Texas 75083-0688
{nxl081000,nguyentn,huynh}@utdallas.edu

Abstract. The approach of using topology control to reduce interference in wireless sensor networks has attracted attention of many researchers. There are several definitions of interference in the literature. In a wireless sensor network, the interference at a node may be caused by an edge that is transmitting data [16], or it occurs because the node itself is within the transmission range of another [2], [4], [7]. The interference load of a node is either the number of nodes in the broadcasting disk defined by this node or the number of nodes whose disks cover it [2], [4], [7]. In this paper we show that the problem of assigning power level to a set of nodes in the plane to yield a connected geometric graph whose total node interference is bounded is NP-complete under both definitions. We also introduce some heuristics as well as a simplified version of an $O(logn)$ approximation algorithm in [10] and study their performance through simulation.

Keywords: wireless sensor networks, interference, NP-completeness, geometric graphs.

1 Introduction

Wireless sensor networks (WSNs) have been extensively used in military and civilian applications in the last two decades. A primary issue concerning WSNs is interference, which occurs when communication between a pair of nodes is affected by another node that is transmitting data. One well known approach to minimize interference is to use topology control by reducing the power usage of certain nodes thereby establishing a simple connected network with low interference.

The approach of using topology control to reduce interference was first discussed in a number of papers including [4], [12]. The authors in [4] defined the notion of interference load of an edge in a network, and showed an interesting result that certain sparse networks may not have low interference. Following the work in [4], the authors in [12] introduced a notion of (receiver-based) node interference that is caused by surrounding nodes whose transmission range includes the given node. They analyzed the special case of the exponential 1-dimensional node chain which is also called the *highway* model. They showed that this sparse

J. Zheng, D. Simplot-Ryl, V.C.M. Leung (Eds.): ADHOCNETS 2010, LNICST 49, pp. 507–523, 2010.

network has $\Omega(\sqrt{n})$ node interference, where n is the number of nodes in the network. The authors described an algorithm that provides an $O(\sqrt[4]{\delta})$ approximation of the optimal connectivity preserving topology in the highway model where δ is the maximum node degree. Similarly, the papers [16], [9], and [8] focused on the notion of interference that is based on edges of the network. [8] gave a distributed algorithm called Average Path Interference that tries to preserve the spanner property of the original graph while reducing the interference in the network. [9] showed that the relative neighborhood graph and local spanning tree algorithms have a constant bounded average interference ratio.

The NP completeness (NPC) of both types of node interference were discussed in [16], [2], [3], [11] and [14]. [16] extended the work of [4] with an NP-completeness proof for the problem of minimizing edge interference for general graphs along with a couple of heuristics, and [3] provided an NP-completeness proof for finding a spanning tree with minimum node interference for grid graphs. For the receiver-based interference model, in [2] the authors showed among other results that the problem of minimizing the maximum node interference is hard to approximate. On the other hand, [7] showed that for a set of n points in the plane a network with $O(\sqrt{\Delta})$ interference can be constructed using computational geometric tools (Δ is the maximum interference in the uniform-radius network). They left open the question whether this problem is NP-hard including the 1-dimensional case. [11] was able to provide an answer to some of the questions raised in [7] by showing that minimizing receiver-based node interference is NPC for the 2-dimensional case. However, the sender-based model this problem turned out to be solvable in polynomial time as shown in [2].

In this paper, we are concerned with total node interference. (Notice that this problem is equivalent to the average node interference problem.) We study the problem of assigning power to nodes in the plane to form a connected graph in which the total interference load is bounded. Specifically, we prove that the problem of assigning power to a set of nodes in the plane to yield a connected geometric graph whose total node interference is bounded is NP-complete. Our result is significant in view of the result in [10] where NP completeness was proved for the (general) ad hoc metric model only. Note that in our work as well as most of the works in WSNs, two nodes are connected by an edge if they are within the transmission range of one another. This definition is not strictly followed in a number of papers including [1] and [13]. We also simplify an $O(logn)$ approximation algorithm reported in [10] for this problem. The performance of this algorithm is compared against a number of heuristics through simulation. The result shows that this algorithm outperforms others.

The rest of this paper is organized as follows. Section II provides the definitions and explanations used in this paper. Section III is devoted to the NP completeness result, and Section IV discusses the $O(logn)$ approximation algorithm as well as a number of heuristics. Section V contains some concluding remarks.

2 Preliminaries

2.1 Network Model

Consider a set V of transceivers (nodes) in the plane. Each node u is assigned a power level denoted by $p(u)$. The signal transmitted by node u can only be received by a node v if the distance between u and v, denoted by $d(u, v)$, is $\leq p(u)$. We only consider the bidirectional case in which a communication edge (u, v) exists between two nodes u and v, if both power levels $p(u) \geq d(u, v)$ and $p(v) \geq d(u, v)$. Thus, the set V of nodes in the plane together with the power levels assigned to the nodes define a *geometric* (also known as intersection) graph $G = (V, E)$. A geometric graph is said to be *planar* if no edge crosses another.

2.2 Interference Model

There are several definitions of the notion of interference in the literature. In this paper we consider two definitions of interference described in [2], [4] and [8]. Assuming that there is no obstacle blocking the broadcasting range, the two definitions of the interference load are as follows.

Let $D(u, r_u)$ be the broadcasting disk of node u with radius r_u. For interference load, the definition from [2] is formally defined for the receiver and sender-based model as follow:

For receiver based interference load:

$$RE(x) := |\{w \in V | w \neq x \text{ } and \text{ } x \in D(w, r_w)\}|$$

For sender based interference load:

$$SE(x) := |\{w \in V | w \neq x \text{ } and \text{ } w \in D(x, r_x)\}|$$

For the total interference load of a graph $G = (V, E)$

$$TNI(G(V, E)) := \sum_{x \in V} RE(x) \quad = \quad \sum_{x \in V} SE(x)$$

In the interference load model based on Euclidean distance, the radius r_v of node v is defined to be $r_v := max_{(v,w) \in E} \{d(v, w)\}$, whereas in the interference load model based on power usage, the radius r_v of node v is defined to be $r_v := p(v)$.

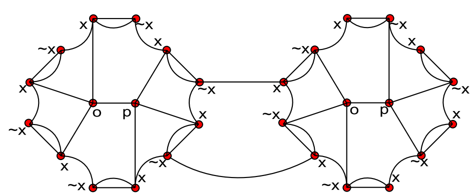

Fig. 1. Gadgets used in representing a variable x in a PL1-EX-3SAT instance

3 The NP-Completeness of Minimum Total Node Interference in Geometric Graphs

In this section, we prove the NP-completeness of the problem of assigning power levels to nodes in the plane to produce a connected geometric graph G with bounded total node interference load $TNI(G)$ based on Euclidean distance. As a corollary, the same problem for the power level based model is also NP-complete.

MINIMUM TOTAL NODE INTERFERENCE IN CONNECTED GEOMETRIC GRAPHS

Instance: *Given a set of* N *nodes* $V = \{v_1, v_2, ..., v_N\}$ *in the plane, a set of* M *power levels* $P = \{p_1, p_2, .., p_M\}$ *that a node can transmit, and a positive number* R*.*

Question: *Is there a power assignment to all nodes which induces a connected geometric graph* $G(V, E)$ *such that* $TNI(G(V, E)) \leq R$*?*

In the following we show that Minimum Total Node Interference in Geometric Graphs based on Euclidean distance is NP-complete.

Theorem 1: Minimum Total Node Interference is NP-complete for connected geometric graphs.

Proof. Minimum Total Node Interference for Connected Geometric Graphs (MTNICG) is obviously in NP. Given a set V of nodes in the plane, a set P of power levels and a positive integer R, we can nondeterministically assign power levels to the nodes, and verify in polynomial time that (1) the power assignment yields a connected geometric graph $G(V, E)$, (2) the total interference of all nodes is $\leq R$.

To prove the NP-hardness of MTNICG, we construct a polynomial time reduction from the planar 1-Exact-3SAT problem (PL1-EX-3SAT) which was proven NP-complete in [6]. Consider an instance ϕ of PL1-EX-3SAT where each clause has exact 3 literals, and the planar instance graph G of ϕ, where $G = (X \cup C, E \cup E')$ with edge sets $E = \{\{x, c\} | x \in C \vee \neg x \in C\}$ and $E' = \{\{x_i, x_{i+1}\} | 1 \leq i \leq n-1\}$. ϕ satisfiable iff it has a Boolean assignment such that *exactly* 1 literal per clause has the value *TRUE*.

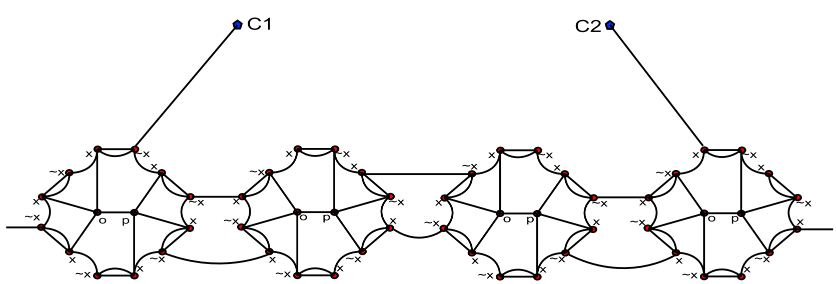

Fig. 2. A series of gadgets representing a variable x of degree 2 in PL1-EX-3SAT

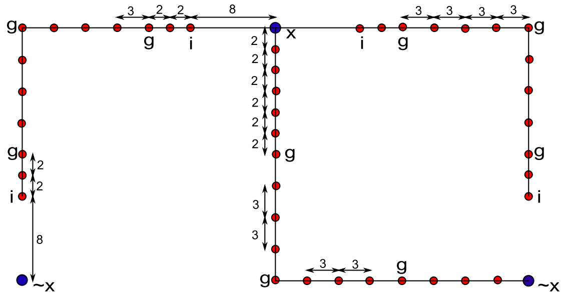

Fig. 3. Nodes added on the line segments replacing two *curved* edges (left and right) and a *straight* edge (down) at variable x with degree 3

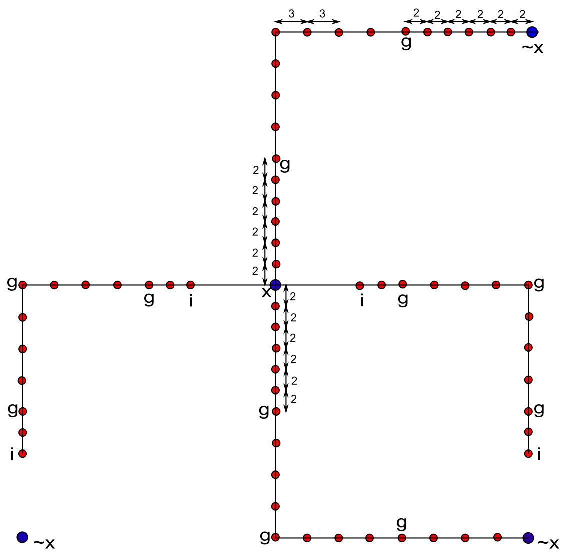

Fig. 4. Nodes added on the line segments replacing two curved edges (left and right), a *straight* edge (up) and a *straight bridge* (down) at variable x with degree 4

To construct an instance $< V, P, R >$ of MTNICG, we first create a pair of gadgets for each variable x of ϕ. As shown in Fig. 2, each gadget for a variable x contains 14 nodes: 12 of these nodes form 6 pairs of x and $\neg x$, and the remaining two nodes are center nodes o and p. Every pair of variable and its negation node (x and $\neg x$) is connected by an edge called *curved* edge. There are 12 *curved* edges for each gadget. 6 pairs of a variable node and its negation (x,$\neg x$) are connected by *straight* edges such that every node in a gadget is adjacent with exactly one *straight* edge. In one gadget, one set of nodes (all x nodes or all $\neg x$ nodes) are connected to the center nodes (o and p) so that every center node is connected with exactly three x (or $\neg x$) nodes. The nodes in the second gadget are connected similarly. However, if nodes x are used to connect to center nodes

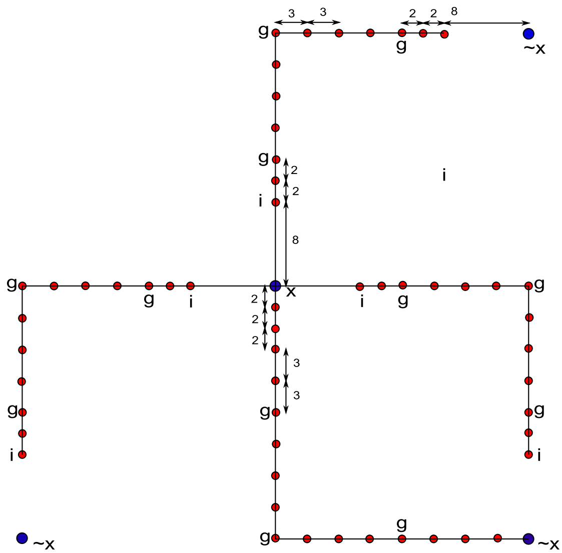

Fig. 5. Nodes added on the line segments replacing two curved edges (left and right), a *straight* edge (down) and a *curved bridge* (up) at variable node x with degree 4

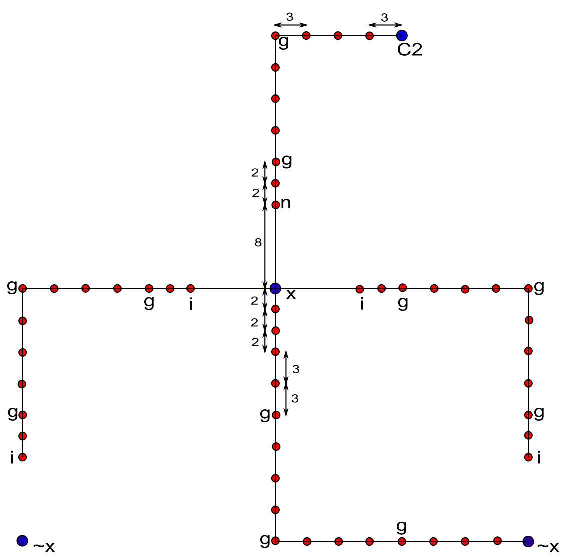

Fig. 6. Nodes added on the line segments replacing two curved edges (left and right), a *straight* edge (down) and the edge (up) connecting to a clause node $C2$ at variable node x with degree 4

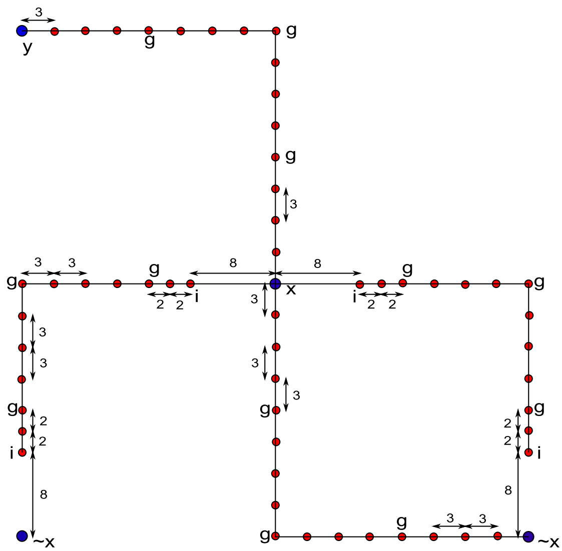

Fig. 7. Nodes added on the line segments replacing two curved edges (left and right), a *straight* edge (down) and a *variable link* (x,y) at variable node x with degree 4

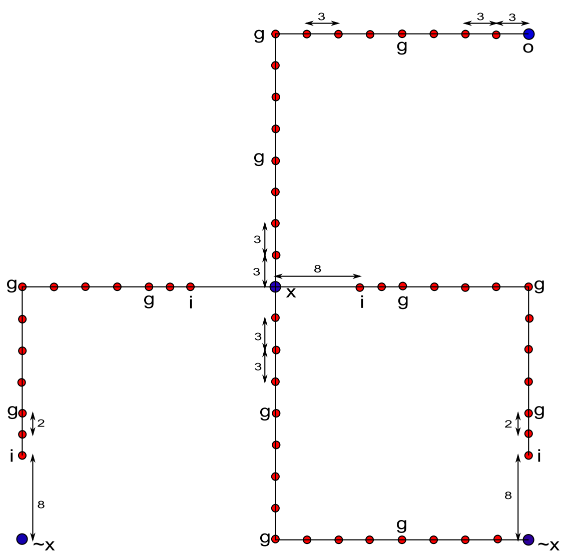

Fig. 8. Nodes added on the line segments replacing two curved edges (left and right), a *straight* edge $(x,\neg x)$ and a *straight* edge (x,o) to *center* node o at variable node x with degree 4

in the first gadget, then nodes $\neg x$ are used in the second gadget, or vice versa. Notice that each gadget has 6 nodes of degree 3 to connect to other gadgets and clause nodes. Inside each gadget, the two center nodes (o and p) are connected by an edge. To connect every two gadgets, two edges are used: a *straight bridge* and a *curved bridge*. These *bridges* connect degree 3 nodes of each gadget. Thus, each gadget uses up to 4 degree 3 nodes for *bridges* and has at least 2 degree 3 nodes to connect with clause nodes (see Fig. 2.)

If the degree of node x is d_x, then the number of x's pair of gadgets in this chain is also d_x (for a total of $2d_x$ gadgets). Thus, the new graph (also denoted G for convenience) obtained from the original graph has a maximum degree of 4 while the planarity of the graph is still preserved.

Next, we use Valiant's result [15] to embed the graph G (with maximum degree 4) into the Euclidean plane:

> *A planar graph with maximum degree 4 can be embedded in the plane using $O(|V|)$ area in such a way that its vertices are at integer coordinates and its edges are drawn so that they are made up of line segments of form $x = i$ or $y = j$, for integers i and j.*

Moreover, this embedding process can easily be designed to satisfy the additional requirement that each line segment drawn to connect two original vertices of the graph G must be of length at least 2 and every two parallel segments must be at least 2 units apart.

Let's call the units of the embedded graph G *original* units. Each *original* unit is divided further into 12 smaller "pieces" of equal length. Let the original unit be δ. We define the radii r_i, $1 \leq i \leq 8$ as follows: $r_i := (i/12) * \delta$. For the sake of convenience let us call the variable, clause and center nodes of G embedded in the plane the *variable*, *clause* and *center* nodes, respectively. We can further modify each line segment by placing additional new nodes to create the instance $< V, P, R >$ of MTNICG as follows:

1. On every line segments, add a node on every grid point. These nodes are called *grid* nodes.
2. On the line segments representing a connection of a *variable* node and a *center* node, or between two *center* nodes, add 3 nodes on each unit to divide it into 4 pieces of length r_3 each.
3. On the line segments representing a *curved* edge or a *curved bridge*, add one node at distance r_8 from the *variable* node on each of the two adjacent units. These interfacing nodes are called i nodes. (See nodes i's in Fig. 3, 4, 5, 6, 7,). Then add another node at distance r_2 between the previously added i node and the *grid* node. On other unit segments in between, add 3 nodes to divide each unit into 4 equal pieces of length r_3 each.
4. On the line segments representing a *straight bridge*, add 3 nodes to each unit to divide it into 4 equal pieces of length r_3 each except the two end units adjacent with *variable* nodes. For these two units, add 5 nodes to divide it into 6 equal pieces of length r_2 each (see Fig. 4.)

5. On the line segments representing a *straight* edge, add three nodes on each unit to divide it into 4 equal pieces of length r_3 each, except the unit that is adjacent with a *variable* node. For this unit, there are four cases:
   - If this unit is adjacent with a degree 3 *variable* node, add 5 nodes on the unit to divide it into 6 pieces of length r_2 each (see Fig. 3).
   - If this unit is adjacent with a degree 4 *variable* node having an adjacent edge which is a *curved bridge* or a link to a *clause* node, add 2 pairs of nodes. The first pair is added at the distance of r_3 from the *variable* node and from each other. The second pair is added at the distance of r_2 from the *grid* node and from each other (see down units in Fig. 5 and 6.)
   - If this unit is adjacent with a degree 4 *variable* node having a *straight bridge* edge, add 5 nodes on the unit to divide it into 6 equal pieces of length r_2 each. (See "up" unit in Fig. 4.)
   - If this unit is adjacent with a degree 4 *variable* node different than those in the previous cases, add 3 nodes to divide this unit into 4 equal pieces of length r_3 each.
6. On the line segments representing a link from a *variable* node to a *clause* node, add one node at distance r_8 to the *variable* node denoted n on the unit adjacent with the *variable* node. (See "up" unit in Fig. 6.) Add a second node at distance r_2 to the previously added node (n) and the *grid* node on the same unit. On other unit segments leading to the *clause* node, add 3 nodes on each segment to divide each unit into 4 equal pieces of length r_3 each.
7. On all other line segments, add 3 nodes to each unit to divide it into 4 equal pieces of length r_3 each.

(Figures 3, 4, 5, 6, and 7 show *variable* nodes x have power level r_8 to connect with nodes i's while its negation nodes $\neg x$ have power level r_3 and are not connected with its neighboring i nodes.)

Before defining the interference bound R, we note there are 5 categories of nodes in $< V', P, R >$: the set of variable nodes also denoted X, where each x has degree d_x, the set M of center nodes o's and p's, the set I of i nodes, the set C of clause nodes, and the set T of remaining nodes. For each set of nodes we define a corresponding interference quantity as follows. For a set U of nodes let $SE(U) := \sum_{u \in U} SE(u)$.

For a variable $x \in X$ of degree d_x, there are d_x pairs of gadgets, and each gadget has 12 *variable* nodes. Each of exactly $11d_x$ *variable* nodes is allowed to have the interference load of 6 when assigned the power level r_8 corresponding to *TRUE* (see Fig. 3, 5, 6); and each of the other d_x variable nodes is allowed to have the interference load of 10 (Fig. 4) to connect through a *straight bridge.* Each of the other 8 and 4 variable nodes (when assigned *FALSE* corresponding to power level r_3) is allowed to have the interference load of 2 (Fig. 7, 8) and 1 (Fig. 3, 5, 6), respectively:

$$SE(X) := \sum_{x \in X} (6 * 11d_x + 10 * d_x + 2 * 8d_x + 4 * d_x) = \sum_{x \in X} 96 * d_x$$

For the *center* nodes in M we define:

$$SE(M) := \sum_{x \in X} (4 * 4d_x) = \sum_{x \in X} 16 * d_x$$

This allows each center node to have an interference load of 4 to connect to its 4 neighbors. (Note that there are $2d_x$ nodes os and $2d_x$ nodes ps for a variable node x of degree d_x.) The interference quantity for nodes is is so defined to accommodate the interference load of the variable nodes x: From the $24d_x$ *curved* edges, $24d_x$ (half) of nodes is are allowed to have the interference load of 4 each, while the other $24d_x$ (half) of nodes i's to have the interference load of 1. From the $2d_x - 1$ *curved* bridges, $2d_x - 1$ (half) of nodes is are allowed to have interference load of 4 while the other half have interference load of 1.

$$SE(I) := \sum_{x \in X} ((4 * 24 + 1 * 24)d_x + (2 * d_x - 1) * 5) = \sum_{v \in V} (130 * d_v - 5)$$

For the *clause* nodes $c \in C$, the interference quantity is defined as follows. Each clause node has degree 3 and is allowed to cover its 3 neighbors. Each clause node connects to its literal via the 3 nodes ns: exactly one of these 3 nodes is allowed to cover 4 neighbors, the other 2 are allowed to cover only 1 neighbor each. Thus,

$$SE(C) := \sum_{c \in C} (4 * c + 3 * c + 2 * c) = \sum_{c \in C} 9 * c$$

For the set T of remaining nodes which is the set of all nodes minus nodes i's, *variable* nodes, center nodes o's and p's, nodes n's as well as nodes c's, we define the interference quantity $SE(T)$ as $SE(T) := 2|T|$.

Let $P := \{0, r_1, r_2, r_3, ..., r_8\}$ and $R := SE(X) + SE(O) + SE(M) + SE(I) + SE(C) + SE(T)$. The correctness of the above polynomial-time reduction follows from the following claim:

Claim. The instance ϕ of PL1-EX-3SAT is satisfiable if and only if the instance $< V', P, R >$ of MTNICG has a power assignment that yields a connected geometric graph $G'(V', E')$ such that the total interference load $TNI(G'(V', E')) \leq R$.

The proof of the Claim can be found in the Appendix. From the Claim, Theorem 1 follows.

4 Heuristics and Their Performance

4.1 Heuristics

In this section we present three simple heuristics and an approximation algorithm which is a simplified version of a greedy algorithm reported in [10].

Node interference approach: Our first approach is built on the sender-based node interference problem, where the interference load of a node v is the number of nodes within the broadcasting disk of v and the problem is to minimize the maximum node interference. This problem has been shown to be solved in polynomial time in [2] through a simple greedy heuristic called OPT-MINMAXSIP. Based on this result, we propose the Node Power Level Search heuristic (NPLS) which first computes the minimum sender-based node interference, and then reduce the power level for each node while preserving connectivity. Although this step does not necessarily decrease the maximum node interference of the graph, it may improve the interference load of other nodes, and hence the total node interference load. The pseudo code of this algorithm is depicted in Figure 9.

```
Node-Power-Level-Search(V)
 Input: a set V of nodes in the plane
 Output: A power assignment P that
   yields a connected geometric graph
   whose total node interference is minimized
   Using the greedy technique, find a power
        assignment P that yields a minimum
        node interference graph
   // Reduce power levels of nodes
   // while maintaining connectivity
   for each node v ∈ V
        Using binary search find the least power level
        for P[v] while P yields a connected graph
   return the power assignment P
```

Fig. 9. Node-Power-Level-Search

Minimum spanning tree approach: The basic idea of this approach is to find a minimum spanning tree for a given complete graph. We use this approach in two different algorithms, the distance-based minimum spanning tree (DMST) algorithm and the interference-based minimum spanning tree (IMST) algorithm. The main difference between these two algorithms is the definition of the edge weight. The former uses the Euclidean distance as edge weight while the latter uses the DEI edge interference load. In implementing these two heuristics, we use Kruskals algorithm [5] to find the minimum spanning tree. This algorithm uses two well-known subroutines find-set and union. find-set(u) is to find the root of tree-based connected component containing node u whereas union(u, v) is to connect two connected components of u and v. The details of these two subroutines can be found in [5]. Figure 10 contain the pseudo code of the IMST algorithm. The pseudo code of the DMST algorithm is similar.

Greedy approach: In [10], the authors suggest a greedy algorithm for the ad hoc interference model with $O(\log n)$ approximation ratio. This algorithm selects the most cost-efficient star at each iteration to join some current connected

```
INFERENCE-BASED-MINIMUM-SPANNING-TREE(V)
  Input: a set V of nodes in the plane
  Output: A power assignment P that
    yields a connected geometric graph
    whose total node interference is minimized
  E = set of edges between all nodes in V
  A = ∅
  w(e) = interference of e for all e ∈ E
  Sort edges of E into non-decreasing order by w
  for each edge (u, v) ∈ E
      if FIND-SET(u) ≠ FIND-SET(v)
          add edge (u, v) to A
          UNION(u, v)
          B = edges created by adding (u, v)
          A = A ⋃ B
          E = E \ B
          Recalculate w and sort edges in E
  Compute the power assignment P from A
  return P
```

Fig. 10. INTERFERENCE-BASED-MINIMUM-SPANNING-TREE

components by increasing the transmission range of each node in this star until a connected graph results. This algorithm does not seem to work as claimed in [10] since there is a discrepancy between the algorithm and its proof. In the proof, the authors consider the size of each star as the number of connected components to be connected. However, the suggested method to find the best star does not guarantee that this requirement is satisfied.

In the following, we provide a simpler version of the greedy algorithm in [10]. This simplified algorithm is in polynomial time and also has an $O(\log n)$ approximation ratio. The idea of our modification is based on Kruskal's minimum spanning tree approach. The algorithm proceeds in a greedy fashion and selects in each step a proper pair of nodes to connect two connected components by increasing the transmission ranges of these nodes. The algorithm will connect pairs of nodes until there is only one connected component in the resulting graph.

The major issue is the greedy property used to determine the proper pair of nodes at each step. First, considering an edge $e(u, v)$ we defined the induced power assignment of this edge as follows.

$$\widehat{P}^e(a) = \begin{cases} f_u(v) \text{ if } a = u \\ f_v(u) \text{ if } a = v \\ 0 \qquad \text{otherwise} \end{cases}$$

where $f_u(v)$ denotes the smallest power level such that v is covered by $D(u, f_u(v))$. Let t_k^a denote the number of nodes covered by $D(a, k)$. Given a graph G_i

```
GREEDY(V)
 Input: a set V of nodes in the plane
 Output: A power assignment P that
   yields a connected geometric graph
   whose total node interference is minimized
 for each a ∈ V P_0(a) = 0
 i = 0
 while G induced from P_i is not connected
       E = set of edges connecting two different
       connected components
       e(u,v) = argmin_{e∈E}(cost(e)/|C_{P_i}(e)|)
       for all a ∈ V
             P_{i+1}(a) = max(P_i(a), \hat{P}^{e(u,v)}(a))
       i = i + 1
 return P_i
```

Fig. 11. Greedy Algorithm

induced from a power assignment P_i at iteration i, we define the cost of an edge $e(u,v)$ as $cost(e) = cost(u) + cost(v)$, where

$$cost(a) = \begin{cases} t^a_{\widehat{P}^e(a)} & \text{if } \widehat{P}^e(a) > P_i(a) \\ 0 & \text{if } \widehat{P}^e(a) \leq P_i(a) \end{cases}$$

Intuitively, the cost of an edge measures the amount of additional interference created by connecting the nodes of this edge based on the current power assignment P_i. In addition, while connecting edge $e(u,v)$, by increasing the transmission range of u and v, we might also connect u (or v) with other nodes, which are covering u (or v), if these nodes will be covered with the new transmission range of u (or v). Let $C_P(e)$ be the set of connected components which will be connected by selecting edge e based on the current power assignment P. At each iteration i of the algorithm, we will select such an edge $\widehat{e}_i$ such that $cost(\widehat{e}_i)/|C_{P_{i-1}}(\widehat{e}_i)|$ is minimized. Figure 11 gives a pseudo-code of this algorithm.

Lemma 2: Let $\widehat{P}$ be the optimal solution and λ_i denote the number of connected components in the graph induced by P_i. Then for every iteration of this algorithm, it holds that $\frac{cost(\widehat{e}_i).\lambda_{i-1}}{|C_{P_{i-1}}(\widehat{e}_i)|} \leq TNI(\widehat{P})$, where $\widehat{e}_i$ is the edge selected by the algorithm at the i-th iteration.

Proof. The proof of this lemma is similar to the proof in [10], which makes use of an integer linear program formulation of the interference problem, and is therefore omitted.

The $O(\log n)$ approximation ratio follows from the following theorem whose proof can be found in the Appendix.

Theorem 3: Let P^{Greedy} denote the solution obtained from the greedy algorithm. It holds that $TNI(P^{Greedy}) \leq TNI(\widehat{P}).O(\log n)$.

	DMST	IMST	Greedy	NPLS
DMST		39%	34%	100%
IMST	70%		54%	100%
Greedy	82%	71%		100%
NPLS	0%	0%	0%	

(a) Performance comparison of four heuristics

	IMST	NPLS	DMST
Aver	0.39%	144.87%	1.71%
	-		
Min	3.76%	61.29%	-4.03%
Max	5.04%	326.51%	11.57%

(b) Average performance of Greedy Algorithm vs. others

Fig. 12. Experimental Results

4.2 Experimental Results

In our experiments, we only consider geometric graphs. We randomly generate 50 nodes on an area of size of 1000x1000. For each graph, we run all four algorithms. We compare the performance of one algorithm against another. The results are shown in Figure 12(a). As seen in this table, the Greedy algorithm performs better than the others. In fact, in 82 cases (71 cases and 100 cases) out of all 100 test cases, its results are equal or better than the results obtained by DMST (IMST and NPLS, respectively). Moreover, Figure 12(b) shows that the results of the Greedy algorithm is on average 1.83% (3.18% and 148.39%) better than the results obtained by IMST (DMST and NPLS, respectively).

5 Conclusions

In this paper we have studied the total node interference problem in WSNs. We have shown that assigning power levels to a set of nodes in the plane to yield a geometric graph with bounded total node interference is NP-complete, complementing a result in [10] by T. Moscibroda and R. Wattenhofer who proved that the problem is NP-complete for the more general ad hoc metric model. We have also provided a simplified version of their $O(logn)$ approximation algorithm and compared its performance against other heuristics through simulation. The result shows that this algorithm is better than others. An interesting future research topic is how to construct network topologies that have low interference and are fault-tolerant at the same time. Results obtained along this line will be of practical relevance.

References

1. Benkert, M., Gudmundsson, J., Haverkort, H., Wolff, A.: Constructing Interference-Minimal Networks. In: Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2006. LNCS, vol. 3831, pp. 166–176. Springer, Heidelberg (2006)
2. Bilò, D., Proietti, G.: On the Complexity of Minimizing Interference in Ad-Hoc and Sensor Networks. Theoretical Computer Science 402, 43–55 (2008)

3. Buchin, K.: Minimizing the Maximum Interference is Hard (2008), http://arxiv.org/abs/0802.2134
4. Burkhart, M., von Rickenbach, P., Wattenhofer, R., Zollinger, A.: Does Topology Control Reduce Interference? In: MOBIHOC 2004, pp. 9–19 (2004)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. Section 23.2: The algorithms of Kruskal and Prim, pp. 567–574. Press and McGraw-Hill (2001) ISBN 0-262-03293-7
6. Dyer, M., Frieze, A.: Planar 2DM is NP-complete. Journal of Algorithms (7), 174–184 (1986)
7. Halldórsson, M.M., Tokuyama, T.: Minimizing Interference of a Wireless Ad-Hoc Network in a Plane. In: Nikoletseas, S.E., Rolim, J.D.P. (eds.) ALGOSENSORS 2006. LNCS, vol. 4240, pp. 71–82. Springer, Heidelberg (2006)
8. Johansson, T., Carr-Motyckovaá, L.: Reducing Interference in Ad Hoc Networks Through Topology Control. In: DIALM-POMC 2005, pp. 17–23 (2005)
9. Moaveni-Nejad, K., Li, X.-Y.: Low-Interference Topology Control for Wireless Ad Hoc Networks. In: Ad Hoc and Wireless Sensor Networks, vol. 1, pp. 41–64 (2005)
10. Moscibroda, T., Wattenhofer, R.: Minimizing Interference in Ad Hoc and Sensor Networks. In: DIALM-POMC 2005 (2005)
11. Nguyen, T.N., Huynh, D.T.: Minimum Interference Planar Geometric Topology in Wireless Sensor Networks. In: Liu, B., Bestavros, A., Du, D.-Z., Wang, J. (eds.) Wireless Algorithms, Systems, and Applications. LNCS, vol. 5682, pp. 149–158. Springer, Heidelberg (2009)
12. Rickenbach, P.V., Schmid, S., Wattenhofer, R., Zollinger, A.: A Roburst Interference Model for Wireless Ad-Hoc Networks. In: Proc. 19th IEEE Int. Par. and Dist. (2005)
13. Sharma, A., Thakral, N., Udgata, S., Pujari, A.: Heuristics for Minimizing Interference in Sensor Networks. In: Garg, V., Wattenhofer, R., Kothapalli, K. (eds.) ICDCN 2009. LNCS, vol. 5408, pp. 49–54. Springer, Heidelberg (2009)
14. Nguyen, T., Lam, N., Huynh, D.: Minimum Edge Interference in Wireless Sensor Networks. To appear in Proc. of Intern. Conf. on Wireless Algorithms, Systems and Applications (2010)
15. Valiant, L.: Universality Considerations in VLSI Circuits. IEEE Trans. on Compupters C-30, 135–140 (1981)
16. Wu, K.-D., Liao, W.: On Constructing Low Interference Topology in Multihop Wireless Networks. Int. J. of Sensor Networks 2, 321–330 (2007)

APPENDIX

Proof of Claim in Theorem 1

For the "only-if" direction, suppose that ϕ has a satisfying Boolean assignment. We assign power levels to nodes in V as follows:

1. Assign power level r_8 to *variable* node $x \in V'$ if variable x has value $TRUE$; otherwise, assign power level r_3.
2. Assign power level r_8 to nodes i's that are neighbors of *variable* nodes that were assigned power r_8. Other nodes i's are assigned power level r_3.
3. Assign power level r_8 to all nodes n's that are neighbors of the *variable* nodes that were assigned power level r_8.
4. Assign the power level r_3 to all remaining nodes.

Figures 3, 4, 5, 6, 7 and 8 show the *variable* node x with power level r_8 and $\neg x$ with power level r_3. The total interference load of the graph is calculated based on the interference load of the sets of nodes X, O, M, I, C and T. First, note that half of the total *variable* nodes have power level r_8. If such a *variable* node is a degree 3 or degree 4 node, it covers 6 neighbors except for the one that connects with a *straight bridge*. This node covers 10 neighbors. Next, observe that half of nodes i's have power level r_8 to connect to the variable nodes with power level r_8. Each of these nodes i's cover 4 nodes while the other half of nodes i's only cover 1 neighbor each. All center nodes o and p cover 4 neighbors. Only 1/3 of nodes n's on the segment connecting to a *clause* node cover 3 nodes, while the other 2/3 of nodes n's cover only 1 neighbor. Each *clause* node cover 3 neighbors and the rest of nodes cover 2 neighbors each. It is straightforward to verify the total of node interference load is exactly R and the graph is a connected graph.

For the "If" direction, suppose the instance $< V', P, R >$ has a power assignment that yields a connected geometric graph $G'(V', E')$ of which the total interference load $\sum_{v \in V} SE(v) \leq R = SE(X) + SE(O) + SE(M) + SE(I) + SE(C) + SE(T)$, we construct a satisfying Boolean assignment for the PL1-EX-3SAT instance ϕ based on the following observations:

1. Exactly 12 nodes i's in each gadget and 1 node i of each *curved bridge* must connect to 6 *variable* nodes for the graph to be connected and the total interference load in each gadget to be minimum.
2. Exactly 6 *variable* nodes of one of the two gadgets (all x variable nodes or all $(\neg x)$ variable nodes) must connect to nodes i's for the graph to be connected. If any *variable* node from the other set also connects to a node is, then the total interference load is $> R$, a contradiction.
3. The *curved* bridge between two gadgets requires at least one of the *variable* nodes at both ends to have power level r_8 for connection.
4. The *straight* bridge connecting two gadgets allows only one *variable* node of either end to have power level r_8. Otherwise, if both *variable* nodes at both ends have power level r_8, then the total interference load is $> R$.
5. At least one node n has power level r_8 to connect to a *variable* node for the graph to be connected. Furthermore, if there are more than 1 node n's of a clause node connected to *variable* nodes, then the total interference load is $> R$.

From the above observations, we can construct a Boolean assignment for the PL1-EX-3SAT instance ϕ using the following rules:

– If a *variable* node in G' has the power level r_8, then assign the value $TRUE$ to that variable in ϕ; otherwise, assign value $FALSE$.

From Observation 5 it follows that each clause in ϕ is satisfied by exactly 1 literal having value *TRUE*. Moreover, from Observations 1, 2, 3 and 4 the Boolean assignment for ϕ is consistent. This concludes the proof of Theorem 1.

Proof of Theorem 3

The proof is similar to the one in [10], and the details are as follows. By the definition of $C_{P_i}(e)$, the number of connected components λ_i can be computed by

$$\begin{aligned}\lambda_i &= \lambda_{i-1} - |C_{P_{i-1}}(\widehat{e}_i)| + 1 \\ &\overset{Lemma2}{\leq} \lambda_{i-1} - \frac{cost(\widehat{e}_i).\lambda_{i-1}}{TNI(\widehat{P})} + 1 \\ &\leq \lambda_{i-1}.(1 - \frac{cost(\widehat{e}_i)}{2.TNI(\widehat{P})})\end{aligned}$$

Let q be the number of iterations required by the Greedy algorithm. Since $\lambda_0 = |V|$ and $\lambda_q = 1$, it follows that

$$1 = \lambda_q \leq |V| \prod_{i=1}^{q} (1 - \frac{cost(\widehat{e}_i)}{2.TNI(\widehat{P})})$$

By taking the logarithm on both sides and rearranging the inequality, we obtain the following.

$$\begin{aligned}\ln 1 &\leq \ln |V| + \sum_{i=1}^{q} \ln(1 - \frac{cost(\widehat{e}_i)}{2.TNI(\widehat{P})}) \\ &\leq \ln |V| - \sum_{i=1}^{q} \frac{cost(\widehat{e}_i)}{2.TNI(\widehat{P})}\end{aligned}$$

$$\sum_{i=1}^{q} cost(\widehat{e}_i) \leq 2.TNI(\widehat{P}) \ln |V|$$

Since whenever the transmission range of a node is increased, $cost(\widehat{e}_i))$ accounts for the entire cost of this increase. Thus, it holds that $TNI(P^{Greedy})$ is at most $\sum_{i=1}^{q} cost(\widehat{e}_i)$. The theorem therefore follows.