Journal of Interconnection Networks

Average Interference Minimization under the Protocol Model in Wireless Sensor Networks*

HAISHENG TAN

Institute for Interdisciplinary Information Sciences, Tsinghua University
Beijing, 100084, China
tan@tsinghua.edu.cn

TIANCHENG LOU

Google Inc., California, USA
tiancheng.lou@gmail.com

AMY YUEXUAN WANG

Institute for Interdisciplinary Information Sciences, Tsinghua University
Beijing, 100084, China
amywang@tsinghua.edu.cn

YONGCAI WANG

Institute for Interdisciplinary Information Sciences, Tsinghua University
Beijing, 100084, China
wangyc@tsinghua.edu.cn

FRANCIS C. M. LAU

Department of Computer Science, The University of Hong Kong
Pokfulam, Hong Kong, China
fcmlau@cs.hku.hk

Received 5 July 2012
Revised 27 December 2012

Reducing interference is one of the main challenges in wireless communication. How to minimize interference through network topology control in wireless sensor networks is a well-known open algorithmic problem. In this paper, we answer the question of how to minimize the average interference when a node is receiving a message. We adopt the protocol interference model, which defines the interference range of a node to be a constant times larger than its transmission range. We study the problem for nodes arbitrarily deployed in one-dimensional (1D) and two-dimensional (2D) regions respectively. For 1D networks, we propose a fast polynomial-time exact algorithm that can compute the minimum average interference. For 2D networks, we give a proof that the maximum interference can be

*This is an extended version of the preliminary one which appeared in Proceedings of ALGOSENSORS 2011 [8]. One important improvement is that we extend the interference model from the receiver-centric model to the more general protocol model.

bounded while minimizing the average interference. The bound is only related to the distances between nodes but not the network size. Based on the bound, we propose the first exact algorithm to compute the minimum average interference in 2D networks. Optimal topologies with the minimum average interference can be constructed through traceback in both 1D and 2D networks.

1. Introduction

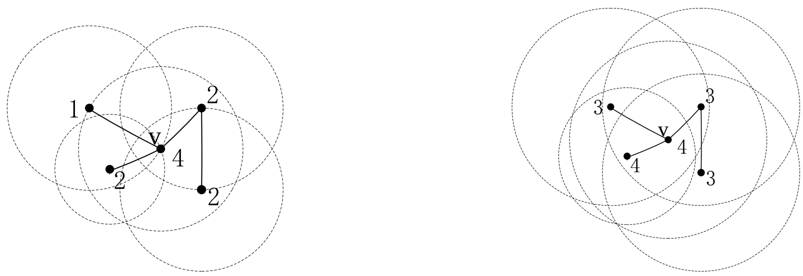
A wireless sensor network (WSN) consists of a set of nodes deployed across a region of interest. The nodes can adjust their transmission powers to achieve their desired transmission ranges with which a multi-hop network is then formed. WSNs have many applications in real life such as environmental monitoring, intrusion detection, and health care.

Energy is a precious resource in wireless sensor networks. One way to conserve energy, and to simultaneously improve communication efficiency, is to reduce *interference* due to concurrent transmissions of two or more nearby nodes. There exist numerous models for capturing the essence of interferences in a wireless network at various abstraction levels of interest. Two types of models that have been frequently studied in recent algorithmic research on wireless sensor networks are *graph-based models* and SINR-based *physical models* [5]. Each type has its own merits (see for instance [11]). SINR-based protocols, which take interference accumulation from all nodes in the network into account, capture more accurately certain important wireless signal propagation characteristics [4, 9, 11, 13]. The graph-based models, although simplistic, are a good estimation of interferences, which have been particularly popular with high-layer protocol designers.

One of the popular graph-based models is the *sender-centric model*, where interference is computed for each edge [1,3,7,12,17]. The interference of an edge (u, v) is the number of other nodes that are covered by the disk centered at u or v with radius $|uv|$—that is, interference is considered at the sender but not the receiver. However, interference actually prevents correct data reception in real networks. Moreover, the sender-centric model is not stable, i.e., by adding a node sufficiently far away, we can create an edge that can interfere with all the other nodes in the network [14]. Thus, the authors in [14] proposed *the receiver-centric model*, where the interference range of a node is set equal to its transmission range, and the interference on a node is the number of other nodes whose transmission ranges cover the node. The receiver-centric model is more realistic than the sender-centric ones. However, in real applications, the interference range of a node is commonly defined larger than its transmission range. In this work, we adopt *the protocol model* [5]. In this model, the interference on a node is likewise equal to the number of other nodes that can interfere it. Different from the receiver-centric model, the protocol model defines the interference radius r_v of a node v to be a constant times larger than its transmission radius r_v^t,

$$r_v = (1 + \delta) r_v^t, \qquad \delta \geq 0, \tag{1.1}$$

where δ is a constant.[a] Figure 1 gives an example of interference under the protocol model, where the transmission radius of a node is set as the distance to its farthest neighbor.



(a) (b)

Fig. 1: The protocol interference model: straight lines are the links; the dashed circle centered at a node indicates its interference range; and the number beside a node is interference on it. (a) $\delta = 0$, and (b) $\delta = 0.5$.

Generally, *topology control* refers to selecting a subset of the available communication links for data transmission, which can help save energy and reduce interference. The problem of minimizing interference through topology control is one of the most well-known open algorithmic problems in wireless communication. Researchers study the problem in two directions: minimizing the maximum interference and minimizing the average interference. Interference minimization is hard because it entails an unusually complicated combinatorial structure, and intuitive ideas such as low node degree, spare topology and Nearest Neighbor Forest (connecting each node to its nearest neighbor) can not guarantee low interference [3, 14].

In the literature, interference minimization is studied in both 1D and 2D networks. Despite their simplicity, 1D networks, i.e. the nodes are arbitrarily distributed along a line, have revealed many interesting challenges and features of the general minimization problem. Studying 1D networks is justified also from a practical point of view as some real networks are one-dimensional, such as sensors deployed along a railway, a corridor, or inside a tunnel. For 1D networks, paper [14] bounded *the minimum maximum interference* (MMI) in the receiver-centric model by $O(\sqrt{\Delta})$ and presented an approximation with ratio $O(\sqrt[4]{\Delta})$. Here Δ is the maximum node degree when each node is connected to all the other nodes within the maximum transmission range r_{max}. The only sub-exponential-time (but super-polynomial) exact algorithm to minimize the maximum interference in the receiver-centric model was given in [15]. For 2D networks, the problem of computing the MMI in the receiver-centric model was shown to be NP-complete in [2]. The algorithm in [6] can bound the maximum receiver-centric interference by $O(\sqrt{\Delta})$. For the problem of

[a]Therefore, the receiver-centric model is a special case of the protocol model when $\delta = 0$.

computing *the minimum average interference* (MAI) in the receiver-centric model, better results are known. In papers [15,16], a polynomial-time, $O(n^3\Delta^3)$-time exact algorithm is proposed and further improved to $O(n^3\Delta)$ for minimizing the average interference in a 1D network, where n is the network size. For 2D networks, the authors of [10] gave an asymptotically optimal approximation algorithm with an approximation ratio $O(\log n)$. To our knowledge, there are few works on designing exact algorithms to minimize the average or the maximum interference in the general protocol model where the interference range is larger than the transmission range. In the following, interference means the interference under the protocol model unless specified otherwise.

Our Contribution: In this paper, we answer the question of how to minimize the average interference when a node is receiving a message under the protocol model.

(1) To minimize the average interference in 1D networks, we propose an exact algorithm that substantially improves the time complexity to $O(n\Delta^2)$. The fastest algorithm known previously used $O(n^3\Delta)$ time [16], which is under the receiver-centric model.
(2) In previous works, the MAI and the MMI were studied separately. We give a proof that the maximum interference can be bounded by $O(\log\lambda)$ while minimizing the average interference. Here $\lambda = \frac{\min(d_{max}, r_{max})}{d_{min}}$, where d_{max} and d_{min} are the longest and shortest distance between any two nodes respectively. The upper bound is only determined by the distances between nodes but not the network size.
(3) Based on the upper bound, we propose an exact algorithm to compute the MAI in 2D networks exactly in time $n^{O(m\log\lambda)}$, where m is the minimum number of parallel lines so that all the nodes are located on the lines. Optimal topologies with the MAI can be constructed through traceback. To the best of our knowledge, this is the first exact algorithm that computes the MAI in 2D networks.

The rest of the paper is organized as follows. We give some formal definitions in Section 2. In Section 3, we propose a fast exact algorithm to compute the MAI in 1D networks. The upper bound of the MMI while minimizing the average interference is proved in Section 4. Section 5 presents the exact algorithm to compute the minimum average interference in 2D networks. Section 6 concludes the paper and suggests some future work.

2. Problem Formulation

We model a wireless sensor network as an undirected graph $G = (V, E)$, where $V = \{v_0, v_1, \cdots, v_{n-1}\}$ is the set of nodes and E is the set of communication links. The nodes have the same maximum transmission radius, denoted as r_{max}. Each node can self-adjust its transmission radius from 0 to r_{max} in a continuous manner. An edge $(u, v) \in E$ exists only if both transmission radii, r_u and r_v, are not shorter than the Euclidean distance $|uv|$. Therefore, in G, the transmission radius of a node is set to the distance to its farthest neighbor. (Two nodes are neighbors means there is an edge incident on them.) We assume the unit disk graph $UDG(V)$, in which each node connects to all the other nodes within a distance of r_{max}.

We adopt the protocol interference model, where the interference radius r_v of a node v is a constant times larger than its transmission radius r_v^t (see Equation 1.1). The interference of a node v, denoted as $RI(v)$, is the number of other nodes whose interference ranges cover v:

$$RI(v) = |\{u|u \in V \setminus \{v\}, |uv| \leq (1+\delta)r_u^t\}|. \tag{2.1}$$

The average node interference in G, $RI_{avg}(G)$, is defined as:

$$RI_{avg}(G) = \frac{\sum_{v \in V} RI(v)}{|V|}. \tag{2.2}$$

For a node v, the interference created by v with transmission radius r_v^t is defined as the number of the other nodes covered by the interference range of v:

$$CI(v, r_v^t) = |\{u|u \in V \setminus \{v\}, |uv| \leq (1+\delta)r_v^t\}|. \tag{2.3}$$

Therefore, we have

$$RI_{avg}(G) = \frac{\sum_{v \in V} RI(v)}{|V|} = \frac{\sum_{v \in V} CI(v, r_v^t)}{|V|}. \tag{2.4}$$

Deleting an edge will not increase any interference. Therefore, the optimal connected topology with minimum interference should be a spanning tree. Therefore, our problem can be defined as:

Definition 2.1. Given n nodes arbitrarily distributed in a 1D or 2D region, construct a spanning tree, $G = (V, E)$, to connect all the nodes with edges no longer than r_{max} and that induces the minimum average interference.

3. Minimizing Average Interference in 1D networks

3.1. *Independent Subproblems*

For a 1D network, the nodes are arbitrarily deployed along a line from left to right. We can view the line as an x-axis, and set $v_0 = 0$. For a segment $\overline{v_s v_t}$ on the line, where $s \leq t$, the nodes *located on* $\overline{v_s v_t}$ are $\{v_s, v_{s+1}, \cdots, v_{t-1}, v_t\}$; the nodes outside $\overline{v_s v_t}$ are the other nodes not including the ones that are on the line; and the nodes *inside* $\overline{v_s v_t}$ are $\{v_{s+1}, \cdots, v_{t-1}\}$.

We draw all the edges on one side of the line. A *cross* is defined as two edges that share at least a common point excluding their endpoints. Paper [16] presents the *no-cross property* in the receiver-centric model. The property also holds under the protocol model.

Theorem 3.1 (No-cross Property) *For any spanning tree connecting the nodes on a line with crosses, there is always another spanning tree to remove the crosses without increasing interference on any node.*

Proof. The proof in [16] proposes a method to delete a cross to preserve the network connectivity without increasing the transmission radius of any node. Therefore, the deletion will not increase the interference radius of a node under the protocol model. That is to say, we can also use the method to delete all the crosses under the protocol model. □

Based on the no-cross property, if there is an edge $(v_s v_t)$, $s < t$, the nodes inside the segment $\overline{v_s v_t}$ cannot be adjacent to the nodes outside. Further, according to Equation 3.9, we compute the average interference using the sum of the interferences created by all the nodes. The interference created by a node is only related to its interference radius and the positions of the other nodes. Recall that the interference radius is a constant times the node transmission radius, which is set to be the distance to its farthest neighbor, and the nodes are stationary after deployment. Therefore, for an edge (v_s, v_t), $s < t$, the total interference created by the nodes inside $\overline{v_s v_t}$ is independent of the topology of the nodes outside, and vice versa. Thus, we can now compute the MAI in 1D networks through dynamic programming.

3.2. *Algorithms*

For $s < t$, we define a topology $A(s,t)$, called an *arch*, for the nodes from v_s to v_t, such that 1) there is an edge (v_s, v_t); 2) $A(s,t)$ is a connected subgraph; and 3) there is no cross. In addition, several auxiliary functions are defined in Table 1.

Table 1: Definition of the functions $(s < t)$

Function	Definition
$f(s,t)$	In $A(s,t)$, returns the minimum total interference created by the nodes inside $\overline{v_s v_t}$[1]
$f_1(s,p,m)$	In $A(s,t)$ and $s \leq p < m < t$, returns the minimum total interference created by nodes inside $\overline{v_s v_m}$ when v_p is the leftmost node adjacent to v_m.
$f_2(m,p,t)$	In $A(s,t)$ and $s < m < p \leq t$, returns the minimum total interference created by nodes inside $\overline{v_m v_t}$ when v_p is the rightmost node adjacent to v_m.
$f_1'(s,m)$	In $A(s,t)$ and $s \leq m < t$, returns the minimum total interference created by nodes $v_{s+1}, v_{s+2}, \cdots, v_m$.
$f_2'(m,t)$	In $A(s,t)$ and $s < m \leq t$, returns the minimum total interference created by nodes $\{v_m, v_{m+1}, \cdots, v_{t-1}\}$.
$g(p,m)$	When v_p is the leftmost node adjacent to v_m, returns the minimum total interference created by nodes $\{v_0, v_1, \cdots, v_{m-1}\}$.

[1] By definition, $f(s,t)$ does not count the interference created by the endpoint v_s or v_t.

As there is no cycle, in $A(s,t)$, there must be a node v_m $(s \leq m < t)$

such that no other links cross the line $x = \frac{v_m+v_{m+1}}{2}$ except (v_s, v_t) (Figure 2). The minimum total interference created by nodes $\{v_{s+1}, v_{s+2}, \cdots, v_m\}$ and nodes $\{v_{m+1}, v_{m+2}, \cdots, v_{t-1}\}$) are $f_1^{'}(s,m)$ and $f_2^{'}(m+1,t)$ respectively. So, we calculate

$$f(s,t) = \begin{cases} 0 & s+1=t, \\ \min_{s\le m<t}\{f_1^{'}(s,m) + f_2^{'}(m+1,t) & \text{otherwise.} \end{cases} \tag{3.1}$$

Here, by definition, we have

$$f_1'(s,m) = \min_{s\le p<m}\{f_1(s,p,m) + CI(v_m, |v_p v_m|)\}, \tag{3.2}$$

$$f_2'(m,t) = \min_{m<p\le t}\{f_2(m,p,t) + CI(v_m, |v_m v_p|)\}. \tag{3.3}$$

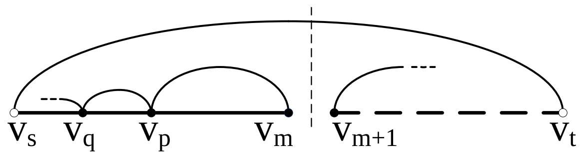


Fig. 2: The structure of $A(s,t)$: $f_1^{'}(s,m)$ is the minimum total interference created by the nodes on the solid segment from v_s to v_m (the left endpoint v_s excluded), and $f_2^{'}(m+1,t)$ that on the dashed segment from v_{m+1} to v_t (the right endpoint v_t excluded).

Specifically, we show how to compute $f_1(s,p,m)$, where v_p is the leftmost node adjacent to v_m (Figure 2):

- When $p = s$,

$$f_1(s,p,m) = f_1(s,s,m) = f(s,m). \tag{3.4}$$

- When $s < p < m$, let v_q be the leftmost node adjacent to v_p. The minimum total interference created by the nodes inside $\overline{v_p v_m}$ and $\overline{v_s v_p}$ are $f(p,m)$ and $f_1(s,q,p)$ respectively. The interference created by node v_p is $CI(v_p, \max(|v_p v_q|, |v_p v_m|))$. Thus we have

$$f_1(s,p,m) = \min_{s\le q<p}\big\{f_1(s,q,p) + f(p,m) + CI(v_p, \max(|v_p v_q|, |v_p v_m|))\big\}. \tag{3.5}$$

Similarly, to calculate $f_2(m,p,t)$, we have

$$f_2(m,p,t) = \begin{cases} f(m,t) & p=t \\ \min_{p<q\le t}\big\{f_2(p,q,t) + f(m,p) & \\ \qquad +CI(v_p, \max(|v_p v_q|, |v_p v_m|))\big\} & m<p<t. \end{cases} \tag{3.6}$$

With $f(s,t)$, the function $g(p,m)$ can be computed as follows:

- When $|v_0 v_m| \le r_{max}$ and $p = 0$, by definition

$$g(p,m) = f(0,m) + CI(v_0, |v_0 v_m|). \tag{3.7}$$

- When $p > 0$ and $|v_p v_m| \leq r_{max}$, let v_q be the leftmost node adjacent to node v_p. The minimum total interference created by the nodes $\{v_0, v_1, \cdots, v_{p-1}\}$ is $g(q, p)$, and by the nodes inside $\overline{v_p v_m}$ is $f(p, m)$. The interference created by node v_p is $CI(v_p, \max(|v_p v_q|, |v_p v_m|))$. Thus, we obtain

$$g(p, m) = \min_{0 \leq q < p \ \& \ |v_p v_q| \leq r_{max}} \left\{ g(q, p) + CI(v_p, \max(|v_p v_q|, |v_p v_m|)) + f(p, m) \right\} \quad (3.8)$$

Finally, the minimum average interference of the total network, AVG_{min}, can be calculated by enumerating the leftmost neighbor v_p of node v_{n-1}:

$$AVG_{min} = \min_{0 \leq p < n-1 \ \& \ |v_p v_{n-1}| \leq r_{max}} \left\{ g(p, n-1) + CI(v_{n-1}, |v_{n-1} v_p|) \right\} \times \frac{1}{n}. \quad (3.9)$$

3.3. *Improved Algorithms*

Now, we show how to compute $f_1(s, p, m)$ efficiently. Firstly, we define

$$f_1^{[1]}(s, p, m) = \min_{s \leq q < p \ \& \ |v_p v_q| \geq |v_p v_m|} \left\{ f_1(s, q, p) + f(p, m) + CI(v_p, |v_p v_q|) \right\} \quad (3.10)$$

$$f_1^{[2]}(s, p, m) = \min_{s \leq q < p \ \& \ |v_p v_q| < |v_p v_m|} \left\{ f_1(s, q, p) + f(p, m) + CI(v_p, |v_p v_m|) \right\} \quad (3.11)$$

Thus, we can rewrite Equation 3.5 ($s < p < m$) as

$$f_1(s, p, m) = \min\{f_1^{[1]}(s, p, m), f_1^{[2]}(s, p, m)\}. \quad (3.12)$$

In Equations 3.10 and 3.11, the values of q are continuous numbers, and the range of q, $\{s \leq q < p \ \& \ |v_p v_q| \geq |v_p v_m|\}$, can be computed in constant time by some simple pre-comparisons with time complexity $O(n\Delta)$. Thus, we can use the RMQ (Range Minimum Query) [18] to compute them efficiently.

Definition 3.1. For an array $A = \{a_0, a_1, \cdots, a_{m-1}\}$, where m is the size of A, and the entries in A are from a linearly ordered set (under the relation $\leq$), a Range Minimum Query (RMQ) asks for the minimum element in the query range $q = [a, b] \subseteq [0, m-1]$, i.e.,

$$RMQ[A, q] = \min_{i \in q} a_i.$$

An RMQ oracle on array A, denoted as $\mathcal{O}_A(q)$, is defined as an oracle that can answer an online query in any query range q within constant time.

The RMQ oracle can be constructed with linear-time $O(m)$ preprocessing [18].

For each $A(s, t)$ ($s < t$), we define two arrays $\alpha(s, t)$ and $\beta(s, t)$ of size $(t - s)$ as:

$$\alpha(s, t)[i] = f_1(s, s + i, t), \qquad i = 0, \cdots, t - s - 1, \quad (3.13)$$

and

$$\beta(s, t)[i] = f_1(s, s + i, t) + CI(v_t, |v_t v_{s+i}|), \qquad i = 0, \cdots, t - s - 1. \quad (3.14)$$

We construct RMQ oracles on each $\alpha(s,t)$ and $\beta(s,t)$. Then, we have

$$\begin{aligned} f_1^{[1]}(s,p,m) &= \min_{s\le q<p \ \& \ |v_pv_q|\ge|v_pv_m|} \{f_1(s,q,p)+CI(v_p,|v_pv_q|)\}+f(p,m) \\ &= \mathcal{O}_{\beta(s,p)}\Big(\{q-s|s\le q<p \ \& \ |v_pv_q|\ge|v_pv_m|\}\Big)+f(p,m), \end{aligned} \quad (3.15)$$

and

$$\begin{aligned} f_1^{[2]}(s,p,m) &= \min_{s\le q<p \ \& \ |v_pv_q|<|v_pv_m|} \{f_1(s,q,p)+f(p,m)+CI(v_p,|v_pv_m|)\} \\ &= \mathcal{O}_{\alpha(s,p)}\Big(\{q-s|s\le q<p \ \& \ |v_pv_q|<|v_pv_m|\}\Big) \\ &\quad +f(p,m)+CI(v_p,|v_pv_m|). \end{aligned} \quad (3.16)$$

The function $f_2(m,p,t)$ can be computed similarly.

The whole algorithm is described in Algorithm 1:

- Lines 1–15 are to compute each $f(s,t)$ when $0\le s<t<n-1$ and $|v_sv_t|$ does not exceed the maximal transmission radius r_{max} through calling Algorithm 2.
- Lines 6–13 are to prepare the RMQ oracles for computing $f_1(s,p,m)$ and $f_2(s,p,m)$ in the coming round of the loop.
- Based on the values of the f functions, Lines 16–26 are to compute each $g(p,m)$ according to Equations 3.7 and 3.8.
- Based on the values of the g functions, Lines 27–30 compute and return the value of the MAI according to Equation 3.9.

The following explains the computation of $f(s,t)$ in Algorithm 2.

- Lines 1–7 are to compute $f(s,t)$ according to Equation 3.1. Lines 1–2 are the initial case. Lines 4–7 are based on the values of f_1' and f_2' which have been computed in the previous rounds.
- Lines 8–16 are to compute the values of $f_1(s,p,t)$ based on Equations 3.4 and 3.12, where Equation 3.4 is the initial case in Line 8, and Equation 3.12 is the improved method with the RMQ oracles. The values of $f_2(s,p,t)$ are computed similarly.
- Based on the values of f_1 and f_2, Lines 17–24 compute the values of $f_1'(s,t)$ and $f_2'(s,t)$ according to Equations 3.2 and 3.3.

3.4. *Analysis*

Our algorithm actually compares the average interference on all the spanning trees without a cross, which guarantees that the output is optimal with the MAI. Further, our methods have also been verified by comparing the results with the outputs generated by the brute-force search, which runs slowly in time $O(n^{\Delta})$.

According to the process of dynamic programming, the computation of the d-ifferent functions $f_1(s,p,m)$ and $f_2(m,p,t)$ (as defined in Table 1) contributes the main part of the time complexity. Δ is the maximum number of neighbors for a

Algorithm 1: Compute MAI of n nodes $\{v_0, \cdots, v_{n-1}\}$ in a 1D network

$s \leftarrow n-2$;
while $s \geq 0$ **do**
    $t \leftarrow s+1$;
    while $t < n$ *and* $|v_s v_t| \leq r_{max}$ **do** `/* all subranges of` $[s,t]$ `have been computed */`
        Call Algorithm 2 to compute $f(s,t)$ and all related functions;
        foreach $p \in [s, t-1]$ **do**
            $\alpha(s,t)[p-s] = f_1(s,p,t)$;
            $\beta(s,t)[p-s] = f_1(s,p,t) + CI(v_t, |v_t v_p|)$;
        end
        Construct two RMQ oracle $\mathcal{O}_{\alpha(s,t)}$ and $\mathcal{O}_{\beta(s,t)}$ on arrays $\alpha(s,t)$ and $\beta(s,t)$;
        foreach $p \in [s+1, t]$ **do**
            $\gamma(s,t)[t-p] = f_2(s,p,t)$;
            $\eta(s,t)[t-p] = f_2(s,p,t) + CI(v_s, |v_s v_p|)$;
        end
        Construct two RMQ oracle $\mathcal{O}_{\gamma(s,t)}$ and $\mathcal{O}_{\eta(s,t)}$ on arrays $\gamma(s,t)$ and $\eta(s,t)$;
        $t \leftarrow t+1$;
    end
    $s \leftarrow s-1$;
end
$p \leftarrow 0$;
while $p < n-1$ **do**
    foreach $m \in [p+1, n-1]$ *and* $|v_p v_m| \leq r_{max}$ **do**
        if $p = 0$ **then**
            $g(p,m) \leftarrow f(0,m) + CI(v_0, |v_0 v_m|)$;
        end
        else
            $z_0 \leftarrow \infty$;
            foreach $q \in [0, p-1]$ *and* $|v_p v_q| \leq r_{max}$ **do**
                $z_0 \leftarrow \min(z_0, g(q,p) + CI(v_p, \max(|v_p v_q|, |v_p v_m|)) + f(p,m))$;
            end
            $g(p,m) \leftarrow z_0$;
        end
    end
    $p \leftarrow p+1$;
end
$AVG_{min} \leftarrow \infty$;
foreach $p \in [0, n-2]$ *and* $|v_p v_{n-1}| \leq r_{max}$ **do**
    $AVG_{min} \leftarrow \min(AVG_{min}, \frac{1}{n} \cdot g(p, n-1) + CI(v_{n-1}, |v_{n-1} v_p|))$;
end
return AVG_{min};

Algorithm 2: Subfunction to compute $f(s,t)$ and all related functions

if $s+1=t$ **then**
    $f(s,t) \leftarrow 0;$
end
else
    $z \leftarrow \infty;$
    foreach $m \in [s, t-1]$ **do**
        $z \leftarrow \min(z, f_1^{'}(s,m) + f_2^{'}(m+1,t));$
    end
    $f(s,t) \leftarrow z;$
end
$f_1(s,s,t) \leftarrow f(s,t);$
$f_2(s,t,t) \leftarrow f(s,t);$
foreach $p \in [s+1, t-1]$ **do**
    $f_1^{[1]}(s,p,t) \leftarrow \mathcal{O}_{\beta(s,p)}\Big(\{q-s \mid s \leq q < p \text{ \& } |v_pv_q| \geq |v_pv_t|\}\Big) + f(p,t);$
    $f_1^{[2]}(s,p,t) \leftarrow \mathcal{O}_{\alpha(s,p)}\Big(\{q-s \mid s \leq q < p \text{ \& } |v_pv_q| <$
    $|v_pv_t|\}\Big) + f(p,t) + CI(v_p, |v_pv_t|);$
    $f_1(s,p,t) \leftarrow \min(f_1^{[1]}(s,p,t), f_1^{[2]}(s,p,t));$
    $f_2^{[1]}(s,p,t) \leftarrow \mathcal{O}_{\eta(p,t)}\Big(\{t-q \mid p < q \leq t \text{ \& } |v_pv_q| \geq |v_pv_s|\}\Big) + f(s,p);$
    $f_2^{[2]}(s,p,t) \leftarrow \mathcal{O}_{\gamma(p,t)}\Big(\{t-q \mid p < q \leq t \text{ \& } |v_pv_q| <$
    $|v_pv_s|\}\Big) + f(s,p) + CI(v_p, |v_pv_s|);$
    $f_2(s,p,t) \leftarrow \min(f_2^{[1]}(s,p,t), f_2^{[2]}(s,p,t));$
end
$z_1 \leftarrow \infty;$
$z_2 \leftarrow \infty;$
foreach $p \in [s, t-1]$ **do**
    $z_1 \leftarrow \min(z_1, f_1(s,p,t) + CI(v_t, |v_pv_t|));$
end
foreach $p \in [s+1, t]$ **do**
    $z_2 \leftarrow \min(z_2, f_2(s,p,t) + CI(v_s, |v_pv_s|));$
end
$f_1^{'}(s,t) \leftarrow z_1;$
$f_2^{'}(s,t) \leftarrow z_2;$

node constrained by the maximum transmission radius r_{max}. v_t is a neighbor of v_s. For a given s, there are at most Δ different choices of t and at most $t-s$ choices of m. Since all the nodes are deployed along a line, $t-s \leq \Delta$. Also, for a given m, there are at most Δ choices of p as v_p is a neighbor of v_m. Therefore, the total number of different functions $f_1(s,p,m)$ is $O(n\Delta^2)$. A similar result can be achieved for $f_2(m,p,t)$. For a similar reason, the overall preprocessing time of all RMQ oracles is $O(n\Delta^2)$. With dynamic programming, each function $f_1(s,p,m)$ and $f_2(m,p,t)$ can be computed in $O(1)$ time. Thus, the time complexity to compute the MAI in 1D networks is $O(n\Delta^2)$. The optimal spanning tree can be computed through traceback efficiently.

4. Bound on MMI while Minimizing Average Interference

In this section, we derive an upper bound on the MMI while minimizing the average interference.

4.1. *Preliminaries*

Firstly, we define the following property, dubbed the EX property which stands for 'mutual EXclusion of the long edges'.

Definition 4.1 (EX property) *For four nodes a, b, c, and d, if* $\min(|ab|,|cd|) > \max(|ad|,|bc|)$*, the edges (a,b) and (c,d) are not in a spanning tree simultaneously. It also holds when $a = d$.*

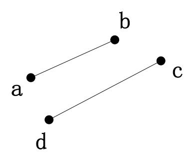


Fig. 3: Four nodes in $T = (V,E)$

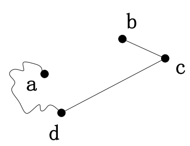


Fig. 4: Replace (a,b) with (b,c)

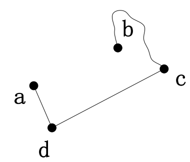


Fig. 5: Replace (a,b) with (a,d)

Next, we show that we can always find an optimal spanning tree with the MAI that satisfies the EX property.

Theorem 4.1. For a set of nodes V deployed in a 2D plane, there is always a spanning tree, $T_{ex} = (V, E_{ex})$, with the MAI that satisfies the EX property.

Proof. For a spanning tree $T = (V,E)$ with the MAI, if it satisfies the EX property, we set $T_{ex} = T$ and we have the proof. If not, we can construct T_{ex} as follows. For each set of four nodes a, b, c and d such that $\min(|ab|,|cd|) > \max(|ad|,|bc|)$ and $(a,b) \in E, (c,d) \in E$ (note that here a and d can be the same node) (Figure 3),

(1) if a has a path to d in the graph $T_1(V, E - \{(a,b),(c,d)\})$, we set $E' = E - (a,b) + (b,c)$ (Figure 4);
(2) if a does not have a path to d in the graph $T_1(V, E - \{(a,b),(c,d)\})$, we set $E' = E - (a,b) + (a,d)$ (Figure 5).

Firstly, we show that T_{ex} is a spanning tree. According to the construction of T_{ex}, in case 1, as a and d have a path, the four nodes are still connected and $|E_{ex}| = |E| = n - 1$; therefore, T_{ex} is a spanning tree. The same result can be obtained similarly for case 2. Secondly, we show that T_{ex} also has the MAI. In case 1, we delete (a, b) and add (b, c). As $|bc| < |ab|$ and $|bc| < |cd|$, the modification does not increase the transmission radius of any node, which means that the total interference created by the nodes is not increased. The same conclusion applies to case 2. Thus, T_{ex} is a spanning tree with the MAI that satisfies the EX property. The theorem is proved. □

As T_{ex} satisfies the EX property, we have

Corollary 4.1. *For two regions S_1 and S_2 of diameters d_1 and d_2 respectively, there is at most one edge $(u, v) \in E_{ex}$ such that $|u, v| > \max(d_1, d_2)$ with $u \in S_1$ and $v \in S_2$. (Figure 6).*

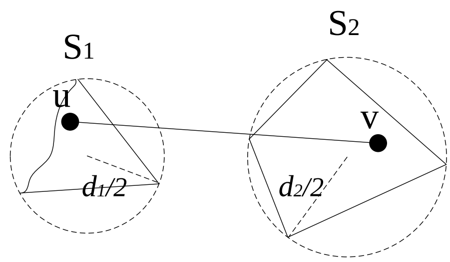


Fig. 6: There is at most one edge $(u, v) \in E_{ex}$ where $u \in S_1$, $v \in S_2$ and $|u, v| > \max(d_1, d_2)$

4.2. *The Upper Bound*

According to Corollary 4.1, we can bound the maximum interference in T_{ex} as described in Theorem 4.2.

Theorem 4.2. In the spanning tree T_{ex}, the maximum interference is bounded by $O(log\lambda)$, where $\lambda = \frac{\min(d_{max}, r_{max})}{d_{min}}$. d_{max} and d_{min} are the longest and shortest distance between any two nodes respectively.

Proof. For any node $v \in E_{ex}$, the set H contains the other nodes that can interfere with v. We separate the elements in H into subsets according to their transmission radii as follows:

$$h_i = \{u | u \in H \quad and \quad (1+\epsilon)^{i-1} d_{min} \le r_u < (1+\epsilon)^i d_{min}\}, \quad i = 1, 2, 3 \cdots \tag{4.1}$$

where ϵ is a positive constant. The subsets have the following properties:

$$H = \sum_i h_i, \quad and \quad \{h_i \cap h_j = \varnothing \quad if \quad i \neq j\}. \tag{4.2}$$

Since the possible longest transmission radius in T_{ex} is $\lambda \times d_{min}$, we have the maximal i, denoted as i_{max}, as follows.

$$(1+\epsilon)^i \leq \lambda \Rightarrow i_{max} = O(\log \lambda). \tag{4.3}$$

As the transmission radii of the nodes in h_i are smaller than $(1+\epsilon)^i d_{min}$, the nodes are all inside the circle[b] $c(v, (1+\delta)(1+\epsilon)^i d_{min})$. Further, the nodes in h_i and their neighbors are all inside the circle $c(v, (2+\delta)(1+\epsilon)^i d_{min})$. We use a set of squares, the length of whose edges is $\frac{\sqrt{2}}{4}(1+\epsilon)^{i-1} d_{min}$, to fully cover the area inside the circle $c(v, (2+\delta)(1+\epsilon)^i d_{min})$. So, the number of the squares needed is

$$c_0 = (\lceil \frac{2 \times (2+\delta)(1+\epsilon)^i d_{min}}{\frac{\sqrt{2}}{4}(1+\epsilon)^{i-1} d_{min}} \rceil)^2 = (\lceil 4\sqrt{2}(2+\delta)(1+\epsilon) \rceil)^2. \tag{4.4}$$

For each node $u \in h_i$, since $r_u \geq (1+\epsilon)^{i-1} d_{min}$, u must have an edge $(uu') \in E_{ex}$ which lies inside the circle $c(v, 2(1+\epsilon)^i d_{min})$ such that $|uu'| \geq (1+\epsilon)^{i-1} d_{min}$.

The diameter of each square is $\frac{(1+\epsilon)^{i-1} d_{min}}{2}$. According to Corollary 4.1, for each pair of the squares, s_1 and s_2, there is at most one edge $(v_1 v_2)$ such that $|v_1 v_2| \geq (1+\epsilon)^{i-1} d_{min}$ and $v_1 \in s_1$, $v_2 \in s_2$. Therefore, the number of nodes in h_i is:

$$|h_i| \leq 2 \times \binom{c_0}{2} = c_1 \tag{4.5}$$

where c_1 is a constant. Based on Equation 4.3, the interference on the node v is

$$RI(v) = |H| = \sum_i |h_i| \leq c_1 \times i_{max}. \tag{4.6}$$

According to Equations 4.3 and 4.6, we have

$$RI(v) = O(\log \lambda). \tag{4.7}$$

Therefore, the maximum interference in T_{ex} is bounded by $O(\log \lambda)$. The theorem is proved. □

Based on the above theorem, we have the following corollary:

Corollary 4.2. *In 2D networks, the MMI is bounded by $O(\log \lambda)$ while minimizing the average interference.*

5. Minimizing Average Interference in 2D Networks

5.1. *Basic Ideas*

Given n nodes arbitrarily deployed in a 2D region, we can simply find the minimum number, denoted as m, of parallel lines so that all the nodes are located on the lines

[b] $c(v, r)$ stands for a circle centering at point v with radius of r.

(Figure 7). We set a parallel line as the x-axis, and refer to the n nodes from left to right as $V = \{v_0, v_1, \cdots, v_{n-1}\}$, where for two nodes $v_i = (x_i, y_i)$ and $v_j = (x_j, y_j)$,

$$i < j \quad iff \quad x_i < x_j \ or \ \{x_i = x_j \ and \ y_i < y_j\}. \tag{5.1}$$

According to Equation 4.6, we can construct the topology with the MAI while the maximum interference does not exceed $k = \min(c_1 \times i_{max}, n-1)$. Here, we restrict the maximum interference because it is a critical parameter in determining the time complexity of our algorithm which we analyze in Section 5.3.

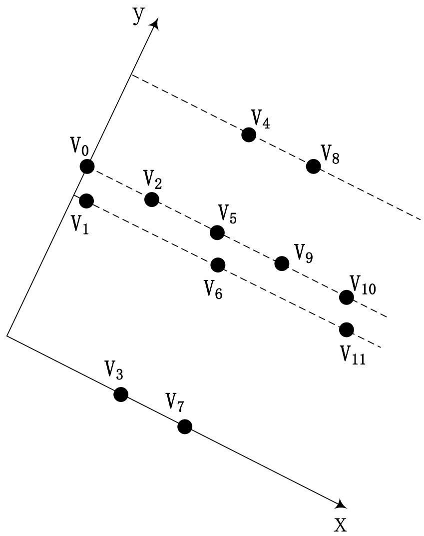


Fig. 7: 12 nodes deployed in a 2D region with the minimum number of parallel lines covering them.

We assume a virtual line *clin* that separates the nodes into the left and right parts. Initially, there is only v_0 on the left of *clin*. We move rightward (and rotate if necessary) the line to include one more node on its left each time until all the nodes are on the left of *clin*. When moving *clin* to include v_p $(0 \leq p < n)$ in the left part, we compute the minimum total interference created by the nodes inside $[0, p]$,[c] while the maximum interference does not exceed k and the total topology for the n nodes is connected. Here, the nodes on the left of *clin* may connect to and interfere with the nodes on the right, and vice versa. When computing the topology for the nodes on the left of *clin*, we need to assume a topology on the right and take the mutual interference into account. Thus, for an interval $[s, t]$ $(0 \leq s \leq t \leq n-1)$, we define the following items:

[c]For an interval $[s, t]$, $s \leq t$, the nodes inside $[s, t]$ are the ones from v_s to v_t. The nodes outside $[s, t]$ are the ones on the left of $[s, t]$ (the nodes from v_0 to v_{s-1}) and on the right of $[s, t]$ (the nodes from v_{t+1} to v_{n-1}).

- c[s,t], which records the interference from the nodes inside $[s, t]$ to the nodes outside, i.e., $c[s, t]$ contains the nodes (together with their transmission radii) that can interfere with the nodes outside $[s, t]$.
- $s[s, t]$, which records the connectivity of the nodes inside $c[s, t]$, i.e., $s[s, t]$ stores all the connected components (or subsets) of the nodes in $c[s, t]$.

Note that the nodes recorded in $c[s, t]$ and in $s[s, t]$ are the same set of nodes. As the maximum interference does not exceed k, we call $c[s, t]$ *valid* if and only if there are no more than k nodes inside $[s, t]$ that interfere with the same node outside $[s, t]$. With the above definitions, we now introduce the algorithms to compute the MAI while the maximum interference does not exceed k.

5.2. *Algorithms to Compute MAI*

We define a function $F(p, c[0, p], c[p + 1, n - 1], s[0, p])$, $0 \leq p < n - 1$, to construct a topology minimizing the interference created by the nodes inside $[0, p]$ while satisfying the following conditions:

(1) the interference from nodes inside $[0, p]$ to the nodes inside $[p + 1, n - 1]$ is the same as that recorded in $c[0, p]$;
(2) the interference from nodes inside $[p + 1, n - 1]$ to the nodes inside $[0, p]$ is the same as that recorded in $c[p + 1, n - 1]$;
(3) the connectivity of the nodes in $c[0, p]$ is the same as that recorded in $s[0, p]$;
(4) all the nodes inside $[0, p]$ but not in $c[0, p]$ have a path to at least one node in $c[0, p]$;
(5) the interference on each node inside $[0, p]$ does not exceed k.

If F returns $+\infty$, it means there is no such topology that satisfies all the conditions. Here, conditions 1, 2 and 5 are to guarantee that the maximum interference in the final topology does not exceed k. Conditions 3 and 4 are for the requirement of connectivity. Specifically, condition 4 is to guarantee that the nodes in $[0, p]$ but not in $c[0, p]$ can connect to the nodes in $[p + 1, n - 1]$ through the nodes in $c[0, p]$. The function F can be calculated in Algorithm 3:

- Lines 1–5 are the boundary condition;
- Lines 7–10 are to enumerate the possible situations;
- In Line 10, for a node v, $R(v)$ is defined as $R(v) = \{|uv| | u \in V \ and \ |uv| \leq r_{max}\}$, which is the set of potential transmission radii of v;
- Line 11 is to connect v_p to nodes in $[0, p - 1]$ to maintain connectivity;
- In Line 12, $c'[0, p]$ and $s'[0, p]$, which are defined as the same as $c[0, p]$ and $s[0, p]$ respectively, are computed based on $c[0, p - 1]$, $s[0, p - 1]$ and the newly added edges in Line 11;
- Lines 13–16 are to check all the conditions and update the minimum total interference.

Algorithm 3: Compute $F(p, c[0,p], c[p+1, n-1], s[0,p])$

if $p = 0$ **then** /* the boundary condition */
    if *there are more the* k *nodes in* $c[p+1, n-1]$ *that can interference with* v_0 **then**
        return $+\infty$;
    end
    else
        return $CI(v_0, r_{v_0})$;
    end
end
$total \leftarrow +\infty$;
foreach *valid* $c[0, p-1]$ **do**
    foreach *valid* $c[p, n-1]$ **do**
        foreach $s[0, p-1]$ **do**
            foreach $r_{v_p} \in R(v_p)$ **do**
                Connect v_p to the nodes in $\{v | v\ \textit{is inside}\ [0, p-1]\ \textbf{and}\ |vv_p| \le \min(r_v, r_{v_p})\}$;
                Compute $c'[0,p]$ and $s'[0,p]$;
                if $c[0,p] = c'[0,p]$ ***and*** $s[0,p] = s'[0,p]$ ***and*** *all the nodes in* $[0,p]$ *but not in* $c[0,p]$ *have a path to at least one node in* $c[0,p]$ ***and*** *the interference on* v_p *does not exceed* k **then**
                    $tmp \leftarrow F(p-1, c[0,p-1], c[p,n-1], s[0,p-1]) + CI(v_p, r_{v_p})$;
                    if $tmp < total$ **then**
                        $Total \leftarrow tmp$;
                    end
                end
            end
        end
    end
end
return *Total*;

The MAI of all the nodes can be computed in the algorithm MAI-GRID (Algorithm 4). MAI-GRID checks the interference on v_{n-1} and makes sure that all the nodes in $s[0, n-2]$ have a path to v_{n-1} such that the network connectivity is maintained. MAI-GRID computes the minimum total interference by the sum of interference created by nodes in $[0, n-2]$ and the interference created by v_{n-1}. After computing MAI-GRID, we can also construct the optimal spanning tree with the MAI through traceback. Figure 8 is an example of the optimal topology.

Algorithm 4: MAI-GRID: compute the MAI in a grid network

1 $k \leftarrow \min(c_1 \times i_{max}, n-1)$, $\quad total \leftarrow +\infty$;
2 **foreach** *valid* $c[0, n-2]$ **do**
3   **foreach** $s[0, n-2]$ **do**
4   **foreach** $r_{v_{n-1}} \in R(v_{n-1})$ **do**
5   Connect v_{n-1} to the nodes in $\{v|v \textit{ is inside } [0, n-2] \textbf{ and } |vv_{n-1}| \leq \min(r_v, r_{v_{n-1}})\}$;
6             $c[n-1, n-1] = \{v_{n-1}, r_{v_{n-1}}\}$;
7   **if** *the interference on* v_{n-1} *does not exceed* k ***and*** *all the nodes in* $s[0, n-2]$ *has a path to* v_{n-1} **then**
8                 $t \leftarrow F(n-2, c[0, n-2], c[n-1, n-1], s[0, n-2]) + CI(v_{n-1}, r_{v_{n-1}})$;
9   **if** $t < total$ **then**
10                     $Total \leftarrow t$;
11   **end**
12   **end**
13   **end**
14   **end**
15 **end**
16 **return** $\frac{total}{n}$;

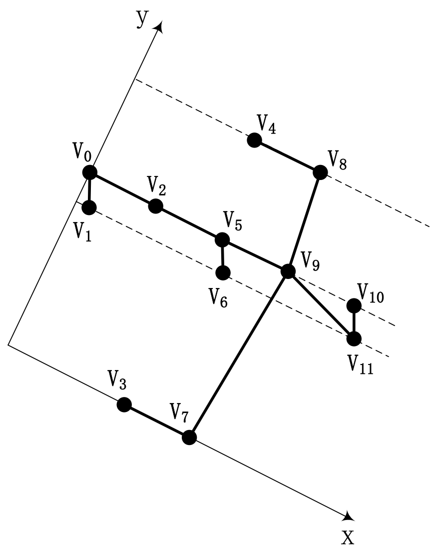


Fig. 8: The optimal topology with the MAI, which is $\frac{29}{12}$.

5.3. *Analysis*

Recall that $c[0, p]$ and $s[0, p]$ record the same set of nodes. Based on the definition of the function F, Condition 4 and the check in Line 7 of Algorithm 4 guarantee the connectivity of our output; Condition 5 and the check of the interference on v_{n-1} (Line 7 in Algorithm 4) guarantee the maximum interference of our output does not exceed k. Further, our algorithm actually compares all the possible connected topologies with the maximal interference equal to or smaller than k. Therefore, the method outputs the optimal topology with the MAI while the maximum interference does not exceed k. The correctness of the algorithms has also been established through comparing our results with the outputs of the brute-force search which runs in time $O(n^{\Delta})$.

The main complexity to construct the optimal spanning tree is to compute the F functions. In our optimal topologies, the maximum interference does not exceed k. If there are more than mk nodes in $c[s,t]$ that interfere with the nodes on the left of $[s,t]$, there must be a parallel line, and the rightmost node on the left of $[s,t]$ on the line will experience an interference larger than k. Therefore, in a valid $c[s,t]$, there are at most $\min(mk,n)$ nodes. Similarly, there are at most $\min(mk,n)$ nodes interfering with one node on the right of $[s,t]$. The number of different transmission radii of a node v is at most Δ. Therefore, the number of valid $c[0,p]$ is $O((n\Delta)^{mk})$. A similar result can be achieved for $c[p+1,n-1]$. The number of variations of $s[0,p]$ is $O((mk)^{mk})$. As $\Delta \leq n-1$ and $k = O(log\lambda)$, the time complexity to construct the optimal spanning tree with the MAI is $n^{mO(\log \lambda)}$.

The minimum number of parallel lines to cover all the nodes can be linear in n, e.g. $m = O(n)$. Therefore, the time complexity is still exponential in the worst case. However, in some cases when the nodes are deployed along a few parallel lines, e.g. m is a small constant, our algorithm is fast.

6. Conclusion

In this paper, we study how to minimize the average interference while preserving connectivity through topology control in wireless sensor networks. The protocol models is adopted, where the interference range of a node is a constant times larger than its transmission range. In 1D networks, based on the no-cross property and dynamic programming, we propose a fast exact algorithm to compute the minimum average interference. In 2D networks, using computational geometry, we prove that the minimum maximum interference can still be bounded while minimizing the average interference. Moreover, we propose exact algorithms to compute the minimum average interference in 2D networks. Future work directions include interference minimization in 3D networks (some real sensor networks are 3D), and how to reduce interference for network properties besides connectivity, such as planarity, low node degree and small spanner.

Acknowledgements

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Foundation of China Grant 61073174, 61103186, 61202360, 61033001, 61061130540, and Hong Kong RGC-GRF grants 714009E, 714311.

References

1. M. Benkert, J. Gudmundsson, H. Haverkort and A. Wolff, Constructing minimum-interference networks, In: *Computational Geometry*, Vol. 40, Issue 3, pp. 179–194, 2008.
2. K. Buchin, Minimizing the Maximum Interference is Hard, In: arXiv: 0802.2134v1, 2008.
3. M. Burkhart, P. von Rickenbach, R. Wattenhofer and A. Zollinger, Does Topology Control Reduce Interference? In: *Proc. of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 9–19, 2004.
4. D. Chafekar, V. S. A. Kumar, M. V. Marathe, S. Parthasarathy and A. Srinivasan, Cross-Layer Latency Minimization in Wireless Networks with SINR Constraints, In: *Proc. of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2007.
5. P. Gupta and P. R. Kumar, The Capacity of Wireless Networks, In: *IEEE Transactions on Information Theory*, Vol. 46(2), pp. 388–404, 2000.
6. M. Halldórsson and T. Tokuyama, Minimizing interference of a wireless ad hoc network in a plane, In: *ALGOSENSORS*, LNCS 4240, pp. 71–82, 2006.
7. T. Johansson and L. Carr-Motyčková, Reducing Interference in Ad Hoc Networks Through Topology Control, In: *the 2nd ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pp. 17–23, 2005.
8. T. Lou, H. Tan, Y. Wang and F. C. M. Lau, Minimizing Average Interference Through Topology Control. In: *Pro. of the 7th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS)*, LNCS 7111, pp. 115-129, 2011.
9. T. Moscibroda, Y. A. Oswald and R. Wattenhofer, How Optimal are Wireless Scheduling Protocols? *In Proc. of the 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2007.
10. T. Moscibroda and R. Wattenhofer, Minimizing interference in ad hoc and sensor networks, In: *Proc. of the 2nd ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pp. 24-33, 2005.
11. T. Moscibroda, R. Wattenhofer and Y. Weber, Protocol Design Beyond Graph-Based Models, In: *Proc. of the 5th Workshop on Hot Topics in Networks (HotNets)*, 2006.
12. K. Moaveni-Nejad and X.-Y. Li, Low-interference topology control for wireless ad hoc networks, In: *Int. J. of Ad Hoc & Sensor Wireless Networks*, Vol. 1, pp.41–64, 2005.
13. T. Moscibroda, R. Wattenhofer and A. Zollinger, Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies. In: *Proc. of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2006.
14. P. von Rickenbach, R. Wattenhofer and A. Zollinger, Algorithmic Models of Interference in Wireless Ad Hoc and Sensor Networks. In: *IEEE/ACM Transation on Networking (TON)*, Vol. 17, Issue. 1, pp. 172–185, 2009.
15. H. Tan, T. Lou, F. C. M. Lau, Y. Wang and S. Chen, Minimizing Interference for the Highway Model in Wireless Ad-hoc and Sensor Networks. In: *Proc. of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, LNCS 6543, pp.520–532, 2011.

16. H. Tan, T. Lou, Y. Wang, Q.-S. Hua, F. C. M. Lau, Exact algorithms to minimize interference in wireless sensor networks. In: *Theoretical Computer Science*, Vol. 412, pp. 6913-6925, 2011.
17. K.-D. Wu and W. Liao, On Constructing Low Interference Topology in Multihop Wireless Networks, In: *International Journal of Sensor Networks*, Vol. 2, pp.321–330, 2007.
18. H. Yuan and M. J. Atallah, Data Structures for Range Minimum Queries in Multidimensional Arrays, In: *Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.