Exact algorithms to minimize interference in wireless sensor networks[☆]

Haisheng Tan [a,1], Tiancheng Lou [a], Yuexuan Wang [a,*], Qiang-Sheng Hua [a], Francis C.M. Lau [b]

[a] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, China
[b] Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong, China

ARTICLE INFO

ABSTRACT

Finding a low-interference connected topology is a fundamental problem in wireless sensor networks (WSNs). The problem of reducing interference through adjusting the nodes' transmission radii in a connected network is one of the most well-known open algorithmic problems in wireless sensor network optimization. In this paper, we study minimization of the average interference and the maximum interference for the highway model, where all the nodes are arbitrarily distributed on a line. First, we prove that there is always an optimal topology with minimum interference that is planar. Then, two exact algorithms are proposed. The first one is an exact algorithm to minimize the average interference in polynomial time, $O(n^3 \Delta)$, where n is the number of nodes and Δ is the maximum node degree. The second one is an exact algorithm to minimize the maximum interference in sub-exponential time, $O(n^3 \Delta^{O(k)})$, where $k = O(\sqrt{\Delta})$ is the minimum maximum interference. All the optimal topologies constructed are planar.

1. Introduction

Wireless sensor networks (WSNs) consist of a set of nodes deployed inside a region of interest. Each node has limited processing ability and is equipped with a wireless radio for communication. Unlike traditional wired networks, they do not have a fixed infrastructure. The nodes can adjust their transmission powers to achieve the desired transmission ranges which would then form a multi-hop network. Wireless sensor networks have many applications in real life such as environmental monitoring, intrusion detection, and health care. It is regarded as one of the most popular networking paradigms of today.

Due to the nature of the environments in which WSNs are typically deployed, wireless nodes can only be powered with relatively weak batteries. Energy is scarce which however is critical for extending the network's lifetime. One way to conserve energy is to reduce *interference* due to concurrent transmissions of close-by nodes. Different models have been proposed and defined to capture the phenomenon of interference. One is called the *sender-centric model*, where interference of each edge is by counting [2,4,7,14,16,19]. 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 $|uv|$, which is to say that interference is considered at the sender but not at the receiver. However, interference actually prevents correct data reception in real networks. Thus, the authors of [20,21] proposed the *receiver-centric model*, where the interference on a node v is the number of other nodes whose transmission ranges cover v. In Fig. 1, the interference on the node v is 3 as all the other nodes can cover it. In this paper, unless specified otherwise, the receiver-centric model is assumed.
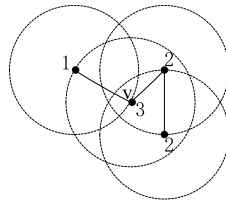
Fig. 1. The receiver-centric interference: the numbers are interference on each node.
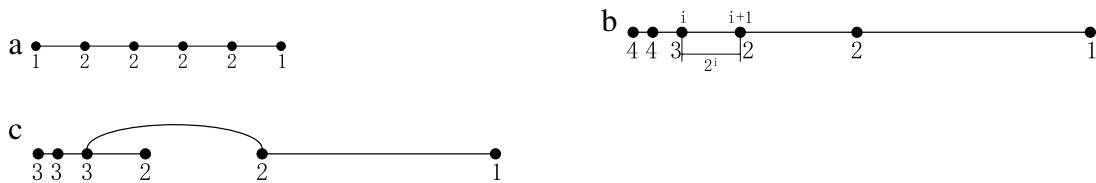

Fig. 2. Low node degree, sparse distribution or connecting each node to the nearest node cannot guarantee low interference: nodes are all distributed along a line and the figures next to the nodes are the node interference. (a) The 6 nodes are evenly distributed and linearly connected; (b) the 6-node exponential chain and nodes are linearly connected; (c) another connected topology for the 6-node exponential chain.

Topology control is about selecting only a subset of the available communication links for data transmission, which has been widely used to construct networks having certain specific properties such as planarity, bounded node degree, the spanner property, or low interference [1,5,10–12,17]. Researchers are not only interested in minimizing the *average interference* on the nodes, but also the *maximum interference*, because the maximum interference determines when the first node will run out of energy, which could mean termination or partial termination of the network operation. Minimizing the maximum interference while maintaining connectivity is one of the most well-known open algorithmic problems in wireless sensor network optimization[2]. The problem is hard because it has an unusual combinatorial structure which is complicated, and intuitions do not seem to always apply.

Example 1. The even distribution of nodes which are linearly connected in Fig. 2(a) has an average interference of $\frac{10}{6}$ and the maximum interference of 2. However, for the linearly connected *exponential chain*, which means the node distances grow exponentially, the average interference is $\frac{16}{6}$ and the maximum is 4 although it is more sparse than the even distribution (Fig. 2(b)). Fig. 2(c) gives another connected topology for the 6-node exponential chain, where the node degrees are larger than those of the linear connection. However, its average interference is $\frac{14}{6}$ and the maximum is only 3.

Despite some significant efforts, known results are few and not all that satisfactory. The authors of paper [3] proved that it is NP-hard to compute the minimum maximum interference (MMI) while preserving connectivity in two-dimensional (2D) networks. The authors of paper [6] proposed an algorithm that could bound the maximum interference by $O(\sqrt{\Delta})$ using ideas from the $\varepsilon - net$ theory and computational geometry. Here, n is the number of nodes and Δ is the maximum node degree when each node is set to the maximum transmission radius and connected to all the other nodes in its range (as all the nodes have the same maximum transmission radius, the topology is actually a unit-disk-graph). In contrast, the problem of computing the minimum average interference (MAI) is structurally simpler. For minimizing average interference in 2D networks, the authors of paper [15] developed an asymptotically optimal algorithm with an approximation ratio of $O(logn)$. Given the lack of progress on the 2D version of the problem, researchers started investigating the one-dimensional (1D) networks (all nodes located on a line), but the problem did not become easier. For minimizing the maximum interference on the exponential chain, the authors of [20,21] proposed an asymptotically optimal algorithm and proved a tight lower bound of $\Omega(\sqrt{\Delta})$. Furthermore, for the general case in which the nodes are arbitrarily distributed on a line, the so called *highway model*, they bounded MMI by $O(\sqrt{\Delta})$ and presented an approximation with ratio $O(\sqrt[4]{\Delta})$.

In this paper, we study minimization of the average and the maximum interference for the highway model. We prove that there is always an optimal topology with minimum interference that is planar (i.e., its edges intersect only at their endpoints when drawing on a 2D plane). Two exact algorithms are proposed. The first one is an exact algorithm that can minimize the average interference in polynomial time–$O(n^3\Delta)$, where n is the number of nodes and Δ is the maximum node degree. The other one minimizes the maximum interference in sub-exponential time–$O(n^3\Delta^{O(k)})$, where $k = O(\sqrt{\Delta})$ is the minimum maximum interference. All the optimal topologies constructed are planar.

The rest of the paper is organized as follows. In Section 2, we give the formal definitions of the interference model and the problem. Section 3 describes the no-cross property and the algorithm to minimize the average interference for the highway model. Section 4 describes how to minimize the maximum interference. Section 5 gives some discussions. Section 6 concludes the paper and points out some open problems and possible future work.

---

[2] It is one of the five open algorithm problems in wireless sensor networking proposed in [13].
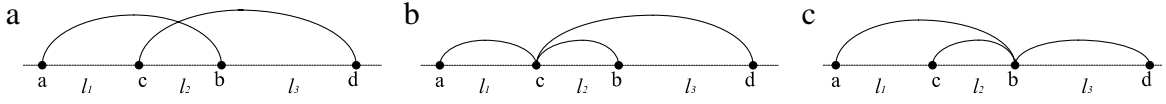
Fig. 3. a, b, c and d are four nodes distributed on a line, where $l_1 = c - a$, $l_2 = b - c$, and $l_3 = d - b$, and (a, b) and (c, d) are two edges: (a) (a, b) and (c, d) have a cross; (b) remove the cross when $l_1 \leq l_2 + l_3$; (c) remove the cross when $l_1 > l_2 + l_3$.

2. Models and problem definitions

We assume a wireless sensor network in which the nodes are stationary after being placed in a region. If at some point they need to be moved, we could re-run the proposed algorithms using the new coordinates. The maximum transmission radius of the nodes is denoted as r_{max}. Each node can self-adjust its transmission radius from 0 to r_{max}. We assume there are no obstacles to block the communications. Therefore, the maximum transmission range of a node v is a disk centered at v with radius r_{max}. For the highway model, we assume r_{max} is not shorter than the longest distance between two consecutive nodes, or else there is no connected topology. If r_{max} is set to or longer than the distance from the first to the last node on the line, it means any node can potentially directly connect to any other nodes.

The network is modeled as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of communication links. For the highway model, the n nodes in $V = \{v_0, v_1, \ldots, v_{n-1}\}$ are arbitrarily deployed along a line from left to right. We can regard the line as an x-axis, and $v_0 = 0$. Then, each node u is denoted as its x-coordinate. An edge $(u, v) \in E$ exists only if both their transmission radii, r_u and r_v, are not shorter than their Euclidean distance $|u - v|$. Therefore, in G, the transmission radius of a node is equal to the distance to its farthest neighbor (two nodes are neighbors means there is an edge incident on them). In addition, we introduce the following terms. 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 $\{v_s, v_{s+1}, \ldots, v_{t-1}, v_t\}$; the nodes *outside* $\overline{v_s v_t}$ are the other nodes not including the ones that are on it; the nodes *inside* $\overline{v_s v_t}$ are $\{v_{s+1}, \ldots, v_{t-1}\}$.

The receiver-centric interference model is adopted. The interference of a node v, denoted as $RI(v)$, is defined as the number of other nodes whose transmission ranges can cover v:

$$RI(v) = |\{u | u \in V/\{v\}, |u - v| \leq r_u\}|. \tag{1}$$

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

$$RI_{avg}(G) = \frac{\sum_{v \in V} RI(v)}{|V|}. \tag{2}$$

The maximum node interference, $RI_{max}(G)$, can be defined as:

$$RI_{max}(G) = \max_{v \in V} RI(v). \tag{3}$$

Besides minimizing interference, we also need to preserve the network connectivity. Therefore, the optimal topology with the minimum interference should be a spanning tree on V. Our problems can then be defined as:

Given n nodes arbitrarily distributed on a 1D line, construct a spanning tree, $G = (V, E)$, to connect all the nodes with edges no longer than r_{max}. The minimization of the average interference problem is to construct a spanning tree that minimizes $RI_{avg}(G)$, and the minimization of the maximum interference problem is to construct a spanning tree that minimizes $RI_{max}(G)$.

3. Minimizing the average interference

3.1. No-cross property

For a spanning tree $G = (V, E)$ constructed on the nodes along a line, we can draw all the edges on one side of the line. A *cross* means there are two edges that share at least a common point excluding their endpoints (Fig. 3(a)). By adding and deleting edges, we show below that a cross can be removed without increasing interference on any node while preserving the network connectivity.

Theorem 2 (*No-cross Property*). *For a 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. We prove this theorem by showing how to remove a cross. Without loss of generality, we handle the cross in Fig. 3(a). Note that there can be other nodes distributed at any other places on the line and the four nodes need not be consecutive. For the case $l_1 \leq l_2 + l_3$, we remove the cross by replacing the edge (a, b) with (a, c) and adding (c, b) (Fig. 3(b)). Firstly, we check whether the newly added edges, (a, c) and (c, b), are valid which means their lengths do not exceed r_{max}. Since $|a - c| = l_1 < l_1 + l_2 = |a - b|$ and (a, b) is valid, (a, c) is also valid. Similarly, (c, b) is also valid. Secondly, there are three nodes, a, b and c, whose edges are changed. We check whether the changes potentially would make them interfere with any new nodes. For a, one of its longer edges (a, b) is replaced with a shorter one (a, c), so a cannot interfere with more nodes in the new topology. A similar conclusion can be arrived at for b. As for node c, we add a new edge (a, c) of length

l_1 and (b, c) of length l_2. However, in both topologies, c already has an edge (c, d) of length $l_2 + l_3$. Since $l_2 + l_3 > l_2$ and $l_2 + l_3 \geq l_1$, the new edges will not cause c interfere with any new nodes. Therefore, the topology in Fig. 3(b) would not add to the interference on any nodes. Thirdly, since there are still paths to connect the nodes a, b and the nodes c, d, the new topology is connected as long as the topology in Fig. 3(a) is connected. Furthermore, since deleting an edge will not increase any interference, we can remove any cycles in the new topology by deleting edges to form a spanning tree. Therefore, for the case $l_1 \leq l_2 + l_3$, we can remove the cross to construct a new spanning tree without increasing interference on any nodes. Similarly, we can prove that the above is also true when $l_1 > l_2 + l_3$ as illustrated in Fig. 3(c), and the theorem is proved. □

The no-cross property is stronger than planarity. Therefore, there is always an optimal topology with minimum interference that is planar. Moreover, according to the no-cross property, if there is already an edge (v_s, v_t), all the nodes inside the segment $\overline{v_s v_t}$ can be only adjacent to nodes located on the segment, but not to any other nodes on the line. (Two nodes are adjacent means they are neighbors.) However, it does not mean that interference of the nodes inside the segment is independent of the nodes outside. The nodes inside $\overline{v_s v_t}$ can interfere with the ones outside, and vice versa. This gives us an important clue in designing algorithms to minimize the average or the maximum interference.

3.2. Algorithms to compute MAI

3.2.1. General ideas

Based on the no-cross property, in the optimal spanning tree with MAI, the nodes can be separated into segments. The nodes inside each segment are only adjacent to the other nodes on the same segment. However, as mentioned above, interference of the nodes inside a segment is still independent of those outside. Therefore, we do not compute the total interference by summing up the interference on each individual node, but the interference created by each node. Here, interference created by a node v with transmission radius r_v, $CI(v, r_v)$, is defined as the number of the other nodes covered by the transmission range of v:

$$CI(v, r_v) = |\{u | u \in V/\{v\}, |uv| \leq r_v\}|, \tag{4}$$

so that $\sum_{v \in V} CI(v, r_v) = \sum_{v \in V} RI(v)$. $CI(v, r_v)$ is only influenced by r_v, which is determined by the neighbors of v, and the locations of the other nodes. If all the nodes inside $\overline{v_s v_t}$ can only be adjacent to the nodes on it, the total interference created by the inside nodes will be independent of the topology of the outside nodes; and vice versa. Therefore, we can construct the optimal spanning tree based on dynamic programming, as follows.

3.2.2. Algorithms

To compute the optimal spanning tree, we need to determine (1) how to divide the line into segments and (2) how to connect the nodes on each segment. Two auxiliary functions are defined. The function $F(s, t)$,[3] where $s < t$, is to compute the topology on $\overline{v_s v_t}$ so that the total interference created by the nodes inside $\overline{v_s v_t}$ is minimized with the following conditions satisfied:

1. the transmission radius of v_s is r_{v_s};
2. the transmission radius of v_t is r_{v_t};
3. all the nodes inside $\overline{v_s v_t}$ can be only adjacent to the ones on the segment $\overline{v_s v_t}$;
4. each node inside $\overline{v_s v_t}$ has a path either to v_s or to v_t.

The function $G(s, t)$, where $s < t$, is to compute the topology on $\overline{v_s v_t}$ so that the total interference created by the nodes inside $\overline{v_s v_t}$ is minimized with the following conditions satisfied:
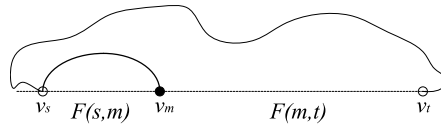
1. the transmission radius of v_s is r_{v_s};
2. the transmission radius of v_t is r_{v_t};
3. all the nodes inside $\overline{v_s v_t}$ can be only adjacent to the ones on the segment $\overline{v_s v_t}$;
4. all the nodes on $\overline{v_s v_t}$ are connected to each other directly or by nodes on $\overline{v_s v_t}$.

Both the functions F and G return the minimum total interference created by the nodes inside $\overline{v_s v_t}$. If $+\infty$ is returned, it means there is no such a topology to satisfy all the conditions. Comparing the fourth conditions, for function F, to achieve connectivity among all the nodes, we actually assume there is already a path from v_s to v_t before adding any edges to the nodes inside $\overline{v_s v_t}$. For G, there is no such a path.

For a node v, the set of its potential neighbors, $N(v)$, are the nodes covered by v's maximum transmission range:

$$N(v) = \{u | u \in V/\{v\}, |uv| \leq r_{max}\}. \tag{5}$$

---

[3] For conciseness, we use $F(s, t)$ and $recF(s, t)$ to stand for $F(v_s, v_t, s, t, r_{v_s}, r_{v_t})$ and $recF(v_s, v_t, s, t, r_{v_s}, r_{v_t})$ respectively, and $G(s, t)$ and $recG(s, t)$ to stand for $G(v_s, v_t, s, t, r_{v_s}, r_{v_t})$ and $recG(v_s, v_t, s, t, r_{v_s}, r_{v_t})$ respectively.

Fig. 4. There is already a path from v_s to v_t before adding any edges inside $\overline{v_s v_t}$, and v_m is adjacent to v_s when $v_p = v_s$. $F(s, t)$ is composed of three parts: $F(s, m)$, $F(m, t)$ and $CI(v_m, r_{v_m})$.

Recall that the transmission radius of v is the distance to its farthest neighbors. So, the set of its potential transmission radii, $R(v)$, is

$$R(v) = \{|uv||u \in N(v)\}, \tag{6}$$

and $|R(v)| \leq |N(v)| \leq \Delta$. If v can only be adjacent to a subset nodes S, its potential neighbors $N(v, S)$ and its potential transmission radii $R(v, S)$ are $N(v, S) = N(v) \bigcap S$ and $R(v, S) = \{|uv||u \in N(v, S)\}$ respectively.

To compute the functions F and G, we calculate and store each $CI(v, r_v)$ in an $n \times \Delta$ array. For $F(s, t)$, the boundary condition is there is no node inside $\overline{v_s v_t}$. For the other cases, to satisfy the condition 4), there must be at least one node v_m inside $\overline{v_s v_t}$ that is adjacent to v_p, where $v_p = v_s$ or v_t. Without loss of generality, we set $v_p = v_s$. Since v_s and v_m are connected as well as v_s and v_t in the assumption, there is already a path from v_m to v_t. Therefore, $F(s, t)$ consists of three parts, $F(s, m)$, $F(m, t)$ and $CI(v_m, r_{v_m})$ (Fig. 4). We can enumerate the node v_m and its transmission radius r_{v_m}, so that the function F can be computed in Algorithm 1:

- In lines 1–2, we first check the boundary condition.
- The set S is defined and store the nodes on $\overline{v_s v_t}$ in line 4.
- Lines 5–13 are to compute the minimum interference created by the nodes inside $\overline{v_s v_t}$ recursively with the four conditions satisfied.
- As v_m can only be adjacent to the nodes on $\overline{v_s v_t}$, its potential transmission radii are defined as $R(v_m, S)$ in line 6.
- In line 9 we assume adding an edge (v_m, v_p).
- Line 10 divides and computes $F(m, t)$ in three parts.
- For constructing the optimal spanning tree, we define the variable $recF(s, t)$ to record the necessary information for traceback in line 13.

Algorithm 1: Compute $F(s, t)$

```
1  if s + 1 = t then                                              /* the boundary condition */
2  |   return F = 0;
3  F ← +∞;
4  S ← {v_s, v_{s+1}, ..., v_t};
5  foreach v_m ∈ S/{v_s, v_t} do
6  |   R(v_m, S) = {|u − v_m||u ∈ N(v_m) ⋂ S};
7  |   foreach v_p ∈ {v_s, v_t} do
8  |   |   foreach r_{v_m} ∈ R(v_m, S) do
9  |   |   |   if |v_p − v_m| ≤ min(r_{v_m}, r_{v_p}) then          /* assume adding (v_m, v_p) */
10 |   |   |   |   tmp ← F(s, m) + F(m, t) + CI(v_m, r_{v_m});
11 |   |   |   |   if tmp < F then
12 |   |   |   |   |   F ← tmp;
13 |   |   |   |   |   recF(s, t) ← {v_m, v_p, r_{v_m}};

14 return F;
```

As for the function $G(s, t)$, in order to satisfy the condition (4), there are two alternative choices. One is that v_s is directly connected to v_t, such that $G(s, t) = F(s, t)$. The other is v_s and v_t are connected by some other nodes inside $\overline{v_s v_t}$. Then, there must be at least one node v_m inside $\overline{v_s v_t}$ which is adjacent to v_s, and $G(s, t)$ can consist of $F(s, m)$, $G(m, t)$, and $CI(v_m, r_{v_m})$. Similar to Algorithm 1, $G(s, t)$ can be computed in Algorithm 2. Some information for traceback is also recorded.

With F and G, MAI can be computed in Algorithm 3 by calling $G(0, n − 1)$. Here, for traceback, we also need to record r_{v_0} and $r_{v_{n-1}}$ in the optimal spanning tree.

When computing MAI, we record v_m and r_{v_m} for each function $G(s, t)$, and v_p, v_m and r_{v_m} for each function $F(s, t)$. Through tracing backwards, we can construct a connected topology of $n − 1$ edges with the minimum average interference, which is the optimal spanning tree. Algorithm 4 describes the process of traceback in linear time. All the edges of the optimal spanning tree are stored in the set *MinAvgTree*. The correctness of the above algorithms has been verified through comparing our results with the outputs generated by the brute-force search which runs slowly in the exponential time $O(n^\Delta)$. Fig. 5

Algorithm 2: Compute $G(s, t)$

---

1 $G \leftarrow +\infty$;
2 **if** $|v_s - v_t| \leq \min(r_{v_s}, r_{v_t})$ **then**   /* assume adding (v_s, v_t) */
3 $\quad$ $G \leftarrow F(v_s, v_t, s, t, r_{v_s}, r_{v_t})$;
4 $\quad$ $recG(s, t) \leftarrow \emptyset$;
5 $S \leftarrow \{v_s, v_{s+1}, \ldots, v_t\}$;
6 **foreach** $v_m \in S/\{v_s, v_t\}$ **do**
7 $\quad$ $R(v_m, S) = \{|u - v_m| | u \in N(v_m) \bigcap S\}$;
8 $\quad$ **foreach** $r_{v_m} \in R(v_m, S)$ **do**
9 $\quad\quad$ **if** $|v_s - v_m| \leq \min(r_{v_m}, r_{v_s})$ **then**   /* assume adding (v_s, v_m) */
10 $\quad\quad\quad$ $tmp \leftarrow F(s, m) + G(m, t) + CI(v_m, r_{v_m})$;
11 $\quad\quad\quad$ **if** $tmp < G$ **then**
12 $\quad\quad\quad\quad$ $G \leftarrow tmp$;
13 $\quad\quad\quad\quad$ $recG(s, t) \leftarrow \{v_m, r_{v_m}\}$;

14 **return** G;

Algorithm 3: Compute MAI in 1D networks

---

1 $total \leftarrow +\infty$;
2 **foreach** $r_{v_0} \in R(v_0)$ **do**
3 $\quad$ **foreach** $r_{v_{n-1}} \in R(v_{n-1})$ **do**
4 $\quad\quad$ $tmp = CI(v_0, r_{v_0}) + CI(v_{n-1}, r_{v_{n-1}}) + G(0, n - 1)$;
5 $\quad\quad$ **if** $tmp < total$ **then**
6 $\quad\quad\quad$ $total \leftarrow tmp$;
7 $\quad\quad\quad$ record r_{v_0} and $r_{v_{n-1}}$;

8 **return** $\frac{total}{n}$
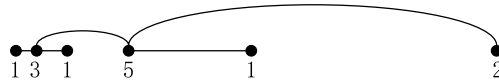
Fig. 5. The spanning tree for the 6-node exponential chain with MAI $\frac{13}{6}$: the numbers next to each node are the interference it creates.

gives an example of an optimal spanning tree for the 6-node exponential chain. In our algorithms, the main time spent is on computing functions F and G. Since the number of possible transmission radii of a node cannot exceed Δ, the time complexity to compute the optimal spanning tree with MAI is $O(n^3 \Delta^3)$.

3.3. Speeding up the computation

In this section, we exploit the properties of the function $F(s, t, r_{v_s}, r_{v_t})$[4] in order to speed up the computation of MAI. When calculating the optimal value of $F(s, t)$, we assume $recF(s, t, r_{v_s}.r_{v_t}) = \{v_m, v_p, r_{v_m}\}$, and denote m, r_{v_m} and p as $m(s, t, r_{v_s}, r_{v_t})$, $mr(s, t, r_{v_s}, r_{v_t})$ and $p(s, t, r_{v_s}, r_{v_t})$ respectively.

Theorem 3 (*Radius Properties*). *For any two nodes v_s, v_t and their possible transmission radii $\{r_{v_s}, s_1, s_2\} \subseteq R(v_s)$ and $\{r_{v_t}, t_1, t_2\} \subseteq R(v_t)$, we have*

1. *if $s_1 > s_2$, then $F(s, t, s_1, r_{v_t}) \leq F(s, t, s_2, r_{v_t})$;*
2. *if $t_1 > t_2$, then $F(s, t, r_{v_s}, t_1) \leq F(s, t, r_{v_s}, t_2)$;*
3. *if $r_{v_s} \geq |v_s - v_t|$, then $F(s, t, r_{v_s}, r_{v_t}) = F(s, t, |v_s - v_t|, r_{v_t})$;*
4. *if $r_{v_t} \geq |v_s - v_t|$, then $F(s, t, r_{v_s}, r_{v_t}) = F(s, t, r_{v_s}, |v_s - v_t|)$.*

Proof. We prove property (1) through mathematical induction:
Basis: when $s + 1 = t$, it holds as $F(s, t, s_1, r_{v_t}) = F(s, t, s_2, r_{v_t}) = 0$.
Inductive step: We assume the property holds for all the cases $t - s < i$ when $i \geq 2$. When $t - s = i$, we assume $recF(s, t, s_2, r_{v_t}) = \{v_m, v_p, r_{v_m}\}$ after $F(s, t, s_2, r_{v_t})$ is calculated. During computing $F(s, t, s_1, r_{v_t})$, we must also consider

---

[4] $F(s, t, a, b)$ and $recF(s, t, a, b)$ stand for $F(v_s, v_t, s, t, r_{v_s} = a, r_{v_t} = b)$ and $recF(v_s, v_t, s, t, r_{v_s} = a, r_{v_t} = b)$ respectively, which are the same as $F(s, t)$ and $recF(s, t)$ but explicitly include the parameters of the two transmission radii.

Algorithm 4: Construct the optimal spanning tree with MAI through traceback in 1D networks

---

1 Main Function

2 retrieve r_{v_0}, $r_{v_{n-1}}$, and $\frac{total}{n}$ set in the Algorithm 3;

3 if $\frac{total}{n} = +\infty$ **then**

4 $\quad\lfloor$ return $MinAvgTree \leftarrow \emptyset$

5 $TracebackG(v_0, v_{n-1}, 0, n-1, r_{v_0}, r_{v_{n-1}})$;

6 return $MinAvgTree$;

7 End Main Function

8 Function $TracebackF(v_s, v_t, s, t, r_{v_s}, r_{v_t})$

9 if $s + 1 = t$ **then**

10 $\quad\lfloor$ **return** ;

11 retrieve $\{v_m, v_p, r_{v_m}\}$ from $recF(m, t)$;

12 $MinAvgTree \leftarrow MinAvgTree \bigcup (v_m, v_p)$ $\qquad\qquad\qquad$ /* add an edge (v_m, v_p) */;

13 $TracebackF(v_s, v_m, s, m, r_{v_s}, r_{v_m})$;

14 $TracebackF(v_m, v_t, m, t, r_{v_m}, r_{v_t})$;

15 End Function $TracebackF(v_s, v_t, s, t, r_{v_s}, r_{v_t})$

16 Function $TracebackG(v_s, v_t, s, t, r_{v_s}, r_{v_t})$

17 if $recG(m, t) = \emptyset$ **then**

18 $\quad|\quad$ $MinAvgTree \leftarrow MinAvgTree \bigcup (v_s, v_t)$ $\qquad\qquad\qquad$ /* add an edge (v_s, v_t) */;

19 $\quad|\quad$ $TracebackF(v_s, v_t, s, t, r_{v_s}, r_{v_t})$;

20 else

21 $\quad|\quad$ retrieve $\{v_m, r_{v_m}\}$ from $recG(m, t)$;

22 $\quad|\quad$ $MinAvgTree \leftarrow MinAvgTree \bigcup (v_s, v_m)$ $\qquad\qquad\qquad$ /* add an edge (v_s, v_m) */;

23 $\quad|\quad$ $TracebackF(v_s, v_m, s, m, r_{v_s}, r_{v_m})$;

24 $\quad|\quad$ $TracebackG(v_m, v_t, m, t, r_{v_m}, r_{v_t})$;

25 End Function $TracebackG(v_s, v_t, s, t, r_{v_s}, r_{v_t})$

---

the case when the segment $\overline{v_s v_t}$ is divided into three parts by the node v_m with a radius of r_{v_m} from line 7 to line 10 in Algorithm 1. The check in line 7, as

$|v_p - v_m| = |v_s - v_m| \leq \min(s_2, r_{v_m}) \leq \min(s_1, r_{v_m})$ if $v_p = v_s$, and

$|v_p - v_m| = |v_t - v_m| \leq \min(r_{v_t}, r_{v_m})$ if $v_p = v_t$,

will return true. As for the calculation, we have

$$\begin{aligned} F(s, t, s_1, r_{v_t}) &\leq F(s, m, s_1, r_{v_m}) + F(m, t, r_{v_m}, r_{v_t}) + CI(v_m, r_{v_m}) \\ &\leq F(s, m, s_2, r_{v_m}) + F(m, t, r_{v_m}, r_{v_t}) + CI(v_m, r_{v_m}) \\ &= F(s, t, s_2, r_{v_t}). \end{aligned} \qquad (7)$$

Property (1) is proved.

Similarly, we can prove the other three properties through mathematical inductions. $\quad\square$

When computing $F(s, t, r_{v_s}, r_{v_t})$, there can be multiple optimal topologies. If there are some topologies in which the node v_m, which divides $F(s, t, r_{v_s}, r_{v_t})$ into three parts, is adjacent to v_s, we choose one of these topologies as the optimal one and define $p'(s, t, r_{v_s}, r_{v_t}) = s$; otherwise, we define $p'(s, t, r_{v_s}, r_{v_t}) = t$. Then, we find if v_m for $F(s, t, r_{v_s}, r_{v_t})$ is adjacent to the rightmost node v_t, then v'_m for $F(s, m, r_{v_s}, r_{v_m})$ will also be adjacent to the rightmost node v_m in its segment. We call it the *inertia property*.

Theorem 4 (*Inertia Property*). *For any s, t, $r_{v_s} \in R(v_s)$, $r_{v_t} \in R(v_t)$, if $p'(s, t, r_{v_s}, r_{v_t}) = t$, $m = m(s, t, r_{v_s}, r_{v_t})$, $r_{v_m} = mr(s, t, r_{v_s}, r_{v_t})$, and $s + 1 < m$, then*

$$p'(s, m, r_{v_s}, r_{v_m}) = m.$$

Proof. We denote $m(s, m, r_{v_s}, r_{v_m}) = m_1$ and $rm(s, m, r_{v_s}, r_{v_m}) = r_{v_{m_1}}$.

If $p'(s, m, r_{v_s}, r_{v_m}) = s$, we have

$$\begin{aligned} F(s, t, r_{v_s}, r_{v_t}) &= F(s, m, r_{v_s}, r_{v_m}) + F(m, t, r_{v_m}, r_{v_t}) + CI(v_m, r_{v_m}) \\ &= F(s, m_1, r_{v_s}, r_{v_{m_1}}) + F(m_1, m, r_{v_{m_1}}, r_{v_m}) + CI(v_{m_1}, r_{v_{m_1}}) + F(m, t, r_{v_m}, r_{v_t}) + CI(v_m, r_{v_m}). \end{aligned} \qquad (8)$$

As $F(m_1, t, r_{v_{m_1}}, r_{v_t})$ returns the minimal total interference, we can get

$$F(m_1, t, r_{v_{m_1}}, r_{v_t}) \leq F(m_1, m, r_{v_{m_1}}, r_{v_m}) + F(m, t, r_{v_m}, r_{v_t}) + CI(v_m, r_{v_m}). \qquad (9)$$

According to Eqs. (8) and (9), if setting $p'(s, t, r_{v_s}, r_{v_t}) = s$, we calculate

$$F(s, t, r_{v_s}, r_{v_t}) \leq F(s, m_1, r_{v_s}, r_{v_{m_1}}) + F(m_1, t, r_{v_{m_1}}, r_{v_t}) + CI(v_{m_1}, r_{v_{m_1}})$$
$$\leq F(s, m_1, r_{v_s}, r_{v_{m_1}}) + F(m_1, m, r_{v_{m_1}}, r_{v_m}) + F(m, t, r_{v_m}, r_{v_t}) + CI(v_m, r_{v_m}) + CI(v_{m_1}, r_{v_{m_1}}), \qquad (10)$$

which means we can also get the optimal value of $F(s, t, r_{v_s}, r_{v_t})$ in case that the node v_{m_1} is adjacent to v_s. Thus, if $p'(s, m, r_{v_s}, r_{v_m}) = s$, we have $p'(s, t, r_{v_s}, r_{v_t}) = s$, which is a contradiction with $p'(s, t, r_{v_s}, r_{v_t}) = t$. The inertia property is proved. $\square$

According to the inertia property, we can rewrite Algorithm 1 as Algorithm 5 to compute $F(s, t)$.

Algorithm 5: Compute $F(s, t)$ based on the inertia property

1 call $F_s(s, t, r_{v_s}, r_{v_t})$;

2 Function $F_s(s, t, r_{v_s}, r_{v_t})$ /* always set $v_p = v_s$ */
3 if $s + 1 = t$ **then** /* the boundary condition */
4   **return** $F_s \leftarrow 0$;
5 $F_s \leftarrow F_t(s, t, r_{v_s}, r_{v_t})$;
6 $S \leftarrow \{v_s, v_{s+1}, \ldots, v_t\}$;
7 foreach $v_m \in S/\{v_s, v_t\}$ **do**
8 $R(v_m, S) = \{|u - v_m| | u \in N(v_m) \bigcap S\}$;
9   **foreach** $r_{v_m} \in R(v_m, S)$ **do**
10   **if** $|v_s - v_m| \leq \min(r_{v_m}, r_{v_s})$ **then** /* assume adding (v_m, v_s) */
11 $tmp \leftarrow F_s(s, m) + F_s(m, t) + CI(v_m, r_{v_m})$;
12   **if** $tmp < F_s$ **then**
13 $F_s \leftarrow tmp$;

14 return F_s;

15 Function $F_t(s, t, r_{v_s}, r_{v_t})$ /* always set $v_p = v_t$ */
16 if $s + 1 = t$ **then** /* the boundary condition */
17   **return** $F_t = 0$;
18 $F_t \leftarrow +\infty$;
19 $S \leftarrow \{v_s, v_{s+1}, \ldots, v_t\}$;
20 foreach $v_m \in S/\{v_s, v_t\}$ **do**
21 $R(v_m, S) = \{|u - v_m| | u \in N(v_m) \bigcap S\}$;
22   **foreach** $r_{v_m} \in R(v_m, S)$ **do**
23   **if** $|v_t - v_m| \leq \min(r_{v_m}, r_{v_t})$ **then** /* assume adding (v_m, v_t) */
24 $tmp \leftarrow F_t(s, m) + F_t(m, t) + CI(v_m, r_{v_m})$;
25   **if** $tmp < F_t$ **then**
26 $F_t \leftarrow tmp$;

27 return F_t;

In the function $F_s(s, t, r_{v_s}, r_{v_t})$, after the check in line 10, we have $r_{v_m} \geq |v_s - v_m|$. According to the radius property, line 11 can be written as

$$tmp = F_s(s, m, r_{v_s}, |v_s - v_m|) + F_s(m, t) + CI(v_m, r_{v_m}). \qquad (11)$$

Now, the first element is independent from r_{v_m}. We take the two elements on the right as a whole, and we have

$$\min\{F_s(m, t) + CI(v_m, r_{v_m}) | r_{v_m} \geq |v_m - v_s| \quad \text{and} \quad r_{v_m} \in R(v_m)\}$$
$$= \min\{F_s(m, t, r'_{v_m}, r_{v_t}) + CI(v_m, r'_{v_m}), F_s(m, t) + CI(v_m, |v_m - v_s|)\},$$

where r'_{v_m} is the shortest radius in $R(v_m)$ that is longer than $|v_m - v_s|$. When computing $F_s(s, t)$ in decreasing order of r_{v_m}, we need not enumerate r_{v_m} any more. A similar conclusion applies for $F_t(s, t)$. Therefore, the time complexity for computing MAI is reduced to $O(n^3 \Delta^2)$.

Furthermore, for the functions $F_s(s, t)$ and $F_t(s, t)$, we have the following two conclusions.

Theorem 5. *For any s, t, when calling $F_s(s, t)$ in the process of computing MAI, we always have $r_{v_t} \geq |v_s - v_t|$.*
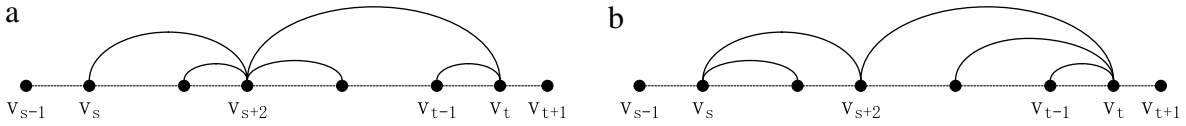
Fig. 6. The skeleton $c(v_s, v_t, 3) = \{(v_s, r_{v_s} = |v_s - v_{s+2}|), (v_{s+2}, r_{v_{s+2}} = |v_t - v_{s+2}|), (v_t, r_{v_t} = |v_t - v_{s+2}|)\}$ on the segment $\overline{v_s v_t}$. (a) and (b) are two possible topologies computed according to $c(v_s, v_t, 3)$.

Proof. The function F_s is called in lines 1 and 11 of Algorithm 5. The checks in line 8 of Algorithm 2 and in line 10 of Algorithm 5 guarantee that the theorem is correct. □

Thus, according to the radius property, we have $F_s(s, t) = F_s(s, t, r_{v_s}, |v_s - v_t|)$.

Theorem 6. *For any s, t, $F_t(s, t) = F_t(s, t, +\infty, r_{v_t})$.*

Proof. When computing $F_t(s, t)$ in Algorithm 5, r_{v_s} is not used. □

Based the above two theorems, the time complexity for computing $F_s(s, t, r_{v_s}, r_{v_t})$ and $F_t(s, t, r_{v_s}, r_{v_t})$ is reduced by Δ. Therefore, we can now compute MAI in time $O(n^3 \Delta)$.

4. Minimizing the maximum interference

4.1. Basic ideas

For the n nodes, $V = \{v_1, v_2, \ldots, v_{n-1}\}$, the minimum maximum node interference in all the possible spanning trees is denoted as k. We have $k \le \Delta \le n - 1$ because all the nodes have the same maximum transmission radius r_{max}. In this section, firstly we design an algorithm to check whether there is a spanning tree with the maximum interference no larger than a given k set from 1 to $n - 1$. After computing k, we can construct the optimal tree with such a maximum interference by traceback.

For a segment $\overline{v_s v_t}$, even when the nodes inside are not allowed to be adjacent to the ones outside, they still interfere with the outside nodes, and vice versa. We record all the interference from the nodes on $\overline{v_s v_t}$ to the outside nodes as a set $C(v_s, v_t, k)$, where $s \le k$. Each element $c(v_s, v_t, k) \in C(v_s, v_t, k)$, called *a skeleton* of the topologies on $\overline{v_s v_t}$, stores the following nodes and their transmission radii:

1. if $s > 0$ and $t < n - 1$: the nodes on $\overline{v_s v_t}$ that interfere with v_{s-1} or v_{t+1};
2. if $s = 0$ and $t < n - 1$: the nodes on $\overline{v_s v_t}$ that interfere with v_{t+1};
3. if $s > 0$ and $t = n - 1$: the nodes on $\overline{v_s v_t}$ that interfere with v_{s-1};
4. if $s = 0$ and $t = n - 1$: meaningless.

Specifically, $C(v, v, k)$ has $|R(v)|$ elements that store the node v and its potential transmission radii in $R(v)$. Since there must be no more than k nodes on $\overline{v_s v_t}$ that interfere with the left or the right nodes outside respectively, we call a skeleton $c(v_s, v_t, k)$ *valid* if and only if there are no more than k nodes in it that interfere with the first node that is on the left or right of $\overline{v_s v_t}$ respectively. Fig. 6 gives an example of a valid skeleton $c(v_s, v_t, 3)$ and two different topologies built according to the skeleton on $\overline{v_s v_t}$, where v_s and v_{s+2} interfere with v_{s-1}, and only v_t interferes with v_{t+1}. Note that a valid skeleton does not guarantee that the maximum interference in the whole topology would not exceed the maximum, such as $RI(v_{s+2}) = 4 > 3$ in Fig. 6(a).

Further, given $c(v_0, v_s, k), c(v_s, v_t, k)$ and $c(v_t, v_{n-1}, k)$, to compute the topology on $\overline{v_s v_t}$, the following two requirements need to be satisfied: (1) together with the interference from nodes in $c(v_s, v_t, k)$, each node outside $\overline{v_s v_t}$ cannot be interfered with more than k nodes; and (2), together with interference from nodes in $c(v_0, v_s, k)$ and $c(v_t, v_{n-1}, k)$, each node on $\overline{v_s v_t}$ cannot be interfered with more than k nodes. Considering the mutual interference among the nodes on or outside each segment, we can design an algorithm to check whether there is a spanning tree with maximum interference no greater than k by dynamic programming, as follows.

4.2. Algorithms to compute MMI

First of all, we define a function $Merge(c(v_{p_1}, v_{p_2}, k), c(v_{p_2+1}, v_{p_3}, k), \ldots, c(v_{p_{m-1}}, v_{p_m}, k))$, where $0 \le p_1 \le p_2 \le \cdots \le p_m \le n - 1$, to merge the skeletons on the consecutive segments and return $c(v_{p_1}, v_{p_m}, k)$. The method is to check every node in the skeletons whether to interfere with the first node left or right to $\overline{v_{p_1} v_{p_m}}$. Note that after merging, the new skeleton $c(v_{p_1}, v_{p_m}, k)$ may not be valid. Similar to computing the average interference, here we define two auxiliary boolean functions. The function *boolean* $F^*(s, t, k)$[5], where $s < t$, is to check whether there is a topology on $\overline{v_s v_t}$ that satisfies the following conditions simultaneously:

---

[5] For conciseness, we use $F^*(s, t, k)$ to stand for $F^*(v_s, v_t, s, t, r_{v_s}, r_{v_t}, c(v_0, v_s, k), c(v_{s+1}, v_{t-1}, k), c(v_t, v_{n-1}, k))$, and $G^*(s, t, k)$ to stand for $G^*(v_s, v_t, s, t, r_{v_s}, r_{v_t}, c(v_0, v_s, k), c(v_{s+1}, v_{t-1}, k), c(v_t, v_{n-1}, k))$.

1. the transmission radius of v_s is r_{v_s};
2. the transmission radius of v_t is r_{v_t};
3. all the nodes inside $\overline{v_s v_t}$ can be only adjacent to the ones on $\overline{v_s v_t}$;
4. the skeleton for $\overline{v_{s+1} v_{t-1}}$ is $c(v_{s+1}, v_{t-1}, k)$;
5. $RI(v) \leq k$, for each v inside $\overline{v_{s+1} v_{t-1}}$;
6. each node inside $\overline{v_s v_t}$ have a path either to v_s or to v_t.

Similarly, the function *boolean* $G^*(s, t, k)$, where $s < t$, is to check whether there is a topology on $\overline{v_s v_t}$ that satisfies the following conditions simultaneously:

1. the transmission radius of v_s is r_{v_s};
2. the transmission radius of v_t is r_{v_t};
3. all the nodes inside $\overline{v_s v_t}$ can be only adjacent to the ones on $\overline{v_s v_t}$;
4. the skeleton for $\overline{v_{s+1} v_{t-1}}$ is $c(v_{s+1}, v_{t-1}, k)$;
5. $RI(v) \leq k$, for each v inside $\overline{v_{s+1} v_{t-1}}$;
6. all the nodes on $\overline{v_s v_t}$ are connected to each other directly or by the nodes on $\overline{v_s v_t}$.

For $F^*(s, t, k)$, we still assume that there has been a path from v_s to v_t before adding any edges to the nodes inside $\overline{v_s v_t}$. When $s < t - 1$, to satisfy the condition 6), there must be a node v_m inside $\overline{v_s v_t}$ that is adjacent to v_p, where $v_p = v_s$ or v_t. Therefore, there is a path from v_s to v_m as well as from v_m to v_t. By ensuring $RI(v_m) \leq k$, $F^*(s, t, k)$ is divided to check $F^*(s, m, k)$ and $F^*(m, t, k)$. So it can be computed by Algorithm 6:

- Lines 1–2 is the boundary condition.
- Lines 4–14 are to compute $F^*(s, t, k)$ recursively.
- In line 8, we assume adding an edge (v_m, v_p).
- Lines 9 and 10 enumerate the possible skeletons on $\overline{v_{s+1} v_{m-1}}$ and $\overline{v_{m+1} v_{t-1}}$.
- Line 11 is to ensure the condition 4 is satisfied.
- In line 12,
  $c(v_0, v_m, k) = Merge(c(v_0, v_s, k), c(v_{s+1}, v_{m-1}, k), c(v_m, v_m, k))$,
  and $c(v_m, v_{n-1}, k) = Merge(c(v_m, v_m, k), c(v_{m+1}, v_{t-1}, k), c(v_t, v_{n-1}, k))$.
  Line 13 is to check their validity.
- All the three components for $F^*(s, m, k)$ are checked in line 13.

Actually, in line 9, we need not enumerate all the elements in $C(v_{s+1}, v_{m-1}, k)$ to set $c(v_{s+1}, v_{m-1}, k)$, because the interference of the nodes on segment $\overline{v_{s+1} v_{m-1}}$ with the node v_s has been defined in the input parameter $c(v_{s+1}, v_{t-1}, k)$. Therefore, we only need to enumerate the nodes on $\overline{v_{s+1} v_{m-1}}$ that interfere with v_m. Similarly, in line 10, we also only need to enumerate the nodes on $\overline{v_{m+1} v_{t-1}}$ that interfere with v_m as interference on v_t has been defined in $c(v_{s+1}, v_{t-1}, k)$.

Algorithm 6: Compute *boolean* $F^*(s, t, k)$

1 **if** $s + 1 = t$ **then**   /* the boundary condition */
2   **return** $F^* \leftarrow true$;
3 $S \leftarrow \{v_s, v_{s+1}, \ldots, v_t\}$;
4 **foreach** $v_m \in S / \{v_s, v_t\}$ **do**
5     $R(v_m, S) \leftarrow \{|u - v_m| \,|\, u \in N(v_m) \bigcap S\}$;
6   **foreach** $v_p \in \{v_s, v_t\}$ **do**
7   **foreach** $r_{v_m} \in R(v_m, S)$ **do**
8   **if** $|v_p - v_m| \leq \min(r_{v_m}, r_{v_p})$ **then**
9   **foreach** $c(v_{s+1}, v_{m-1}, k) \in C(v_{s+1}, v_{m-1}, k)$ **do**
10   **foreach** $c(v_{m+1}, v_{t-1}, k) \in C(v_{m+1}, v_{t-1}, k)$ **do**
11   **if** $Merge(c(v_{s+1}, v_{m-1}, k), c(v_m, v_m, k), c(v_{m+1}, v_{t-1}, k)) = c(v_{s+1}, v_{t-1}, k)$ **then**
12   compute $c(v_0, v_m, k)$ and $c(v_m, v_{n-1}, k)$ by merging;
13   **if** $c(v_0, v_m, k)$ is valid **and** $c(v_m, v_{n-1}, k)$ is valid **and** no more than k nodes that interfere with v_m **and** $F^*(s, m, k)$ **and** $F^*(m, t, k)$ **then**
14   **return** $F^* \leftarrow true$;

15 **return** $F^* \leftarrow false$ /* no topology on $\overline{v_s v_t}$ to satisfy the 6 conditions */;

To compute $G^*(s, t, k)$, we actually assume there is no path from v_s to v_t before adding edges to the nodes inside $\overline{v_s v_t}$. In order to satisfy the condition 6), there must be a node v_m inside $\overline{v_s v_t}$ that is adjacent to v_s. Similarly, by ensuring $RI(v_m) \leq k$,

---

Algorithm 7: Compute *boolean* $G^*(s, t, k)$

---

1 **if** $|v_s - v_t| \leq \min(r_{v_s}, r_{v_t})$ **and** $F^*(s, t, k) = true$ **then**   /* assume adding an edge (v_s, v_t) */
2   **return** $G^* \leftarrow true$;

3 $S \leftarrow \{v_s, v_{s+1}, \ldots, v_t\}$;
4 **foreach** $v_m \in S/\{v_s, v_t\}$ **do**
5 $R(v_m, S) \leftarrow \{|u - v_m| | u \in N(v_m) \bigcap S\}$;
6   **foreach** $r_{v_m} \in R(v_m, S)$ **do**
7   **if** $|v_s - v_m| \leq \min(r_{v_m}, r_{v_s})$ **then**
8   **foreach** $c(v_{s+1}, v_{m-1}, k) \in C(v_{s+1}, v_{m-1}, k)$ **do**
9   **foreach** $c(v_{m+1}, v_{t-1}, k) \in C(v_{m+1}, v_{t-1}, k)$ **do**
10   **if** $Merge(c(v_{s+1}, v_{m-1}, k), c(v_m, v_m, k), c(v_{m+1}, v_{t-1}, k)) = c(v_{s+1}, v_{t-1}, k)$ **then**
11   compute $c(v_0, v_m, k)$ and $c(v_m, v_{n-1}, k)$ by merging;
12   **if** $c(v_0, v_m, k)$ is valid **and** $c(v_m, v_{n-1}, k)$ is valid **and** no more than k nodes that interfere with v_m **and** $F^*(s, m, k)$ **and** $G^*(m, t, k)$ **then**
13   **return** $G^* \leftarrow true$;

14 **return** $G^* \leftarrow false$ /* no topology on $\overline{v_s v_t}$ to satisfy the 6 conditions */;
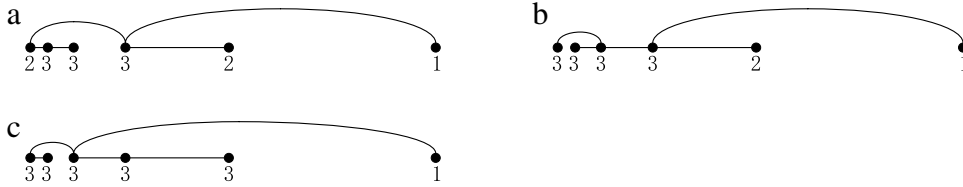

Fig. 7. The different optimal spanning trees for the 6-node exponential chain with MMI of 3. The numbers next to each node are the node interference.

$G^*(s, t, k)$ is divided to check $F^*(s, m, k)$ and $G^*(m, t, k)$. Algorithm 7 gives a detailed description on how to compute $G^*(s, t, k)$.

We design the main function *FindMinMax(V)* to find MMI k, which calls $F^*(s, t, k)$ and $G^*(s, t, k)$. From 1 to $n - 1$, we check and return k immediately when a spanning tree with the maximum interference of k is found. Algorithm 8 illustrates how to compute the function *FindMinMax(V)* by calling $G^*(0, n - 1, k)$, where we only consider the cases of $|V| = n > 2$.

Algorithm 8: *FindMinMax(V)*: Compute MMI in 1D networks

---

1 $k \leftarrow 1$;
2 **while** $k \leq n - 1$ **do**
3   **foreach** $r_{v_0} \in R(v_0)$ **do**
4   **foreach** $r_{v_{n-1}} \in R(v_{n-1})$ **do**
5   **foreach** $c(v_1, v_{n-2}, k) \in C(v_1, v_{n-2}, k)$ **do**
6   **if** no more than k nodes in $c(v_1, v_{n-2}, k)$ and v_{n-1} that interfere with v_0 **and** no more than k nodes in $c(v_1, v_{n-2}, k)$ and v_0 that interfere with v_{n-1} **and** $G^*(0, n - 1, k)$ **then**
7   **return** k;

8 $k = k + 1$;
9 **return** $+\infty$ /* no connected topology on V with the constraint of $r_{\max}$ */;

After computing *FindMinMax(V)*, we can do traceback and construct the optimal spanning tree with MMI by adding exactly $n - 1$ edges, similar to Algorithm 4. For brevity, we omit the traceback function here.

4.3. Analysis

4.3.1. Correctness

The method has been verified through comparing our optimal topologies with the outputs generated by the brute-force search whose running time is $O(n^{\Delta})$. Moreover, our algorithms can find all the optimal topologies of MMI without crosses. We illustrate three different optimal topologies for the 6-node exponential chain in Fig. 7.

4.3.2. Time complexity

Firstly, we analyze the size of the set $C(v_s, v_t, k)$ when $s > 0$ and $t < n - 1$. The nodes in each element $c(v_s, v_t, k)$ can be divided into the node sets CL and CR which contain the nodes on $\overline{v_s v_t}$ that interfere with v_{s-1} and v_{t+1} respectively. As the maximum interference is k, we get $|CL| \leq k \leq \Delta$. Each node has the maximum different transmission radii of Δ. The number of combinations for the nodes in CL and their transmission radii is

$$\binom{\Delta}{0} + \binom{\Delta}{1} \times \Delta + \cdots + \binom{\Delta}{k} \times \Delta^k = O(\Delta^{2k}). \tag{12}$$

A similar result can be obtained for CR. Therefore, the size of $C(v_s, v_t, k)$ is $O(\Delta^{4k})$ when $s > 0$ and $t < n - 1$. For the parameters of the function $F^*(s, t, k)$, the number of different values of v_s or v_t is n, and r_{v_s} or r_{v_t} has Δ different choices at the most. Furthermore, the size of $c(v_{s+1}, v_{t-1}, k)$ is $O(\Delta^{4k})$, while the size of $c(v_0, v_s, k)$ or $c(v_t, v_{n-1}, k)$ is $O(\Delta^{2k})$ since $\overline{v_0 v_s}$ only have nodes on the right side and $\overline{v_t v_{n-1}}$ only have nodes on the left side. Therefore, the total of the functions $F^*(s, t, k)$ is $O(n^2 \Delta^{8k+2})$. Also, $G^*(s, t, k)$ has the same number of variations. For each function of $F^*(s, t, k)$ or $G^*(s, t, k)$, the computing time is $O(n\Delta^{4k+1})$. As there are no functions being repeatedly computed, the time to finish $FindMinMax(V)$ will be $O(n^3 \Delta^{12k+3})$. To construct the optimal spanning tree, the main time cost is for computing k by $FindMinMax(V)$. Thus, the time complexity to construct the spanning tree with the minimum maximum interference is $O(n^3 \Delta^{O(k)})$. Since $\Delta \leq n - 1$ and $k = O(\sqrt{\Delta})$ have been proved in paper [20], the time is sub-exponential. However, when Δ is small, which means a low maximum node degree, our algorithm is fast.

Space complexity: The space is mainly for storing the functions $F^*(s, t, k)$ and $G^*(s, t, k)$ as well as the sets $C(v_s, v_t, k)$. Therefore, the space complexity is $O(n^2 \Delta^{O(k)})$.

5. Discussions

In this paper we study situations where the nodes are arbitrarily distributed along a line. One node can interfere with other nodes even if they are not neighbors. An edge (u, v) exists only if the transmission ranges of both nodes are not shorter than their distance $|uv|$. Adding an edge (u, v) may not affect u, as the transmission range of u is not shorter than (u, v), but may cause v to interfere with more nodes. Or (u, v) may not affect both u and v at all. All these variations make the minimization of interference hard. Whether it is NP-hard to minimize the maximum interference for the highway model is an open question.

All the nodes have the same maximum transmission range r_{max} in our model. In practice, we may use various types of nodes and select a suitable maximum transmission range for each node according to its remaining energy in order to prolong the network's lifetime. Note that when deleting a cross, we do not add an edge to v that exceeds in length the longest edge the node already has in the previous topology. So the no-cross property is still true when each node v has an individual maximum transmission range r_{max}^v. After re-defining the potential neighbors $N(v)$ by replacing r_{max} with r_{max}^v in Eq. (5), our algorithms can still construct the optimal spanning trees with the minimum average or the minimum maximum interference.

Planarity is also an important property of the network. Many efficient routing protocols for wireless networks require the topology to be planar [8,9]. Besides having guaranteed low interference and connectivity, the optimal topologies constructed by our methods are planar. There are however other desirable network properties, such as low node degree. Our algorithms can generate multiple optimal topologies with minimum interference. One example is the topology with MMI for the exponential chains. For the 6-node chain, we can find all the 17 optimal topologies without a cross. For the 8-node, the total number of optimal topologies without a cross can be as many as 241. Therefore, we can try to choose an optimal topology that has the other properties as well as low-interference, planarity, and connectivity.

6. Conclusion

In this paper, we study the problem of minimizing the receiver-centric interference for the highway model. Based on the no-cross property and dynamic programming, a polynomial-time exact algorithm is proposed which can construct a connected topology with minimum average interference. Furthermore, making use of the radius property and the inertia property, we propose a way to substantially speed up the computation. A sub-exponential-time exact algorithm is also presented to construct the connected topology while minimizing the maximum interference. The optimal topologies constructed have the properties of low interference, connectivity, and planarity simultaneously. The problem of whether it is NP-hard to minimize the maximum interference for the highway model is still open. Related open problems include how to extend the exact algorithms to 2D networks, how to design efficient approximations to minimize the maximum interference in 2D networks, how to tackle the interference minimization given other network properties, such as small node degree and low spanner.

Acknowledgements

References

[1] R. Banner, A. Orda, Multi-objective topology control in wireless networks, in: Proc. 27th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, 2008, pp. 448–456.
[2] M. Benkert, J. Gudmundsson, H. Haverkort, A. Wolff, Constructing minimum-interference networks, Computational Geometry 40 (3) (2008) 179–194.
[3] K. Buchin, Minimizing the maximum interference is hard, arXiv:0802.2134v1, 2008.
[4] M. Burkhart, P. von Rickenbach, R. Wattenhofer, A. Zollinger, Does topology control reduce interference? in: 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc, 2004, pp. 9–19.
[5] M. Damian, N. Javali, Distributed construction of bounded-degree low-interference spanners of low weight, In: 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc, 2008, pp. 101–110.
[6] M. Halldórsson, T. Tokuyama, Minimizing interference of a wireless ad hoc network in a plane, in: Proc. 2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks, ALGOSENSORS, in: LNCS, vol. 4240, 2006, pp. 71–82.
[7] T. Johansson, L. Carr-Motyčková, Reducing interference in ad hoc networks through topology control, in: 2nd ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing, DIALM-POMC, 2005, pp. 17–23.
[8] B. Karp, H. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: 6th Annual International Conference on Mobile Computing and Networking, MobiCom, 2000, pp. 243–254.
[9] F. Kuhn, R. Wattenhofer, A. Zollinger, Asymptotically optimal geometric mobile ad-hoc routing, In: 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, DIALM, 2002, pp. 24–33.
[10] N. Li, C.-J. Hou, L. Sha, Design and Analysis of an MST-Based Topology Control Algorithm, IEEE Transactions on Wireless Communications 4 (3) (2003) 1195–1206.
[11] X.-Y. Li, G. Calinescu, P.-J. Wan, Distributed construction of planar spanner and routing for ad hoc wireless networks, in: Proc. 21th Annual Joint Conference of the IEEE Computer and Communications Societies, vol.3, INFOCOM, 2002, pp. 1268–1277.
[12] X.Y. Li, W.Z. Song, W. Wang, A unified energy efficient topology for unicast and broadcast, In: 11th Annual International Conference on Mobile Computing and Networking, MOBICOM, 2005, pp. 1–15.
[13] T. Locher, P. von Rickenbach, R. Wattenhofer, Sensor networks continue to puzzle: Selected open problems, In: 8th International Conference on Distributed Computing and Networking, ICDCN, 2008, pp. 25–38.
[14] K. Moaveni-Nejad, X.-Y. Li, Low-interference topology control for wireless ad hoc networks, International Journal of Ad Hoc & Sensor Wireless Networks 1 (2005) 41–64.
[15] T. Moscibroda, R. Wattenhofer, Minimizing interference in ad hoc and sensor networks, in: 2nd ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing, DIALM-POMC, 2005, pp. 24–33.
[16] T. Nguyen, N. Lam, D. Huynh, J. Bolla, Minimum edge interference in wireless sensor networks, in: 5th International Conference in Wireless Algorithms, Systems, and Applications, WASA, in: LNCS, vol. 6221, 2010, pp. 57–67.
[17] P. Santi, Topology Control in Wireless Ad Hoc and Sensor Networks, Wiley, 2005.
[18] H. Tan, T. Lou, F.C.M. Lau, Y. Wang, S. Chen, Minimizing interference for the highway model in wireless ad-hoc and sensor networks, in: Proc. 37th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM, LNCS, vol. 6543, 2011, pp. 520–532 (in press).
[19] K.-D. Wu, W. Liao, On constructing low interference topology in multihop wireless networks, International Journal of Sensor Networks 2 (2007) 321–330.
[20] P. von Rickenbach, S. Schmid, R. Wattenhofer, A. Zollinger, A robust interference model for wireless ad hoc networks, in: 5th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks, WMAN, 2005.
[21] P. von Rickenbach, R. Wattenhofer, A. Zollinger, Algorithmic models of interference in wireless ad hoc and sensor networks, IEEE/ACM Transactions on Networking (TON) 17 (1) (2009) 172–185.