

Interference Minimization on 1D

Atishaya Maharjan, Stephane Durocher

March 18, 2025

Abstract

This paper delves deep into a FPT interference model first proposed by von Rickenbach et.al.

1 Introduction and Preliminaries

An *ad-hoc* network is a type of decentralized network that deals with mostly wireless devices with various components. Rickenbach, et.al [4] proposed a framework of using points for devices and creating a connected graph using those points if the radii of transmission of the devices were adequate for data transfer for indicating interference.

We consider all the nodes in a network to be able to transmit and route packets to each other. In order for that to be possible, the resulting communication graph must be connected. In addition, we want to minimize the interference between the various nodes of the network. To model the transmission distance and interference received per node, we adopt the framework given by Khabbazzian, Majid and Durocher, et. al [3]. Their framework models the transmission of a node v as the radius of transmission $r(v)$ such that every node inside the radius of transmission receives interference of 1 from v . This also means that for any two nodes u and v in the resulting communication graph, they are connected by an edge if and only if $\text{dist}(u, v) \leq r(u)$ and $\text{dist}(v, u) \leq r(v)$. It is worth noting that this framework heavily adopts the model presented by von Rickenbach *et al.* [4] which is related to the *geometric radio network* model of Dessmark and Pelc [2].

Definition 1.1 (Radius of transmission). *For a node v , the radius of transmission is maximum transmission emitted radially from v . It is denoted by $r(v)$.*

Definition 1.2 (Connection Link). *Let u and v be two distinct nodes. If both u and v are inside each other's radius of transmission, then we say that there is a connection link between them. Mathematically, if*

$$\text{dist}(u, v) \leq r(u) \text{ and } \text{dist}(v, u) \leq r(v)$$

then there is a connection link between u and v .

Definition 1.3 (Communication Graph). *A communication graph $G = (V, E)$ is a connected graph where V is the set of vertices representing the wireless nodes in space and E is the set of edges representing a connection link between any two vertices $u, v \in V$.*

The problem statement now is this:

Conjecture 1.4 (Interference in 1D is NP-hard). *We conjecture that solving the interference problem in 1D is NP-hard in general cases.*

Conjecture 1.5 (There is a better approximation bound for Interference). *We conjecture that there exists a better approximation bound for the interference problem than $O(\sqrt[4]{n})$.*

2 Fixed Parameter Tractable (FPT) Problem Instances

We know from Buchin [1], that the interference problem in 2D is NP-Complete. We also know that the reduction applied to proof of the 2D case is not applicable to the 1D case because of

As such, it might be better for our intuition to build up several Fixed Parameter Tractable problem instances. The parameter that we will consider will be the distance between the nodes of the connected graph. Let the set of the distance parameters be D .

2.1 The distance set is exponential in nature

Let $D = \{d_1, d_2, \dots, d_m\}$ for some $m \in \mathbb{N}$. Then consider the following condition:

$$\forall d_i \in D, d_{i+1} = c \cdot d_i \text{ for some constant } c \in \mathcal{O}(1)$$

We then ask the following decision problem:

k -INTERFERENCE MINIMIZATION IN 1D

Input: Set of points P , distance set D , interference bound k

Question: Is there a connected graph $G = (V, E)$ such that the interference of the graph is less than or equal to k ?

Is there a connected graph $G = (V, E)$ such that the interference of the graph is less than or equal to k ?

We give the following algorithm to solve the problem instance:

Input: Set of points $P = \{p_1, p_2, \dots, p_n\}$, distance set $D = \{d_1, d_2, \dots, d_m\}$, interference bound k

Output: Graph T minimizing interference $\leq k$ or False

Initialize $S \leftarrow \{p_1\}$, $T \leftarrow \emptyset$, $I(r) \leftarrow 0$ for all $r \in P$ and $D' \leftarrow \{d_1, d_2 \in D : d_1 \pm d_2\}$;

Function BuildGraph(S, T, I):

```

    if  $S = P$  then
        if  $\max_{r \in P} I(r) \leq k$  then
            return  $T$  // Valid solution found
        end
        else
            return False
        end
    end
    Select an unconnected point  $p \in P \setminus S$ 
    foreach connected point  $q \in S$  such that  $\text{dist}(p, q) \in D'$  do
         $d \leftarrow |p - q|$ 
         $A(e) \leftarrow \{r \in P \mid r \in U_d(p) \cup U_d(q)\}$ 
        foreach  $r \in P$  do
            if  $r \in A(e)$  then
                 $I'(r) \leftarrow I(r) + 1$ 
            else
                 $I'(r) \leftarrow I(r)$ 
            end
        end
        end
        if  $\max_{r \in P} I'(r) > k$  then
            Continue // Prune search tree
        end
         $S' \leftarrow S \cup \{p\}$ 
         $T' \leftarrow T \cup \{(q, p)\}$ 
        // Recursive call to continue searching
        BuildGraph( $S', T', I'$ )
    end
    return False // No valid connections found

```

2.2 Analysis of the Algorithm

NOTE: This does not work for the worst case. Instead, all we can claim is the following:

Let $D' = \{d_1, d_2 \in D : d_1 \pm d_2\}$. Note that $0 \in D'$. Then, the algorithm runs in $O(|D'|^k n^c)$ for $c \in O(1)$ time. Note that $|D'| = 2^{\binom{|D|}{2}}$. The relation comes from the fact that we are choosing two distances from D and adding or subtracting them.

Apply the same algorithm but using D' instead of D .

You will have to split it into 2 cases. The first case is when you have your distance set being $D = \{0, c^i\}$ for $i = \{0, 1, 2, \dots\}$ and some constant c . The second case is when you don't have your distance set being $D = \{0, c^i\}$ for $i = \{0, 1, 2, \dots\}$ and some constant c , i.e you can't say anything about the structure. In that case, we claim that the maximum interference is just $O\left(\left\lceil \frac{\max D}{\min D} \right\rceil\right)$.

For correctness, all I need to prove is considering both cases, if there exists a solution with the given constraints of the decision problem, then the algorithm will find it.

Lemma 2.1. *For any point $p \in P$, all other points $q \in P \setminus \{p\}$ lie at distance $d \in D'$ from p .*

Lemma 2.2. *Correctness of Algorithm 1 with respect to the decision problem 2.1.*

Proof.

For correctness of the algorithm, we need to show that if there exists a feasible solution to the decision problem, then the algorithm also will find it. In addition, the algorithm will produce the optimal solution; if it exists. That is, the algorithm will not miss any feasible solutions.

Note that if we have a distance set, $D = \{0, c^i\}$ for $i = \{0, 1, 2, \dots\}$ for some constant c , then we argue that we consider all the possible options for the distances. Consider the first node p_1 . We claim that we consider all edge connections with the rest of the points of P (barring the last one). Or rather, the argument is that D' considers all possible nodes privy to the distance set D (barring the last one from the first one). However, we know that there are always better ways of joining the points than joining the first and the last point.

Invoke Lemma 2.1 to show that all points are reachable from each other. This shows that we consider all possible connections between the points, although it is a brute force solution. However, we still note that the total time complexity does NOT depend on n , i.e. does NOT depend on the number of points. \square

Lemma 2.3. *Running time of Algorithm 1 with respect to the decision problem 2.1.*

Proof.

For the case where D is a set of exponential distances, then the running time will just be a straightforward branching argument. This will yield a total running time of $\left(2^{\binom{|D|}{2}}\right)^k \cdot n^c$ where $c \in O(1)$.

Assuming that we have proved the optimal interference for D not being a set of exponential distances is $O\left(\left\lceil \frac{\max D}{\min D} \right\rceil\right)$, then the running time of the algorithm will just be $O(n)$ since it'll just be connecting the nodes in a line. \square

Theorem 2.4. *Algorithm 1 solves the decision problem 2.1 in $O\left(2^{\binom{|D|}{2}} \cdot n^c\right)$ time where $c \in O(1)$.*

Proof.

\square

2.3 The distance set is not exponential in nature

Let $D = \{d_1, d_2, \dots, d_m\}$ for some $m \in \mathbb{N}$. Then consider the following condition:

$$\forall d_i \in D, d_{i+1} \neq c \cdot d_i \text{ for any constant } c \in \mathcal{O}(1)$$

3 Open questions and Future work

These are our open questions now that we have solved the problem:

1. Is this problem NP-hard in 1D?
2. Is there a better approximation bound for the interference problem in 1D?

References

- [1] Kevin Buchin. Minimizing the maximum interference is hard, 2011.
- [2] Anders Dessmark and Andrzej Pelc. Tradeoffs between knowledge and time of communication in geometric radio networks. In *ACM Symposium on Parallel Algorithms and Architectures: Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 59–66, New York, NY, USA, 2001. ACM.
- [3] Majid Khabbazzian, Stephane Durocher, Alireza Haghnegahdar, and Fabian Kuhn. Bounding interference in wireless ad hoc networks with nodes in random position. *IEEE/ACM Transactions on Networking*, 23(4):1078–1091, 2015.
- [4] P. von Rickenbach, S. Schmid, R. Wattenhofer, and A. Zollinger. A robust interference model for wireless ad-hoc networks. In *19th IEEE International Parallel and Distributed Processing Symposium*, pages 8 pp.–, 2005.