A General Approach for Minimizing the Maximum Interference of a Wireless Ad-Hoc Network in Plane

V. Haghighatdoost, M. Espandar

Vahid Haghighatdoost, Maryam Espandar
Computer and Electrical Engineering Department,
Shahed University, opposite Holy shrine of Imam Khomeini,
Khalij Fars Expressway, Tehran, Iran.
{haghighatdoost, espandar}@shahed.ac.ir

Abstract: The interference reduction is one of the most important problems in the field of wireless sensor networks. Wireless sensor network elements are small mobile receiver and transmitters. The energy of processor and other components of each device is supplied by a small battery with restricted energy. One of the meanings that play an important role in energy consumption is the interference of signals. The interference of messages through a wireless network, results in message failing and transmitter should resend its message, thus the interference directly affect on the energy consumption of transmitter. This paper presents an algorithm which suggests the best subgraph for the input distribution of the nodes in the plane how the maximum interference of the proposed graph has the minimum value. The input of the application is the complete network graph, which means we know the cost of each link in the network graph. Without any lose of generality the Euclidean distance could be used as the weight of each link. The links are arranged and ranked according to their weights, in an iterative process the link which imposition minimum increase on the network interference with some extra conditions which is proposed in future sections, is added to resulting topology and is eliminated from list until all nodes are connected together. Experimental results show the efficiency of proposed algorithm not only for one dimensional known distribution like exponential node chain, but also for two dimensional distributions like two Exponential node chains and α-Spiral node chains.

Keywords: wireless ad-hoc network, sensor network, interference, spanning tree.

1 Introduction

Wireless ad-hoc networks consist of mobile nodes equipped with, among other components, a processor, some memory, a wireless radio, and a power source. Due to physical constraints, nodes are primarily powered by a weak battery, so energy is a scarce resource in wireless ad-hoc networks. In a general way, topology control can be considered as the task of, given a network communication graph, constructing a subgraph with certain desired properties while minimizing energy consumption. The subgraph needs to meet some requirements, the minimum requirement being to maintain connectivity and it should be a spanner of all nodes in original graph; Additionally, symmetric links are desired as they permit simpler higher-layer protocols [1]. One of the foremost approaches to achieve substantial energy conservation is by minimizing interference between network nodes. The concept of topology control restricts interference by reducing the transmission power levels at the network nodes and cutting off long-range connections in a coordinated way. At the same time transmission power reduction has to proceed in such a way
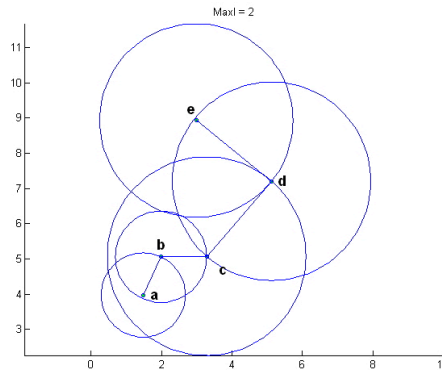
Figure 1: The interference model of a graph with 5 vertexes

that the resulting topology preserves connectivity. Some other works focused on topology control algorithms emphasizing locality while exhibiting more and more desirable properties [2–4], sometimes presenting distributed algorithms that optimize various design goals concurrently. All these approaches have in common, however, that they address interference reduction only implicitly. The intuition was that a low minimizing the maximum degree of nodes of graph would solve the interference issue automatically. As depicted in [1] this intuition was proved wrong in [5], starting a new thread that explicitly studies interference reduction in the context of topology control [6–8]. The general interference model introduced in [9], proposes a natural way to define interference in ad-hoc networks. The general question is: How can one connect the nodes such that as few nodes as possible disturb each other? In the following, we discuss the network and interference model presented in [9].

So far, not many results have been published in the context of explicit interference minimization. For networks restricted to one dimension the authors in [9] present a $\sqrt[4]{n}$-approximation of the optimal connectivity preserving topology that minimizes the maximum interference. For the two dimensional case, the authors in [10] propose an algorithm that bounds the maximum interference to $O(\sqrt{n})$. If average interference of a graph is considered, there is an asymptotically optimal algorithm achieving an approximation ratio of $O(logn)$ [11]. Kevin Buchin in [12] proved that this problem is NP-complete. In the following sections the proposed algorithm is explained briefly. The time complexity of producing the spanning tree with minimum interference is $O(n^5)$.

2   Interference Model of Network

The network is modeled as a geometric graph $G = (V, E)$. As mentioned in previous section the links between nodes are symmetric and have not direction, it means that a message sent over a link can be acknowledged by sending a corresponding message over the same link in the opposite direction. So the matrix E is symmetric.

Let N_u denote the set of all neighbors of a node $u \in V$ in the resulting topology. Then, each node u features a value r_u defined as the distance from u to its farthest neighbor. More precisely $r_u = max_{v \in N_u}\{|u - v|\}$, where $|u - v|$ denotes the Euclidean distance between nodes u and v. Since we assume the nodes to use omnidirectional antennas, $D(u, r_u)$ denotes the disk centered at u with radius r_u covering all nodes that are possibly affected by message transmission of u to one of its neighbors. Then the interference of a node v is defined as the number of other nodes that potentially affect message reception at node v.

Definition 1. Given a graph $G = (V, E)$, the interference of a node $v \in V$ is defined as:

$$I(v) = \left| \left\{ u | u \in V \diagdown \{v\}, v \in D(u, r_u) \right\} \right| \tag{1}$$
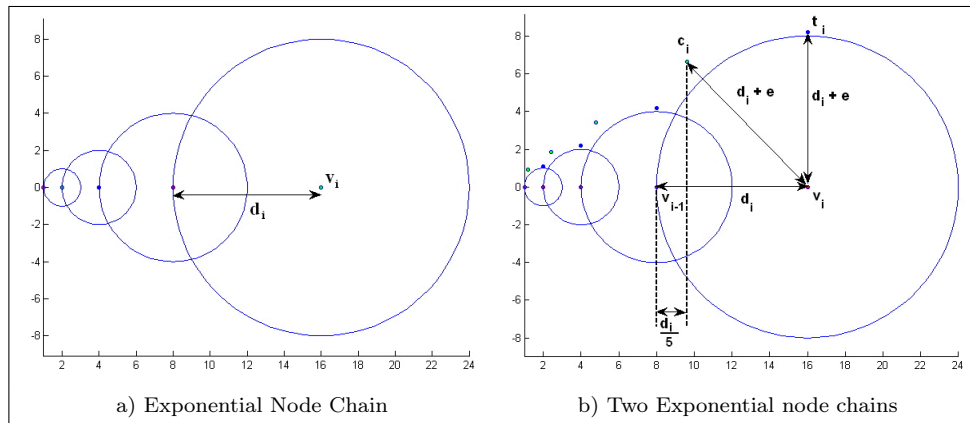
Figure 2: Two Special node distributions

Note that even though each node is also covered by its own disk, we do not consider this kind of self-interference. The graph interference is the maximum interference occurring in a graph:

Definition 2. The interference of a graph $G = (V, E)$ is defined as:

$$I(G) = max_{v \in V} I(v) \tag{2}$$

As shown in Figure 1 the interference of nodes is as follow:

Node:	a	b	c	d	e
Interference:	2	2	2	2	1

According to Definition 2 the Interference of graph $I(G)$ is 2.

3   The Nearest Neighbor Forest

In the first view of the interference problem, one may say the nearest neighbor forest or minimum spanning tree is the best subgraph which results in minimum interference. In this section, it is shown that this is already a substantial mistake, as thus interference becomes asymptotically incomparable with the interference-minimal topology. For Some special distribution the nearest neighbor forest results in the worst interference. Authors in [13], introduced an instance which seems to yield inherently high interference: the so called *exponential node chain* is a one-dimensional graph $G = (V, E)$ where the distance between two consecutive nodes grows exponentially from left to right as depicted in Figure 2(a).

That is, the distance between nodes v_i and v_{i+1} is 2^i for $i = 0, 1, 2, ..., n-1$. So as shown in Figure 3(a) the nearest neighbor forest results in the interference of $\Omega(n)$. Also in [9] they introduced the Two Exponential node chains as shown in Figure 2(b), on the bottom, there is a horizontal chain of nodes v_i with exponentially growing distances, the same as the one dimensional exponential chain, thus distance between v_i and v_{i+1} is 2^i. Each of these nodes v_i has a corresponding node t_i vertically displaced by a little more than v_i's distance to its left neighbor, that is, $|v_i - t_i| > d_i$ where $d_i = |v_i - v_{i-1}| = 2^{i-1}$. Note that the nodes t_i also form a (diagonal) exponential node chain. Finally, between two of these diagonal nodes t_{i-1} and t_i an additional helper node c_i is placed such that $|v_i - c_i| \geq |v_i - t_i|$. The Nearest Neighbor Forest for this node distribution is shown in Figure 3(b). The nearest neighbor forest for the distributions in Figure 2 and their disk graph is shown in Figure 3.

The algorithm proposed in [9] finds a subgraph for the exponential node chain (Figure 2(a)) with $I(G) \in O(\sqrt{n})$. And they proved bellow theorem:
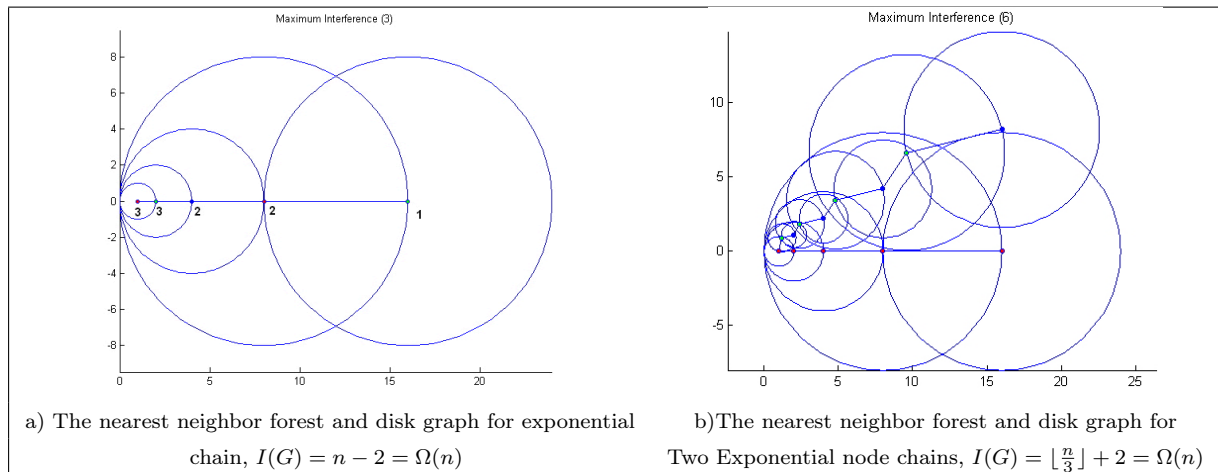
a) The nearest neighbor forest and disk graph for exponential chain, $I(G) = n - 2 = \Omega(n)$

b)The nearest neighbor forest and disk graph for Two Exponential node chains, $I(G) = \lfloor \frac{n}{3} \rfloor + 2 = \Omega(n)$

Figure 3: The disk graph and the interference of nearest neighbor forest

a) The best spanning tree for the Two Exponential node chains where I(G) always is equal to 3

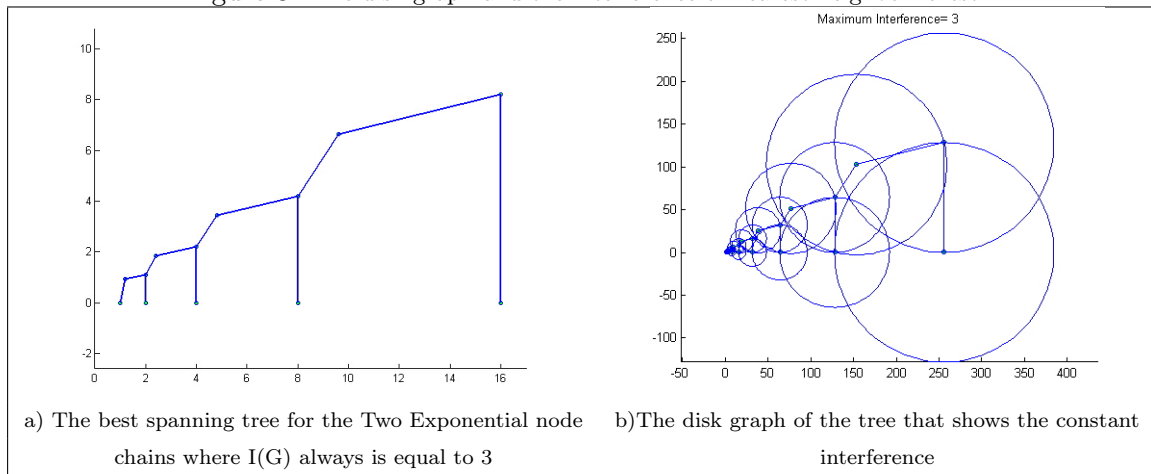b)The disk graph of the tree that shows the constant interference

Figure 4: Suggested topology for Two Exponential node chains in [9]

Theorem 1: Given an exponential node chain $G = (V, E)$ with $n = |V|$, $\sqrt{n}$ is a lower bound for the interference $I(G)$.

They also proposed a topology with constant interference for the Two Exponential node chains which is depicted in Figure 4 but there is no algorithmic method which generate automatically similar subgraph.

According to the construction of the exponential node chain, only nodes connecting to at least one node to their right increase the v_1's (leftmost node) interference. They called such a node a hub and define it as follows [9]:

Definition 3. Given a connected topology for the exponential node chain $G = (V, E)$. A node $v_i \in V$ is defined to be a hub in G iff there exists an edge (v_i, v_j) with $i < j$.

The A_{exp} algorithm which is proposed in [9] for exponential chain is as follows:

The algorithm starts with a graph $G_{exp} = (V, E_{exp})$, where V is the set of nodes in the exponential node chain and E_{exp} is initially the empty set. Following the scan-line principle, A_{exp} processes all nodes in the order of their occurrence from left to right. Initially, the leftmost node is set to be the current hub h. Then, for each node v_i, A_{exp} inserts an edge $\{h, v_i\}$ into E_{exp}. This is repeated until $I(G_{exp})$ increases due to the addition of such an edge. Now node v_i becomes the current hub and subsequent nodes are connected to v_i as long as $I(G_{exp})$ the overall interference does not increase. Figure 5 shows the resulting topology when A_{exp} algorithm is used for the exponential node chain with 17 nodes and I(G)=6.
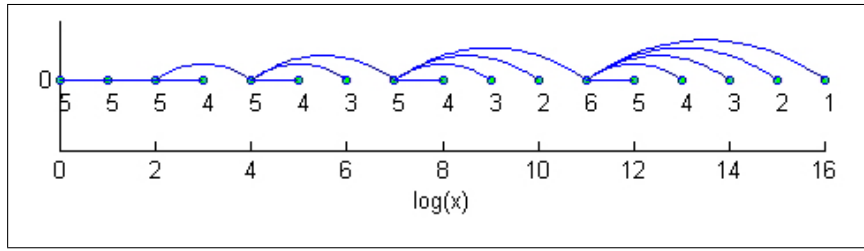
Figure 5: The result topology of A_{exp} algorithm for exponential node chain. The interference of the exponential node is bounded by $O(\sqrt{n})$. For clarity of representation edges are depicted as arcs and x dimension is shown in logarithmic scale. The interference of each node is wrote under the node position

In the next section a new algorithm which is the extension of A_{exp} is proposed for the nodes in the plane. This algorithm finds the best sub graph with minimum interference. The output of proposed algorithm for exponential chain is very similar to A_{exp}'s output with equal complexity. Also it shows the subgraph with constant interference for Two Exponential node chains.

4 Proposed Algorithm (A_{pln})

The following algorithm A_{pln} is the extension of A_{exp} for finding a spanning tree with minimum interference of nodes distributed in the plane. The A_{pln} the same as A_{exp} is an iterative algorithm which generates the resulting graph in n steps. Where n is the number of nodes. In the beginning the result graph has no edge and in each step of algorithm one edge is added to result graph until the connected result graph is generated.

Suppose $G_{in} = (V_{in}, E_{in})$ be the input graph, where $V_{in} = \{v_1, v_2, ..., v_n\}$ is the set of n separate nodes in the plane. (x_i, y_i) is the coordinate of v_i. Also E_{in} is the $n \times n$ adjacent matrix of G_{in} with e_{ij}^{in} elements. e_{ij}^{in} is the Euclidean distance between i'th and j'th nodes.

$$e_{ij}^{in} = |v_i - v_j| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad \forall i, j = 1, 2, ..., n \qquad (3)$$

The steps of generating the spanning tree with minimum interference from G_{in} are as follows:
Step 1) Preparing data: Generate the adjacent list of G_{in}'s edges. The adjacent list of edges of a graph is a list of triples (i, j, e_{ij}); where i points to i'th vertex and j points to j'th vertex of graph and e_{ij}) is the weight of edge (in our problem it is the Euclidean distance). G_{in} is a complete graph and the adjacent matrix E_{in} is symmetric so the adjacent list would have $\frac{n(n-1)}{2}$ elements.

Step 2) Preparing data: Sort the elements of adjacent list according to the weight of each edge and call the new list $SrtE^1$. Thus the elements of the $SrtE$ are arranged as follow:

$$EdgeLength(SrtE_i) \leq EdgeLength(SrtE_{i+1}) \qquad \forall i = 1, 2, ..., \frac{n(n-1)}{2} \qquad (4)$$

Step 3) Find Start Point: Find the smallest edge from $SrtE$ list; Set $E_{min} = SrtE_1$.
Step 4) Initialization: Suppose that $G_{pln} = (V_{pln}, E_{pln})$ is the result graph. Set $V_{pln} = V_{in}$ and E_{pln} contains of only the smallest edge $E_{min} = (h, k, e_{hk})$, in other word all elements of E_{pln} are valued with zero instead of e_{hk}^{pln} and e_{kh}^{pln}. Initialize the maximum interference of result

---
[1]SrtE: obtained from Sorted Edge list

graph $MaxI_{pln}$ with:

$$\begin{cases} a) & V_{pln} = V_{in} \\ b) & e_{ij}^{pln} = 0 \qquad \forall i,j = 1,2,...,n \\ c) & e_{hk}^{pln} = e_{kh}^{pln} = e_{hk}^{in} \\ d) & MaxI_{pln} = 1 \end{cases} \qquad (5)$$

Step 5) Initialize the Active Vertexes List: The nodes which are in sub connected graph are named as **Active Vertexes**. At the beginning sub connected graph consist of only the smallest edge. The Active Vertex list AV is a $1 \times n$ array; iff the node v_i be an Active Vertex the i'th element of $AV(AV_i)$ gets value 1 otherwise its value is 0.

$$AV = \begin{cases} 1 & \text{if } i = h,k \\ 0 & \text{if } otherwise \end{cases} \qquad \forall i = 1,2,...,n. \qquad (6)$$

In other word:

	1	2	...	k	...	h	...	N
$AV=$	0	0	0	1	0	1	0	0

(7)

Where k,h are the corresponding nodes of smallest edge.

Step 6) Check the Status: while all nodes of G_{pln} are not connected together repeat steps 7 to 9.

Step 7) Generate Candidate Edges: According to AV list, generate the Active Edge List (AE). AE is a subset of $SrtE$; An edge from $SrtE$ exist in AE iff only one of its endpoints have value 1 in AV.

$$\text{For Each } SrtE_k \in SrtE \text{ if } (AV_i = 1 \oplus AV_j = 1) \quad \rightarrow \quad \text{Add } SrtE_k \text{ to } AE$$
$$Where SrtE_k = (i,j,e_{ij}) \qquad (8)$$

The xor symbol ($\oplus$) in the predicate part of relation (8) ensures the intolerance of multi selecting one edge and recursion in final subgraph. According to (4) and (8) we can write:

$$EdgeLength(AE_i) \leq EdgeLength(AE_{i+1}) \quad \forall i = 1,2,...,size(AE)-1 \qquad (9)$$

Step 8) Find The best Edge: Select the edge $AE_m = (p,q,e_{pq}^{in})$ from AE, which leads to minimum increase on $MaxI_{pln}$ when is added to G_{pln}. After adding the AE_m to G_{pln}, update the AV and $MaxI_r$. The algorithm of finding the AE_m will be explained in next section. The specification of AE_m is as follow:

$$\exists AE_m \in AE | \forall AE_j \in AE \rightarrow I(G_{pln}, E_{pln} \cup AE_m) \leq I(G_{pln}, E_{pln} \cup AE_j) \qquad (10)$$

$I(G,E)$ determines the maximum interference of graph G according to adjacent matrix E. updating the variables is done as follow:

$$\begin{align} (a) & \quad Suppose \quad AE_m = (p,q,e_{pq}^{in}) \\ (b) & \quad E_{pln} = E_{pln} \cup AE_m \quad \underrightarrow{means} \quad e_{pq}^{pln} = e_{pq}^{in}, e_{qp}^{pln} = e_{qp}^{in} \\ (c) & \quad MaxI_{pln} = I(G_{pln}, E_{pln}) \\ (d) & \quad AV_p = 1, AV_q = 1 \end{align} \qquad (11)$$

Go to step 6 for checking the status.

Step 9) Finalizing: the obtained $G_{pln} = (V_{pln}, E_{pln})$ is the spanning tree with minimum interference for the input distribution of nodes V_{pln} in the plane.
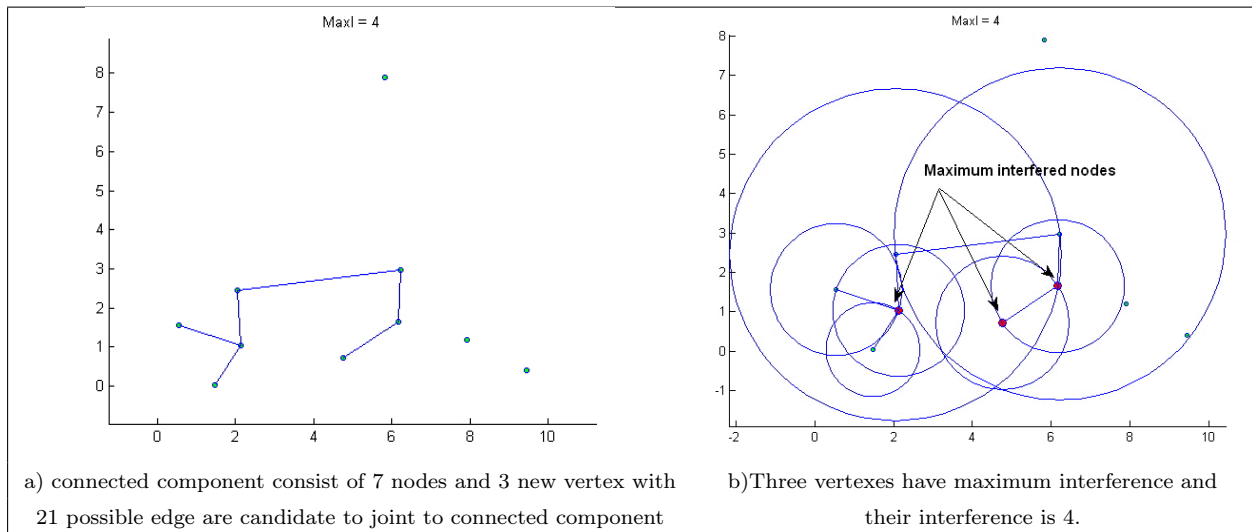
Step 10) Finish

a) connected component consist of 7 nodes and 3 new vertex with 21 possible edge are candidate to joint to connected component

b)Three vertexes have maximum interference and their interference is 4.

Figure 6: The topology of graph after 6 step with I(G)=4

5   Find best Edge Algorithm

In Step 8 of the A_{pln} algorithm we have addressed an algorithm which has found the best edge from AE list. The brief introduction of finding the best edge from AE list is as follow:
Suppose that the current state of algorithm is a connected sub graph with m vertex and $(n-m)$ disconnect nodes which are shown in Figure 6(a) we want to expand the connected sub graph with minimum possible increase of $MaxI$.
Lemma 1) Adding a new vertex to connected sub graph in the worst case result in two unit increase of the current Maximum interference.
Proof) suppose R_i determines the distance from V_i to its farthest neighbour and D_i is a disk with radius R_i and centered by V_i. The disk D_i shows the domain of i'th transmitter and the nodes inside the D_i affect by V_i. When a new vertex V_k is added to the sub graph by connecting to V_j, the new disk D_k is added to current disks and the radius of D_j may increase. Thus in the worst case D_j and D_k will dominate all other vertexes. So the Current max-interfered node is affected by two new transmitter signals and for this reason we will have two unit increase of $MaxI$. And in the best case $MaxI$ not changes.

$$\text{Adding a new Edge in the worst case}   \rightarrow   MaxI_{new} = MaxI_{old} + 2 \tag{12}$$

Figure 6 shows the increase of interference when a new vertex is added to sub graph.
 The algorithm of finding the best edge is as follows:
Step 1) as lemma 1 determined, In the worst case we will have 2 unit increase on Maximum interference of the graph so repeat the bellow Steps for $\triangle I = 0, 1, 2$
Step 2) For Each AE_i from AE list repeat bellow
Step 3) Set $MaxI_{new} = I(G_{pln}, E_{pln} \cup AE_i)$
Step 4) if $MaxI_{new}$ is equal to $(MaxI_{npln} + \triangle I)$ determine the AE_i as the best edge and go to Step 6, else check for next Edge.
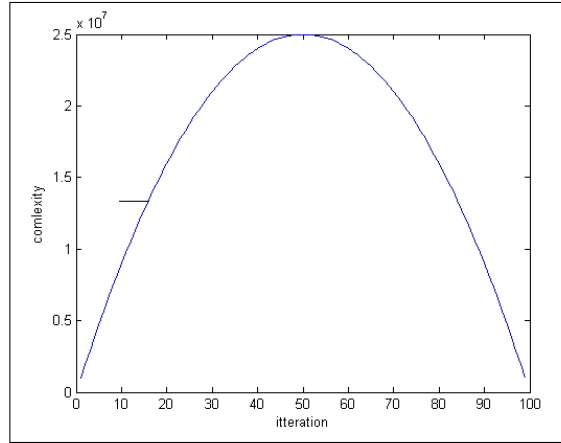**Step 5)** Set $\triangle I = \triangle I + 1$ and go to Step 2.
Step 6) Finish.

Figure 7: Relation of iteration and complexity of find the best edge for n=100.

The complexity of determining the best edge is as follow:

$$
\begin{aligned}
Complexity \quad &= O(3 * length(AE) * O(I(G,E))) \\
&= O(3 * m(n-m) * n^2) \\
&= O(mn^3 - m^2n^2)
\end{aligned}
\tag{13}
$$

Where m is number of active vertexes and n is the number of total vertexes. The relation (13) shows that in the first iteration ($m = 1$) the complexity of finding the best edge is $O(n3)$ and in the final Step ($m = n - 1$) it is also $O(n3)$ but in the middle iteration when $m = n/2$ the complexity is as follow:

$$
If(m = n/2) \rightarrow O(mn^3 - m^2n^2) = O(\frac{n^4}{2} - \frac{n^4}{4}) = O(\frac{n^4}{4}) = O(n^4)
\tag{14}
$$

The relation between complexity of determination of best edge and iteration is depicted in Figure 7.

Applying the Best Edge algorithm for the node distribution which was shown in Figure 6 is depicted in Figure 8.

6   Experimental Results

At first the proposed algorithm A_{pln} is compare with A_{exp}. As depicted in Figure 9 the interference of the exponential chain for both algorithms are equal but the topologies are different.

For A_{exp} the order of nodes is important but the A_{pln} does not have any precondition on the distribution. In following the strength of A_{pln} for finding the spanning tree with minimum interference for Two Exponential node chains and some special similar distributions is depicted. Figure 10 depicts the resulting topology if A_{pln} is applied to the Two Exponential node chains.The most important note on this topology is that the A_{pln} algorithm does not know this distribution but the resulting topology is the same as best topology which is created by human brain in [9]. For random distribution of nodes in plane the A_{pln} algorithm always suggest a better topology with equal or smaller interference value rather than nearest neighbor forest. Another test case is α-Spiral Node Chain. In α-Spiral node chain which is shown in Figure 12, the k'th node is placed in $(2^k \cos(ak), 2^k \sin(ak))$.
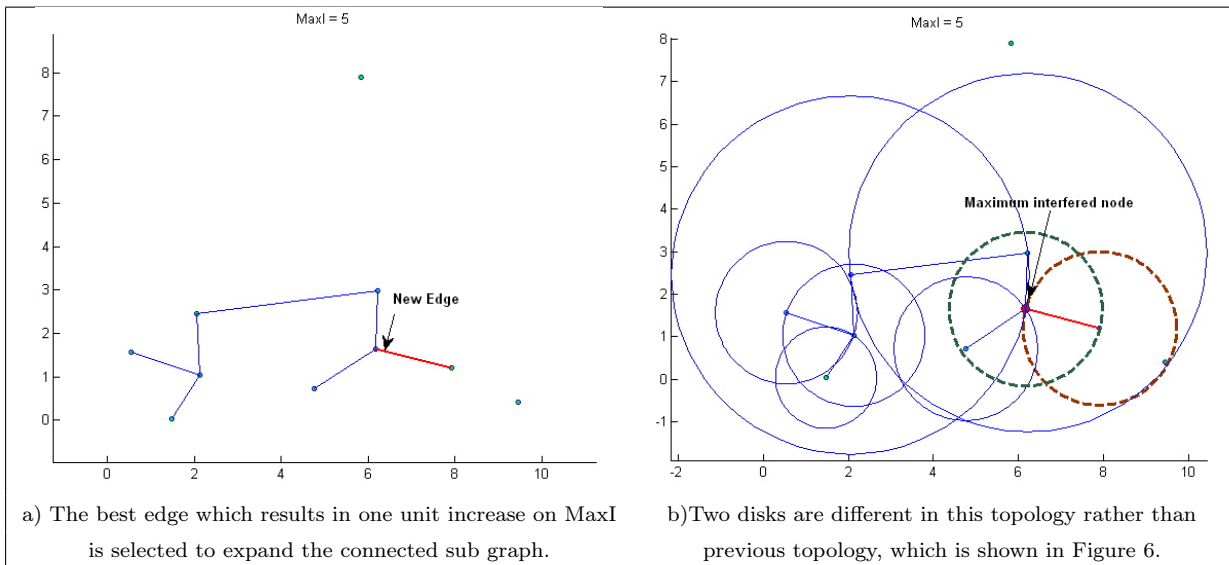
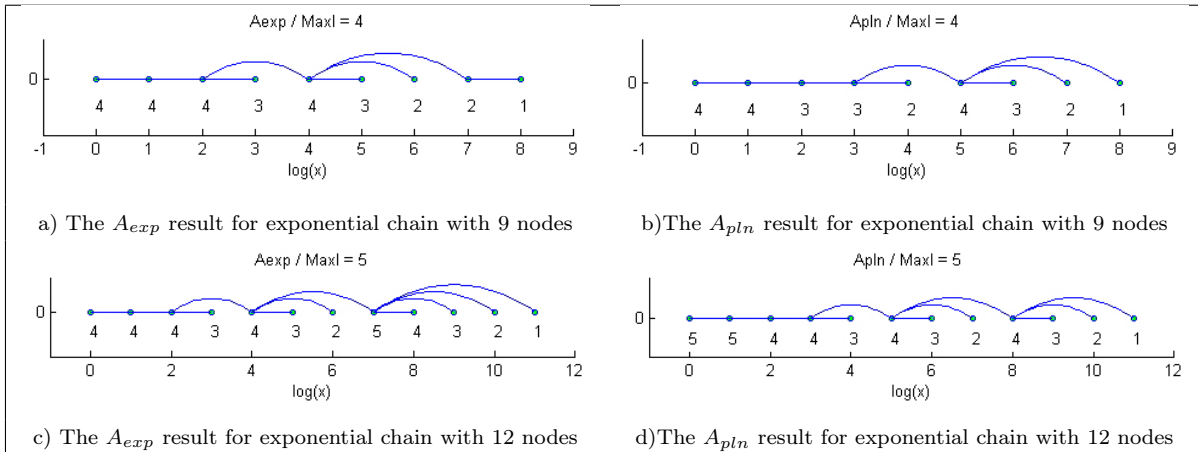Figure 8: The topology of graph after 7 step with I(G)=5

Figure 9: The spanning trees obtained by A_{exp}, A_{pln}, algorithms for exponential chain. For clarity of representation edges are depicted as arcs and x dimension is shown in logarithmic scale. The interference of each node is wrote under the node position

The results of applying the A_{pln} and nearest neighbor forest (A_{MST}) to some α-Spiral Node Chains are depicted in Figure 13.

7   Conclusion

In this paper a general algorithm named A_{pln} for finding the spanning tree of separate nodes in the plane has been proposed. The A_{pln} algorithm presents an iterative routine for minimizing the maximum interference of the resulting spanning tree.

At the beginning the resulting tree has only one edge which is the smallest edge in the input graph, until all input nodes are not connected together, the algorithm adds a new edge to the resulting tree.

For adding a new edge to sub graph the best edge which imposes minimum increase on the interference of all nodes from all available edges is selected. In section 6 the experimental result of using A_{pln} and A_{exp} for some special distributions are depicted and all of them show the good performance of the proposed algorithm. The A_{pln} is a general algorithm for any two dimensional
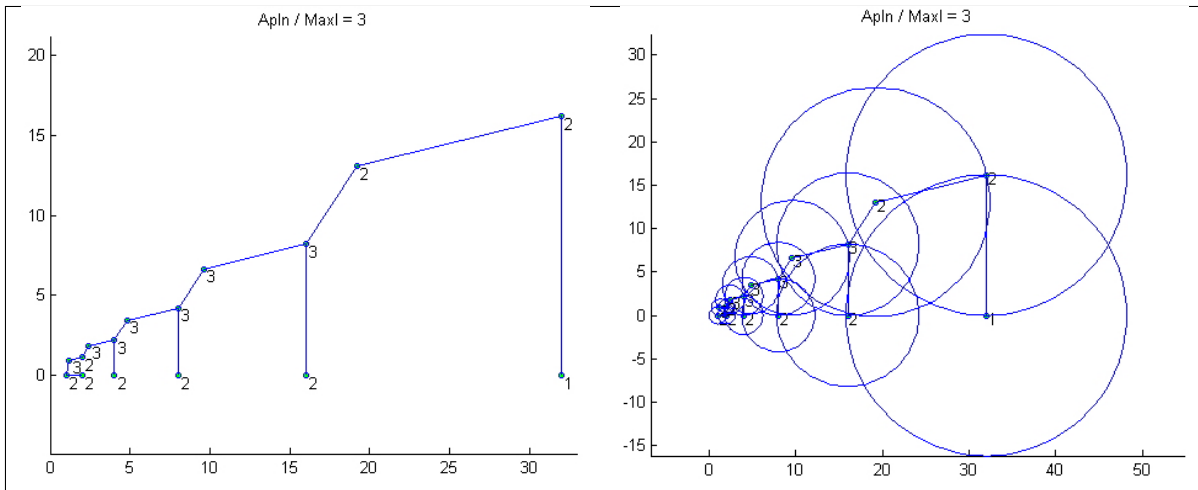
Figure 10: Constant Interference for Two Exponential node chains obtained by A_{pln} algorithm. The interference of each node is wrote beside the node position

a) Result of A_{MST} for random distribution of 20 nodes.

b)Result of A_{pln} for random distribution of 20 nodes

c) Result of A_{MST} for random distribution of 15 nodes

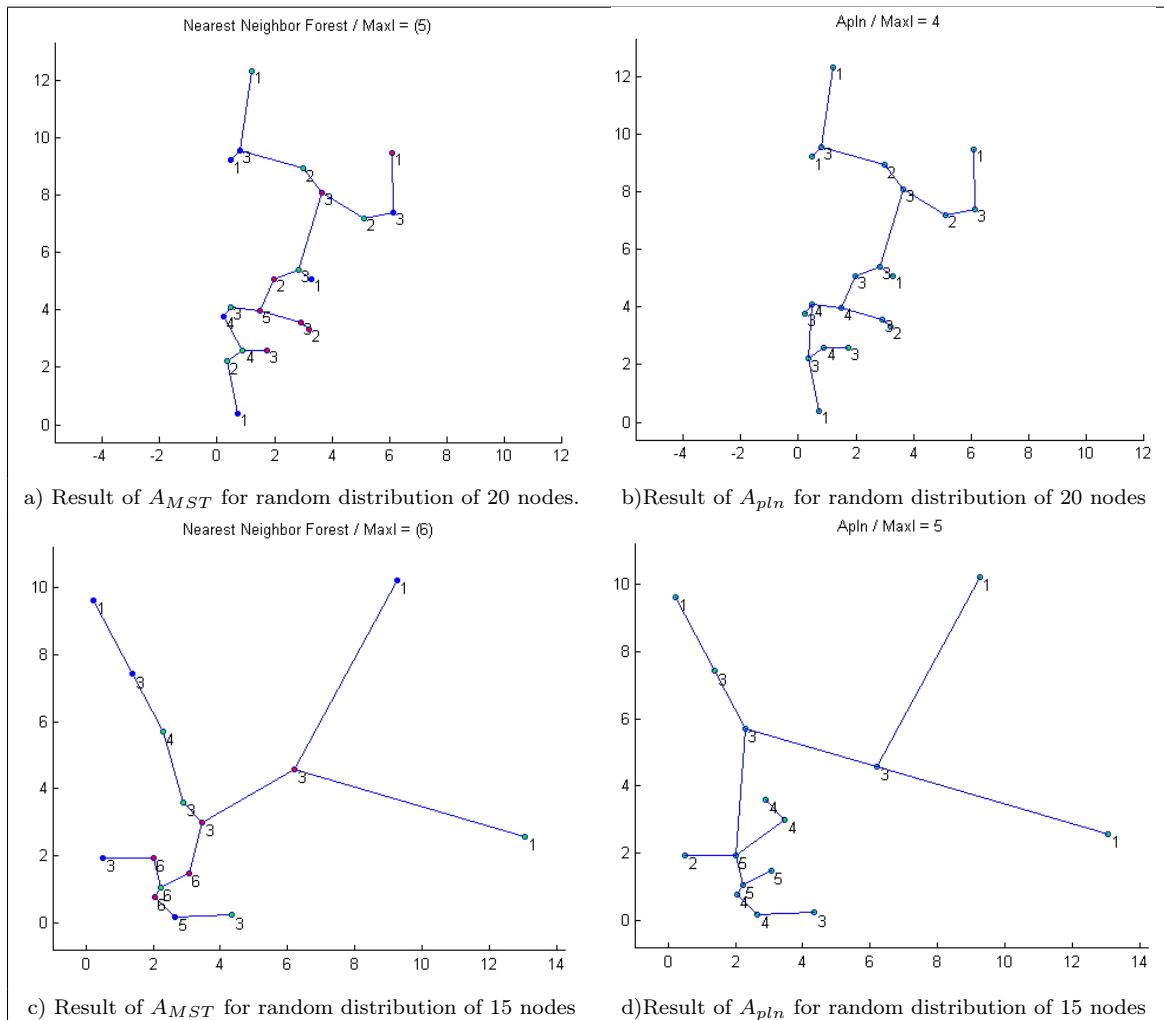d)Result of A_{pln} for random distribution of 15 nodes

Figure 11: Result of applying nearest neighbor forest and A_{pln} algorithms on random distribution of nodes.

distribution and it has no limit or special conditions for the input distribution and there is no need to inform it about the input distribution, the algorithm itself finds the best spanning
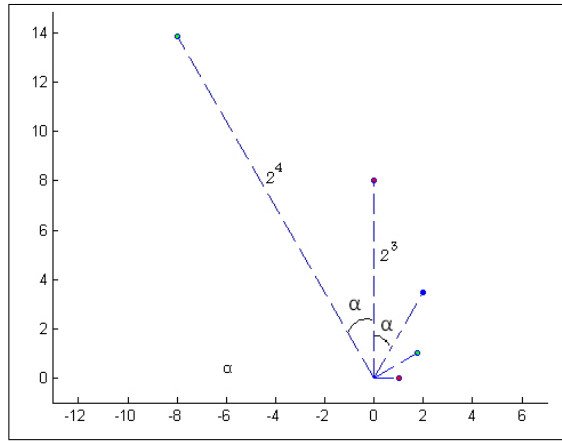
Figure 12: α-Spiral Node Chain.

a)The A_{pln} suggestion for 45-Spiral Node Chain.I(G)=13

b)The A_{MST} suggestion for 45-Spiral Node Chain.I(G)=38

c)The A_{pln} suggestion for 30-Spiral Node Chain.I(G)=18

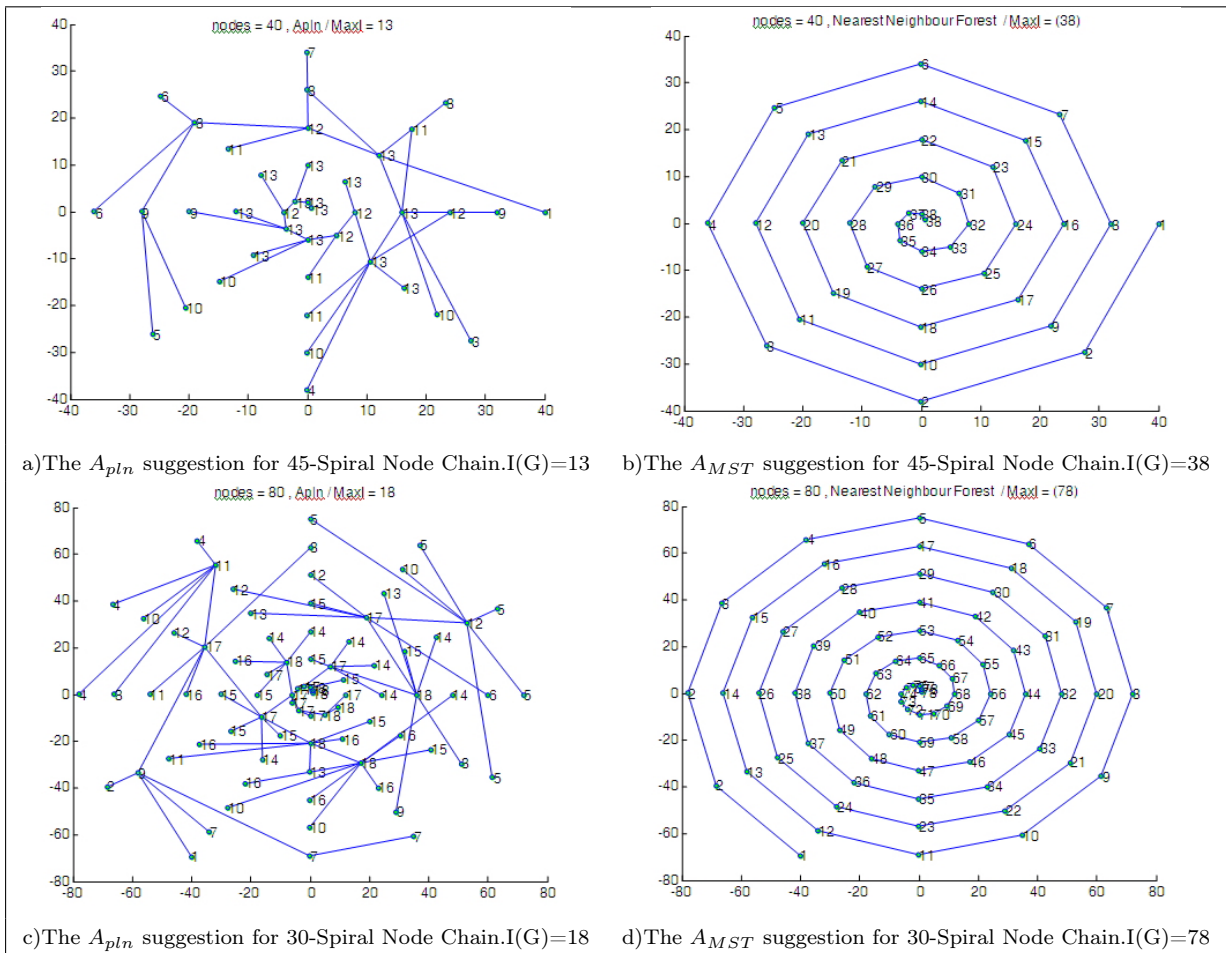d)The A_{MST} suggestion for 30-Spiral Node Chain.I(G)=78

Figure 13: 45-Spiral Node Chain with 40 and 30-Spiral Node Chain with 80 nodes and proposed topologies with A_{pln} and A_{MST}. Note that for clarity of representation, the k⁄th node is positioned in $(kcos(\alpha k), ksin(\alpha k))$ instead of $(2^k cos(\alpha k), 2^k sin(\alpha k))$. The interference of each node is wrote beside the node position.

tree with minimum interference. In this paper we can not compute the final interference order according to the input size but for some special distributions which are criterions for interference problem the resulting topology is generated and all resulting topologies are satisfactory. For the

future work by mathematical relations we will find the order of final topology according to the number of input nodes for the worst case of node distribution.

Bibliography

[1] T. Locher, P. von Rickenbach, and R. Wattenhofer: Sensor networks continue to puzzle: Selected open problems. *In Proc. 9th Internat. Conf. Distributed Computing and Networking (ICDCN)*, 2008

[2] P. Santi: Topology Control in Wireless Ad Hoc and Sensor Networks. *Wiley* , 2005

[3] X.Y. Li , W.Z. Song , W. Wan: A Unified Energy Efficient Topology for Unicast and Broadcast. *In: Proc. of the 11th Int. Conf. on Mobile Computing and Networking (MOBICOM)*, 2005

[4] M. Damian, S. Pandit, S. Pemmaraju: Local Approximation Schemes for Topology Control. *In: Proc. of the 25th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 2006

[5] M. Burkhart, P. von Rickenbach, R. Wattenhofer, A. Zollinger: Does Topology Control Reduce Interference?. *In: Proc. of the 5th ACM Int. Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc)*, 2004

[6] K. Moaveni-Nejad, X.Y. Li: Low-Interference Topology Control for Wireless Adhoc Networks. *Ad Hoc & Sensor Wireless Networks: An International Journal 1(1-2)*,2005

[7] T. Johansson, L. Carr-Motyčková: Reducing Interference in Ad hoc Networks through Topology Control. *In: Proc of the 3rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, 2005

[8] M. Benkert, J. Gudmundsson, H. Haverkort, A. Wolff: Constructing Interference- Minimal Networks. *Computational Geometry: Theory and Applications*, 2007

[9] P. von Rickenbach, S. Schmid, R. Wattenhofer, A. Zollinger: A Robust Interference Model for Wireless Ad-Hoc Networks.*In: Proc. of the 5th Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2005

[10] M.M. Halldórsson, T. Tokuyama: Minimizing Interference of a Wireless Ad-Hoc Network in a Plane. *In: Proc. of the 2nd Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks(ALGOSENSORS)*, 2006.

[11] T. Moscibroda, R. Wattenhofer: Minimizing Interference in Ad Hoc and Sensor Networks. *In: Proc. of the 3rd ACM JointWorkshop on Foundations of Mobile Computing (DIALM-POMC)*, 2005

[12] K. Buchin: Minimizing the Maximum Interference is Hard. *CoRR 2008 and sited in arXiv/0802.2134* , 2008

[13] N. Clark, C. J. Colbourn, D.S. Johnson: Unit Disk Graphs. *Discrete Mathematics, 86:165-177*, 1990

[14] M. Heide, C. Schindelhauer, K. Volbert, M. Gruenewald: Energy: Congestion and Dilation in Radio Networks. *In Proc. of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), pages 230-237*, 2002