Selecting the Best Spanning Tree to Reduce the Interference of a wireless sensor network using Genetic Algorithm

Vahid Haghighatdoost
Department of Computer Engineering,
Shahed University, Tehran, Iran

Maryam Espandar
Department of Computer Engineering,
Shahed University, Tehran, Iran

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 spanning tree for the input distribution of the nodes in the plane how the interference of the network aims the minimum value.
Keywords: Genetic algorithm, interference, sensor network, spanning tree, wireless ad-hoc network.

1. Introduction

Wireless sensor 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 spanning tree wile minimizing energy consumption. 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 that the resulting topology preserves connectivity.

The intuition was that a low minimizing the maximum degree of nodes of graph would solve the interference issue automatically, and as depicted in [1] this intuition was proved wrong in [2]. The general interference model introduced in [3], 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 [3].

A geometric graph is used for modelling of the wireless network. The graph consists of a set of nodes represented by points in the Euclidean plane; we want to connect these nodes by choosing a set of symmetric edges. A node is able to adjust its transmission power to any value between zero and its maximum power level to reach other nodes. An edge exists if and only if the maximum transmission range of both incident nodes mutually includes their counterpart. The minimum requirement of a topology control algorithm is reducing transmission power to compute a subgraph of the given network graph that preserves connectivity. *The interference of a node v is then defined as the number of other nodes that potentially affect message reception at node v*. The maximum interference of a graph is then defined as the maximum node interference.

So far, not many results have been published in the context of explicit interference minimization. For networks restricted to one dimension the authors in [3] present a $\sqrt{n}$ -approximation of the optimal connectivity preserving topology that minimizes the maximum interference. For the two dimensional case, the authors in [4] propose an algorithm that bounds the maximum interference to $O(\sqrt{n})$.

A theoretical problem in topology control which has been stated as essential to understanding sensor networks is the following:

Given *n* nodes in the plane, connect the nodes by a spanning tree. For each node *v* we construct a disk that its canter is located on node *v* with radius equal to the distance to *v*'s furthest neighbour in the spanning tree. The interference of a node v is then defined as the number of disks that include node v. Find a spanning tree that minimizes the maximum interference.

Kevin Buchin in [5] proved that the interference reduction problem is NP-complete. If we have n nodes and we want to find the best spanning tree we should generate N^{N-2} Different trees; it means for n=12 we should generate 61,917,364,224 trees and for a network with n=20, number of trees become 2.62144 E+023. In a similar work,

Manuscript received December 5, 2010
Manuscript revised December 20, 2010

Faghani and his colleagues in [6] used the genetic approach to find the best spanning Tree in metro Ethernet networks. In this paper Faghani's solution is extended and applied to solve the interference problem.

2. Interference model of network

The network is modelled as a geometric graph $G=(V,E)$. Let N_u denote the set of all neighbours of a node $u \in V$ and r_u determines the distance from u to its farthest neighbour. More precisely $r_u = \max_{v \in N_u} \{|u\text{-}v|\}$, where $|u\text{-}v|$ denotes the Euclidean distance between nodes u and v. $D(u,r_u)$ 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 neighbours. 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:

$$I(v)=\left|\{u \mid u \in V \setminus \{v\}, v \in D(u,r_u)\}\right| \quad (1)$$

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:

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

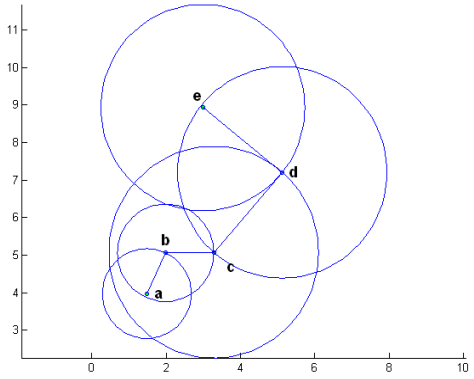


Figure 1. The interference model of a graph with 5 vertexes

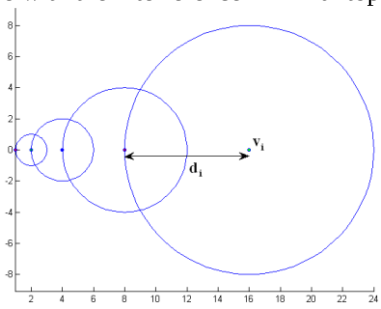
As shown in Figure 1 the interference of nodes is as follow:

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

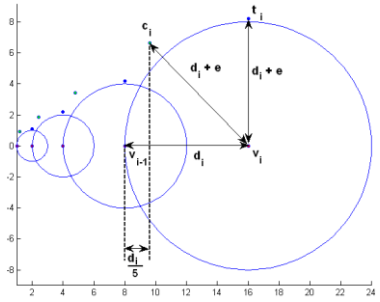
According to Definition 2 the Interference of graph *I(G)*=2.

3. The nearest neighbour forest

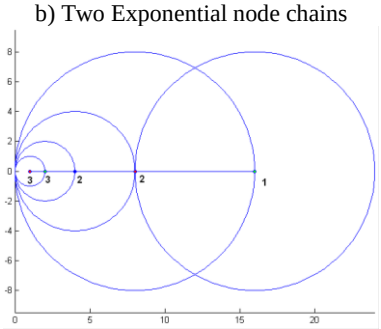
In the first view of the interference problem, one may say the nearest neighbour forest or minimum spanning tree is the best subgraph which results in minimum interference. In this section, it is shown that this is already a substantial mistake, as thus interference becomes asymptotically incomparable with the interference-minimal topology.



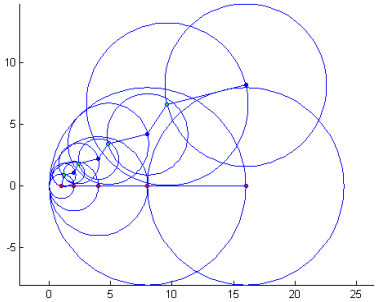
a) Exponential Node Chain



b) Two Exponential node chains



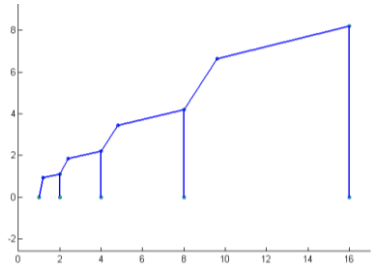
c) The nearest neighbour forest and its disk graph for exponential chain



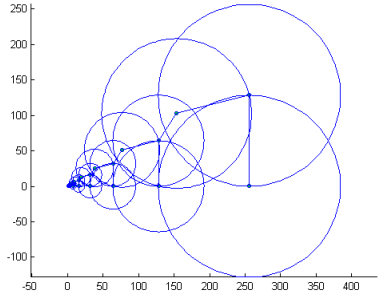
d) The nearest neighbour forest and its disk graph for Two Exponential node chains

Figure 2. Two Special node distributions and resulting topology by applying the A_{MST} algorithm

For Some special distribution the nearest neighbour forest results in the worst interference. Clark et al., in [7] 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 2(c) the nearest neighbour forest results in the interference of $\Omega(n)$. Also authors in [3] 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 neighbour, 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 Neighbour Forest for this node distribution is shown in Figure 2 (b).



a) The best spanning, I(G) always is equal to 3.



b) The disk graph of the best topology

Figure 3. Proposed topology for Two Exponential node chains with constant interference I(G) = 3 in [3]]

The algorithm proposed in [3] finds a subgraph for the exponential node chain (Figure 2(a)) with $I(G) \in O(\sqrt{n})$. They also proposed a topology with constant interference for the Two Exponential node chains which is depicted in Figure 3 but there is no algorithmic method which generates automatically similar subgraph.

Figure 4 shows the $A_{\exp}$ algorithm resulting topology for exponential chain which is proposed in [3].

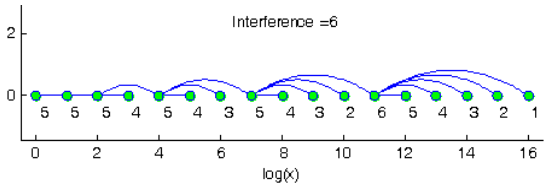


Figure 4. Result topology of $A_{\exp}$ algorithm for exponential node chain with 17 nodes. For clarity of representation edges are depicted as arcs and x dimension is shown in logaritmic scale. The interference of each node is wrote under the node position.

In the next section the solution base on genetic algorithm is proposed and our goal is finding the best topology for the distribution of nodes in the plane as well as $A_{\exp}$ for exponential node chain.

4. Genetic algorithm approach

the approximate solution. Moreover, it improves all potential solutions step by step through biological evolutionary processes like crossover, mutation, etc. Because the process in the GA approach is not wholly operated randomly but includes both directed and stochastic search embedded with a survival of the fittest mechanism, it is possible to enforce the search to reach the optimal solution.

In [8], a genetic algorithm is proposed for degree-constrained Minimum Spanning Tree problem; and in [6], the authors used the idea proposed in [8] to develop a new genetic algorithm for selecting the best spanning tree in Metro Ethernet networks based on load balance criterion.

In our paper, we use a similar idea proposed in [6] to develop a new genetic algorithm for selecting the best spanning tree to connect the nodes in a wireless sensor network distributed in the plane to aim the minimum interference through the network.

For each GA based solution, it is necessary to define:

- The individual on which it operates (encoding),
- The operators it uses,
- Some parameters such as the population size, etc.,
- An objective function.

A. Chromosome Representation (encoding)

For GA approach, it is important to determine the adequate chromosome representation of problem. One of the classical

theorems in graphical enumeration is Cayleys's theorem [9]. It states that there are N^{N-2} distinct labeled trees for a complete graph with *N* vertices. Prüfer provided a constructive proof of Cayley's theorem establishing a one-to-one correspondence between such trees and the set of all permutation of *N-2* digits. This means we can describe our tree with *N-2* uniquely digits for *N* vertices. The sequence of digits is named the Prüfer Number in [6]. Figure 5 shows two different simple trees and their Prüfer numbers. In Our GA approach the Prüfer number is used as a chromosome.

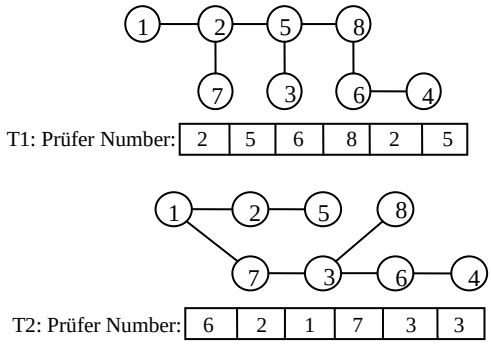


Figure 5: Two trees with their corresponding Prüfer Number

The Prüfer number encoding procedure is as below:
Step 1) Let *i* be the smallest leaf node and node *j* be incident to node *i*. Set *j* be the first digit in the encoding. The encoding is built by appending digits to the right.
Step 2) Remove node *i* and the edge from *i* to *j*.
Step 3) Repeat above operation until only one edge is left.
In a Prüfer number encoding, a tree is encoding as a Prüfer vector P and a set of its eligible nodes $\overline{P}$ (the set of all node not included in P).
The decoding procedure is as below:

Step 1) Let node *i* be the smallest eligible node of $\overline{P}$ and node j be the leftmost element of *P*. If $i \neq j$, add the edge (*i*,*j*) into the tree *T*. If *i* is no longer eligible, then remove node *i* from $\overline{P}$. Delete *j* from *P*. If *j* does not occur anywhere in the remaining part of *P*, then put it into $\overline{P}$.
Repeat the process until *P* is empty.

Step 2) For the remaining last two nodes *u* and *v* of $\overline{P}$, add the edge (*u*,*v*) into the tree *T*.
Genotypes (chromosome values) are uniquely mapped on to the decision variables (phenotypic) domain.

B. Crossover and Mutation

The genetic algorithm uses the individuals in the current generation to create the children that make up the next generation. Besides elite offspring, the individuals in the current generation with the best fitness values, the algorithm creates:

- Crossover offspring by selecting vector entries, or genes, from a pair of individuals in the current generation and combines them to form a child.
- Mutation offspring by applying random changes to a single individual in the current generation to create a child.

Crossover and Mutation are two deterministic operators in the biological evolutionary process. There are several types of crossover operator such as: single point, two point, uniform, and etc. In this paper, we use single point crossover operator. Single point crossover at first generates a random position *R* and then selects genes 1 to R from parent 1 and genes R+1 to N from Parent 2 where N is the length of each chromosome. Figure 6 shows the crossover operation.

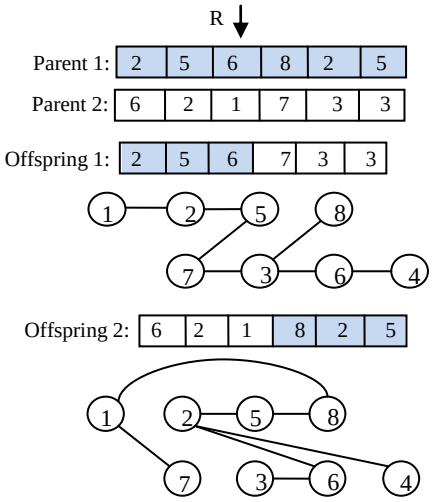


Figure 6: Crossover operation, topology of offspring are displayed under of their Prüfer Numbers

Also Mutation operator applies random changes in some chromosomes to avoid that we will not be placed in local minimum.

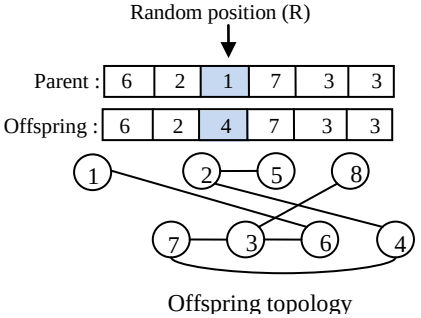


Figure 7: Mutation operation, topology of offspring is displayed under of its Prüfer Number

There are several types of mutation operation, such as: uniform, Gaussian, and etc. In this paper, we use uniform mutation operator. Figure 7 illustrates the mutation operation where a random position is selected first and its digit is replaced with another random digit.

A. Initial Population, Evaluation and Selection Operator

As the Minimum spanning tree (MST) of a graph presents the connectivity with smallest weight of graph, so we add the MST tree to the initial population. To evaluate our population, in GA approach, we need an evaluation function. We name it *Eval(X)*, in which *X* is the Chromosome (In our research is Prüfer number) and it is called fitness value for chromosome *X*. The main goal is to minimize the *Eval(X)*. The evaluation function is used to select the best chromosomes from the population why that chromosomes with higher fitness value will have more chance to be selected for next generation. In this paper, we use Roulette wheel as the selection operator. Each slice in Roulette wheel is proportional to its fitness value.

As expected we use the graph interference for evaluation function.

$$\begin{aligned} &Eval(X) = I(T) \\ &T(V,E) = Corresponding\_Tree(X) \\ &I(T) = \max_{v \in V} I(v) \\ &I(v) = \left| \{u \mid u \in V \setminus \{v\}, v \in D(u, r_u)\} \right| \end{aligned} \tag{3}$$

Where the function Corresponding_Tree returns the adjacent matrix of a graph according to its Prüfer number and $D(u,r_u)$ determines the set of nodes that are located in the disk graph centered by u with radius r_u; and r_u is the distance of farthest adjacent of node u.

B. Proposed GA method

C.

The proposed GA method is outlined as follows:

Step 1) Initialization: Choose the population size *N*, proper crossover probability P_c and mutation probability P_m, and Generate initial population *P(0)*. Let the generation number *t=0*.

Step 2) Crossover: Choose the parents from *P(t)* with probability P_c for crossover. Afterwards, randomly match every two parents as a pair and use the proposed crossover operator to each pair to generate two offspring. All offspring constitute a set denoted by S_c;

Step 3) Mutation: Selection the parents for mutation from set S_c with probability P_m. For each chosen parent, the proposed mutation operator is applied to it to generate a new offspring. These new offspring are replaced with their parents in S_c and constitute a set denoted by S_m;

Step 4) Selection: Select the best *N* individuals among the set $P(t) \cup S_m$ as the next generation population *P(t+1)* using Roulette wheel method, let *t=t+1*;

Step 5) Termination: If termination conditions hold, then stop, and keep the best solution obtained as the approximate global optimal solution of the problem; otherwise, go to step 2. Selected values for the above GA are as follow:

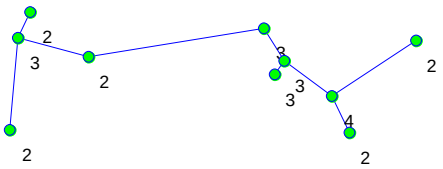
Table 1- Selected Values for Genetic approach

N	P_c	P_m	Max Generations	Stall Gen Limit
60	0.5	0.2	500	150

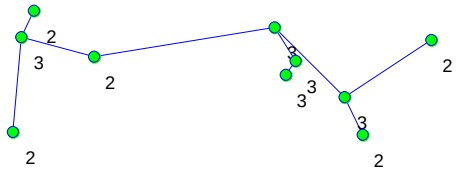
5. Simulation results

The most complex part of proposed algorithm is decoding the chromosomes and calculating of their fitness; so, the computational complexity of our algorithm is:

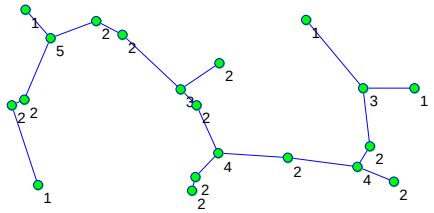
$$O(TNM^2M^2) = O(TNM^4) \tag{4}$$



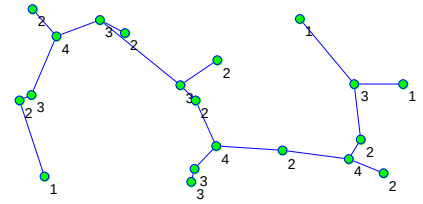
a) suggested topology by MST for 10 nodes with I=4



b) suggested topology by A_a for 10 nodes with I=3



c) suggested topology by MST for 20 nodes with I=5

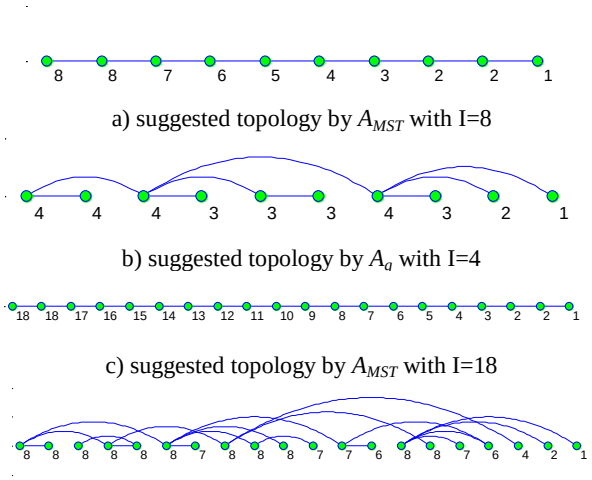


d) suggested topology by A_g for 20 nodes with I=4

Figure 8: A random distribution of 10 and 20 nodes and suggested topologies with A_{MST} and A_g

Where T is the max iteration count and N is population size and M is number of network nodes. All simulations were done in MATLAB R2009a on a computer with 2GB of RAM and an Intel(R) Core(TM)2 T 5870 CPU. Figure 8 displays the resulting topology by using the A_{MST} and A_g for random distribution of nodes in the plane. Figure 9 illustrate the resulting topology by using the MST and A_g and A_{exp} for exponential node chain distribution. Table 2 shows the final

interference of different algorithms with time complexity of Genetic approach.



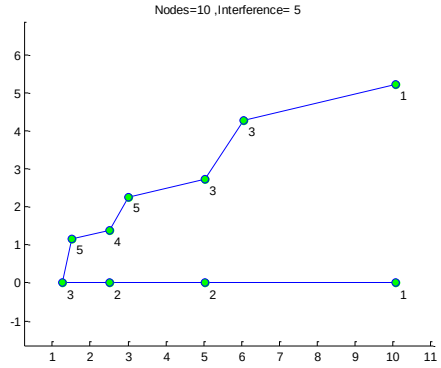
a) suggested topology by A_{MST} with I=8

b) suggested topology by A_g with I=4

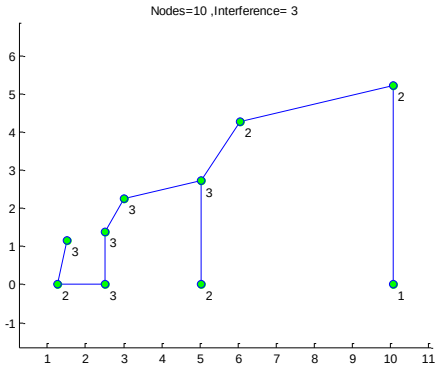
c) suggested topology by A_{MST} with I=18

d) suggested topology by A_g with I=8

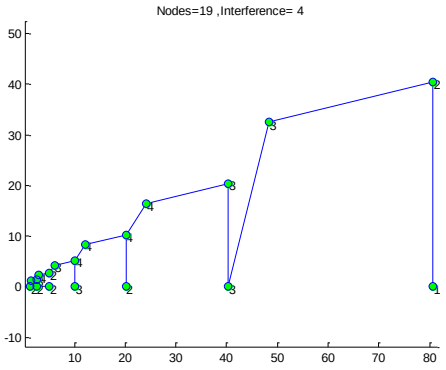
Figure 9: Exponential node chain for 10 and 20 nodes. Some edges are depicted as arcs and x dimension is shown in logaritmic scale.



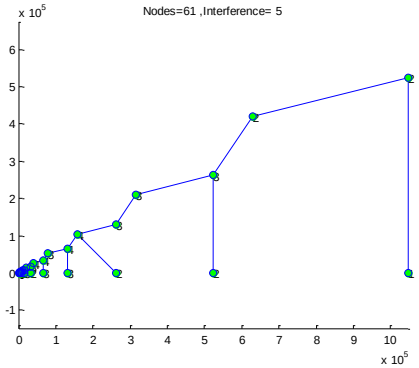
a) resulting topology by applying A_{MST} for 10 nodes



b) resulting topology by applying A_g for 10 nodes



c) resulting topology by applying A_g for 19 nodes



d) resulting topology by applying A_q for 61 nodes

Figure 10: Resulting topology for Two dimentional Exponential node chain by applying AMST and Ag

It is true that the result of A_g for Exponential node chain and Two exponential chain is not the best topology but notice that the proposed A_g is an applicable solution; also it's expressible that the result topology of A_g is acceptable for both distributions.

6. Conclusion

As proved in [5] the Interference Minimization is an NP-Complete problem and already it is introduced as an open problem in [1]. So finding the best topology with minimum interference is impossible when the nodes are more than 20. In this paper, we introduced a new Genetic Algorithm approach for finding the best spanning tree for the input distribution of wireless sensor network. We select the best tree based on the interference of resulting topology. We used the Prüfer number for encoding the individuals. It seems that the proposed algorithm could be used for distribution of nodes in the space (means in three dimensions).

Table 2- Different distributions and the interference of each one using three different algorithms MST, A_{exp} and A_g.

Distribution	Nodes count	MST interference	A_{exp} Interference	A_g Interference	A_g Time (sec)
2D Random	10	4	-	3	14.2344
2D Random	20	5	-	4	27.3906
2D Random	50	6	-	5	78.1406
Exponential chain	10	8	4	4	11.3594
Exponential chain	20	18	6	8	19.9219
Exponential chain	50	48	10	12	60.1094
Exponential chain	100	98	14	20	168.8594
Two Exponential node chains	10	5	-	3	53.7344
Two Exponential node chains	19	8	-	4	112.0625
Two Exponential node chains	61	18	-	5	578.2188

References

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

[2] 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)

[3] 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)

[4] 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)

[5] K. Buchin: Minimizing the Maximum Interference is Hard. CoRR abs/0802.2134: (Submitted on 15 Feb 2008)

[6] F. Faghani, G. Mirjalily: Selecting the Best Spanning Tree in Metro Ethernet Networks using Genetic Algorithm. International Journal of Computer Science and Network Security, Vol.8, No.6, pp. 106-112, 2008.

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

[8] G. Mirjalily, H. Karimi, F. Adibnia and S. Rajai: A new approach to select best spanning tree in Metro Ethernet Networks. 8th IEEE CIT conf., Sydney, Australia, July 2008.

[9] A. Cayley: A theorem on trees. Quarterly Journal of mathematics 23, pp.376-378, 1889.

Manuscript received December 5, 2010
Manuscript revised December 20, 2010