Summary of: Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks by Lloyd, et al

Billy Brumley (52891S)

March 04, 2005

Abstract

This paper presents a brief summary of the given article. An introduction to the subject of topology control in sensor networks is presented. The paper presents two methods for generating algorithms for topology control with minimization objectives in mind. The paper also shows how some previously developed algorithms can be generated using these methods as well. In general, these methods are for a specific kind of graph, which will be explained further. The paper also presents some experimental results, which will be summarized, analyzed, and critiqued.

1 Introduction

1.1 Goals

The subject of topology control in regards to ad hoc and sensor networks has been widely discussed as of late. Topology control is the idea of giving transmission powers to nodes of a network in a way that the network will behave in a certain fashion. This can be modeled abstractly by means of a directed graph. This is studied profusely in the sensor networks arena because of the fact that in some instances the nodes of a sensor network have a very limited amount of power.

So what 'certain fashion' would be desirable? In general, one might want to limit the node communication radius or maybe have a certain level of connectivity. In the paper[1], there were a couple of properties that were researched in depth. These properties are minimizing the maximum power assigned to a node, and minimizing the total power of all nodes. These are abbreviated respectively as $MAXP$ and $TOTALP$.

1.2 Methods and Notation

A directed graph that models a given sensor network can be constructed using the following method. Each node or transceiver is considered a vertex in the graph. Each pair (u, v) of nodes has an associated transmission power threshold (*tpt*), or $p(u, v)$. This means that if node u can only transmit successfully to node v if its current power level is at least $p(u, v)$. If u can transmit to v, then a directed edge is inserted from vertex u to v. This is done for every transceiver, and the end result is a directed graph which models the network.

The general form for any topology control problem is denoted as (M, P, O). M represents the graph model, in this context meaning directed or undirected. P is the wanted graph property, for example *k-connectivity*, which is obviously the smallest number of vertices that when removed yields a disconnected graph (e.g. a tree is 1-connected, a cycle 2-connected, etc). O is the minimization objective, which in the paper is limited to $MAXP$ and $TOTALP$.

Properties P of a specific type, known as *monotone*, are at the heart of this paper. A *monotone* property is one that holds even when the *tpt* of a node is increased. Obviously when *tpt* is increased, for our graph representation this means more edges are getting added. An example of a monotone property is k-connectivity. An example of a non-monotone property is acyclic.

When approximation algorithms are used, two factors are studied. The first is the *performance guarantee*, or ρ. This means that the solution is within a multiplicative factor of ρ of the optimal solution. A *polynomial time approximation scheme (PTAS)* is produced when given a problem instance, an accuracy requirement ϵ, a solution is produced that is within a factor of $1 + \epsilon$ of optimal.

Basically, this paper presents a method for generating algorithms for the described type of topology control problems. Once the method of generation is established, some example algorithms are given. This is done for both $MAXP$ and $TOTALP$. Limiting properties to monotone can produce non NP-hard algorithms which can be very efficient.

2 Minimizing $MAXP$

2.1 Algorithm for monotone and polynomial time testable properties

First, the authors provide some detail of why monotone properties are desirable. Lemma 4.1 states: For any instance of *(Undir, P, MAXP)* and *(Dir, P, MAXP)* where P is monotone, an optimal solution exists where all nodes have the same *tpt*. This is easily provable, since the highest *tpt* of any node in the graph can obviously be assigned to the rest of the nodes and have P still hold, since it is monotone (This is obviously a very inefficient way of doing things, but let's disregard that fact for a few moments).

Theorem 4.1 is stated next, which reads: For any monotone and polynomial time testable graph property P, the problem *(Undir, P, MAXP)* and *(Dir, P, MAXP)* can be solved in polynomial time. This theorem is also easily provable, and is presented. With Lemma 4.1, we know there is an optimal solution with all nodes having equal *tpt*. For a given node u, $n-1$ different *tpt* values need to be considered. So for all nodes, this makes $n(n-1)$, so $O(n^2)$. To find the solution, the $O(n^2)$ values are sorted, then binary searched $O(logn)$. $FP(n)$ is the time needed when testing if P holds. Testing is polynomial time, which yields a running time of $O((n^2 + FP(n))logn)$, which is obviously polynomial.

So, this seems like a good, solid method. An example is given for the problem *(Undir, 2-Connected, MAXP)*. A $O(n^2)$ method is known for testing 2-Connectivity, so an algorithm with running time of $O(n^2 logn)$ is produced. Obviously this method can only be applied to a very specific type of sensor network.

The next idea presented in the paper is similar to the one above, but now the problem is looking at properties P that are not polynomial time testable. This yields an approximation algorithm. So formally, *(Undir, P, MAXP)* and *(Dir, P, MAXP)* with any arbitrary P can be approximated within factor $1 + \epsilon$ in $O((n^2 + FP(n))loglog(max/min))$, where max/min are the max and min *tpt* values. A proof is presented which is very logical. an important thing to note here is that when the max/min ratio is much smaller than 2^n, the property testing portion is smaller than $O(logn)$.

2.2 Why Just Monotone?

The authors proceed to present an example of how non-monotone properties such as G is a tree render the problem NP-hard. The proof is very extensive and involves a reduction from Exact Cover by 3-Sets (X3C), which is NP-hard. Basically, the point is that restricting to monotone properties is the only way to guarantee efficient algorithm generation using these ideas.

2.3 The Disadvantages

As mentioned earlier, assigning the maximum *tpt* to all the nodes is really not practical. A truly practical solution would minimize the number of nodes that have the maximum *tpt*. Unfortunately, adding this additional requirement renders the problem NP-Hard. So basically, these methods presented only work on very specific types of sensor networks, and don't really work all that well.

3 Minimizing $TOTALP$

3.1 Approximation Algorithm for Minimizing $TOTALP$

In the next section, the authors analyze the $TOTALP$ problem. They jump right into approximation algorithms because of the fact that even seemingly simple properties, such as 1-connectivity are NP-Hard. For this algorithm, the property P is also monotone (and at first polynomial time testable, but later on not), and the *tpt* values are symmetric, meaning abstractly that the graph is undirected. A general outline of the algorithm called *Gen-Total-Power (GTP)* is given as follows from the text:
Input: A problem instance I of $(Undir, P, TOTALP)$.
Output: A *tpt* $\pi(u)$ for each node where P holds and $TOTALP$ is minimized.
Steps:

1. From I, construct the complete graph $G_c(V, E_c)$. Weights are given by the *tpt.*

2. Construct a subgraph $G'(V, E')$ of G_c where G' satisfies P and the sum of all the edge weights in E' is minimized as compared to all other subgraphs G_c.

3. For every node u, assign a *tpt* $\pi(u)$ which equals the largest edge weight incident with u.

This is nothing fancy. Obviously, the choke point is in step 2. The next proof provides a performance guarantee for the above algorithm. Given $G_c(V, E_c)$, we have a subgraph $H(V, E_H)$ with min $TOTALP$ and $W(H)$ equal to the total weight. Step 2 produces subgraph $G'(V, E')$ of G. So given $\alpha > 0$ and $\beta > 0$ where *W(H)*$\leq \alpha$*OPT(I)* and $W(G') \leq \beta W(H)$, it follows *GTP(I)*$\leq 2\alpha\beta$*OPT(I)*, meaning performance guarantee is $2\alpha\beta$.

3.2 Using GTP on Existing Algorithms

This is illustrated with the problem *(Undir,1-Connected,TOTALP)* using a pre-existing algorithm, which yields a performance guarantee of 2. The only example provided simply shows how this pre-existing algorithm can be derived using the method above, so it's nothing new really.

3.3 Using GTP to Make "New" Algorithms

In the proofs that follow in the paper, the problem *(Undir,2-Node Connected,TOTALP)* is analyzed. The authors use a pre-existing approximation algorithm with performance guarantee of $(2 + 1/n)$, so $\beta <= (2 + 1/n)$ in Step 2 of GTP, so their use of the term "New" is maybe a little off.

They show that from Step 1's $G_c(V, E_c)$, there is a Step 2 subgraph $G'(V, E')$ where G' satisfies P (2-connected) and $TOTALP$ upper bound is $(2-2/n)OPT(I)$, and we have $\alpha \leq (2-2/n)$. So combining both of these ideas, the performance guarantee is $2(2-2/n)(2+1/n)$ which as n grows approaches 8. This is extended into an algorithm for *(Undir,2-Edge Connected,TOTALP)* with performance guarantee of $8(1-1/n)$ which also approaches 8.

4 Results and Analysis

4.1 Setup

In their experimental results, the authors have compared their *(Undir,2-Node Connected,TOTALP)* algorithm with an existing *(Undir,2-Node Connected,MAXP)* which minimizes the power levels of each node. This is done since their $TOTALP$ algorithm is the only one in existence.

Radio propagation is constructed to emulate 2.4GHz. The node density is varied from 0.625 nodes/sq mile to 6.26 nodes/sq mile in a 4x4 mile area (10-100 nodes). They are placed uniformly as well as skewed. Both average and maximum *tpt* values are measured after trial runs.

4.2 Results

For uniform distribution, with average power $TOTALP$ outperforms $MAXP$ by 5% to 19% which increases with density. However, the max power with $TOTALP$ is 14%-37% higher. $MAXP$ is 60%-83% of max while with $TOTALP$ it is 39%-70% of max, decreasing as density increases. This is to be expected, as you would think that average power would decrease faster than maximum power.

For skewed distribution, with average power $TOTALP$ outperforms $MAXP$ by 6% to 25%, and the difference between max and average is higher. $MAXP$ is 40%-76% of max while with $TOTALP$ it is 25%-64% of max. So with skewed distribution, $TOTALP$ gives a higher maximum power but a lower average power. This is also to be expected, since more dense areas have much lower power levels in regards to average power.

Node degree was also tested. $TOTALP$ yielded average degree of 2.73. Max and average degrees were consistently lower with $TOTALP$ than with $MAXP$. They state that smaller is generally better.

Experiments were also conducted on the $TRANSIMS$ network, which is much more realistic. Instead of transmission power, transmission range was measured, due to the density of this network. The results were consistent with those given above.

5 Conclusion

This paper was very interesting to read. It contained an abundance of theory, with a touch of practice. The methods presented for $MAXP$ minimization are not really practical. Other algorithms exist that have much better features (minimization of max power of every nodes) with the same running time, for example the $MAXP$ algorithm that $TOTALP$ was compared to in the second half of the paper.

The $TOTALP$ algorithm presented is much more interesting. The experiment results seem favorable, however with no real competitors to compare this algorithm with, I think the results provided can be misleading. However, better to compare apples with oranges than nothing at all. It will be interesting to compare this algorithm with future algorithms that have the same minimization objectives in mind.

References

[1] E. L. Lloyd, M. V. Marathe, R. Ramanathan, S. S. Ravi, Algorithmic aspects of topology control problems for ad hoc networks. Mobile Networks and Applications 10 (2005), 19-34.