ON THE COMPLEXITY OF THE BROADCAST SCHEDULING PROBLEM

SERGIY I. BUTENKO, CLAYTON W. COMMANDER*, AND PANOS M. PARDALOS

ABSTRACT. In this paper, a BROADCAST SCHEDULING PROBLEM (BSP) in a time division multiple access (TDMA) ad hoc network is analyzed in terms of its computational complexity. In a TDMA frame, time is divided into equal length transmission slots in which messages are scheduled. The objective of the considered version of the BSP is to provide a collision free broadcast schedule which minimizes the total frame length. Wang and Ansari [6] have observed that the recognition version of the BSP is $\mathcal{NP}$-complete; however, their proof was incorrect. Here, we provide a counterexample to their argurment and give a correct proof that indeed the problem is $\mathcal{NP}$-complete.

1. INTRODUCTION

Due to a large number of practical applications, the importance of ad hoc networks has been increasing in the past few years. These networks provide high-speed communications between a large number of potentiall mobile stations which may be geographically disbursed. Each staiton is capable of both sending and receiving messages over the network. Hence, if a message is sent and the intended station is not able to receive it from the transmitting station, other intermediate stations may serve as relays by forwarding the message to the intended recipient.

Since all stations share the same transmission channel, stations must be scheduled to transmit messages in such a manner that prevents collisions [7]. There are two types of message collision. The first, herein referred to as *direct collision*, occurs when two neighboring stations transmit during the same time slot. The second, which will be called *hidden collision* is a result of two non-neighboring stations transmitting simultaneously to a station that can receive messages from both transmitting stations. One of the objectives of a broadcast schedule is to guarantee collision free transmissions. The time division multiple access (TDMA) protocol [5] can be used for that purpose. In a TDMA network, time is divided into frames, with each frame consisting of a number of unit length slots. The goal is to schedule message transmissions in the slots such that

(a) each station transmits in at least one time slot;
(b) simultaneous broadcasts in the same time slot will not cause any collision, either direct or hidden;
(c) the total number of slots (frame length) is minimized.

There are otherimportant criteria that influence the effectiveness of a broadcast schedule. For example, a common optimization objective in broadcast scheduling research is, given a frame length, maximize the slot usage per frame, which will in turn maximize the throughput [6, 7]. However, in this paper we deal only with the

*Corresponding author.

objective of minimizing the total number of slots in each frame. More specifically, we concentrate on computational complexity of the BSP with this objective.

We show that BSP belongs to a class of computationaly difficult problems known as $\mathcal{NP}$-complete. Belonging to this class suggests that an algorithm which solves the problem to optimality in polynomial time does not exist unless $\mathcal{P} = \mathcal{NP}$ [4]. This complexity result was first mentioned by Wang and Ansari [6], but their proof was flawed. In Section 3, we discuss some shortcomings of their proof and provide a correct proof that that the recognition version of BSP is indeed $\mathcal{NP}$-complete. Complexity results for other related problems can be found in [2] and [3].

2. Review of Graph Theory

An ad hoc network can be conveniently described by an undirected graph $G = (V, E)$, where the vertices in V represent the stations in the network and E is the set of links. Therefore, we introduce some background concepts from graph theory before continuing our discussion.

First, for $v \in V$ let $N(v) \subseteq 2^V$ represents the set of neighbors of v. That is, $N(v)$, referred to as the *neighborhood of* v, contains any node $w \in V$ such that $(w, v) \in E$.

For a subset $W \subseteq V$, let $G(W)$ denote the subgraph induced by W on G. A set of vertices $I \subseteq V$ is called an *independent set* if for every $i, j \in I$, $(i, j) \notin E$. That is, the graph $G(I)$ induced by I is edgeless. An independent set is *maximal* if it is not a subset of any larger independent set (*i.e.*, it is maximal by inclusion), and *maximum* if there are no larger independent sets in the graph. The maximum cardinality $\alpha(G)$ of an independent set of G is called the *independence number* of G.

A *legal (proper) coloring* of G is an assignment of colors to its vertices so that no pair of adjacent vertices has the same color. A coloring induces naturally a partition of the vertex set into independent sets that are precisely the subsets of vertices being assigned the same color. If there exists a coloring of G that uses no more than k colors, we say that G admits a k-coloring (G is k-colorable). The minimal k for which G admits a k-coloring is called the *chromatic number* of G and is denoted by $\chi(G)$. The graph coloring problem is to find $\chi(G)$ as well as the partition of vertices induced by a $\chi(G)$-coloring. For other standard concepts from graph theory that are used in this paper, the reader is referred to a textbook on the subject (see, *e.g.*, [1]).

3. Computational Complexity

This section presents the computational complexity results for the BSP. It was first noted that the BSP is $\mathcal{NP}$-complete by Wang and Ansari in [6]. However, their proof of the $\mathcal{NP}$-completeness of the recognition version of the problem was incorrect due to some faulty arguments. Namely, they claimed that the graph coloring problem is equivalent to the maximum independent set problem based on an incorrect assumption that, given an arbitrary graph, an optimal coloring can be found by recursively computing a maximum independent set and removing it from the graph. Thus, by coloring different independent sets in different colors, they claim that the chromatic number of the graph equals the total number of independent sets computed.

Figure 3 presents a counterexample to this statement. It is easy to see that the independence number of the graph in this figure is 3. Assuming that the first maximum independent set found using the so-claimed "optimal" coloring algorithm of Wang and Ansari is $\{4, 5, 6\}$ and consequently removing this set from the graph, we obtain a clique of the three vertices $\{1, 2, 3\}$. The independence number of the remaining graph is 1, so all three of the remaining vertices have to be colored in different colors. Thus, the Wang-Ansari coloring algorithm results in a 4-coloring. However, it is easy to see that the chromatic number of this graph is 3. For example, one optimal coloring is given by the following partition: $\{1, 5, 6\}, \{2, 4\}, \{3\}$. Therefore, the coloring obtained using the Wang-Ansari approach is not optimal.

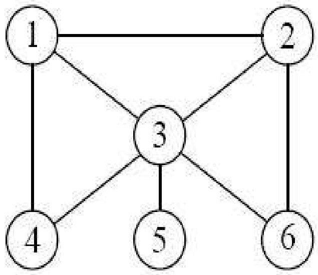

FIGURE 1. A counterexample to the claim of Wang & Ansari that optimal graph coloring can be found by recursively finding a maximum independent set and removing it from the graph.

Next we prove that the recognition version of the BSP is in fact $\mathcal{NP}$-complete. We consider the following problem:

BROADCAST SCHEDULING PROBLEM
INSTANCE: A undirected graph $G = (V, E)$ and an integer K.
QUESTION: Does there exist a broadcast schedule with frame length $\leq K$?

Theorem 1. BROADCAST SCHEDULING PROBLEM *is $\mathcal{NP}$-complete.*

Proof. To show that K-BSP is $\mathcal{NP}$-complete, we need to show that (1) K-BSP $\in \mathcal{NP}$; (2) Some $\mathcal{NP}$-complete problem reduces to K-BSP in polynomial time. Suppose that $n = |V|$ and $m = |E|$. Without the loss of generality, we assume that G is connected (if it is not, we can consider each connected component separately).

(1) K-BSP $\in \mathcal{NP}$ since a given broadcast schedule with frame length $k \leq K$ can be verified for validity in $\mathcal{O}(n^3)$ time. Indeed, the verification of validity consists of checking, for each vertex $i \in V$, that the set L_i of all time slots in which the vertices from $\{i\} \bigcup N(i)$ transmit according to the given schedule does not contain any repeated elements. This can be done using the sorting of time slot numbers in L_i in $\mathcal{O}\big((|L_i| + 1) \log(|L_i| + 1)\big)$ time for vertex i, therefore the total run time will be
$$\mathcal{O}\left(\sum_{i=1}^{n}(|L_i + 1) \log(n)\right) = \mathcal{O}\big((m + n) \log(n)\big) = \mathcal{O}(n^3).$$

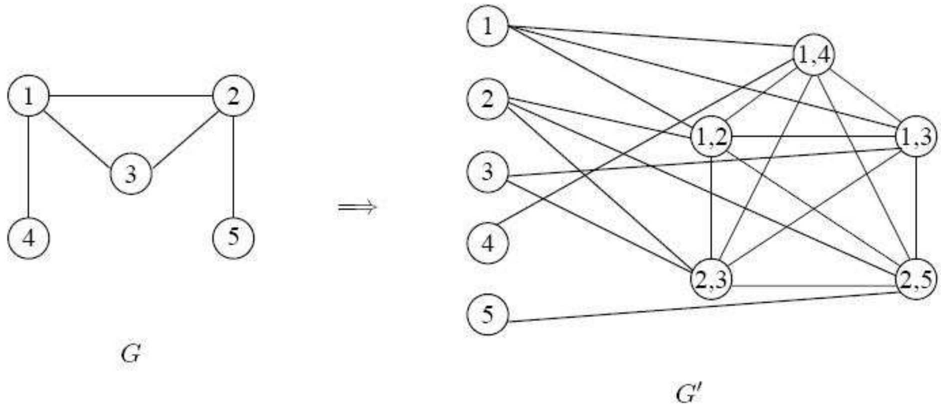

FIGURE 2. An illustration to the construction of graph G' from G.

(2) We will show that the graph k-coloring problem can be reduced to K-BSP in polynomial time. Recall that the k-coloring problems is, given $G = (V, E)$ and an integer k, does there exist a proper coloring of vertices of G that uses $\leq k$ colors? This is a well-known $\mathcal{NP}$-complete problem [4].

Given a graph $G = (V, E)$, we will construct the corresponding graph $G' = (V', E')$, where $V' = V \bigcup E$ and $E' = \{[i, (i, j)] : (i, j) \in E, i, j \in V\} \bigcup \{(e_1, e_2) : e_1, e_2 \in E\}$ (see Figure 3). Obviously G' can be constructed in polynomial time. Moreover, G has a proper k-coloring if and only if G' allows a broadcast schedule with frame length $\leq k + m$. To see this, note that by the construction of graph G', $(v_1, v_2 \in E$ if and only if $v_1, v_2 \in V$ are 2-hop neighbors in G'. Also, $V' \setminus V$ forms a clique in G', and any vertex in this clique is a 2-hop neighbor of any vertex in V, since G is connected. Thus no other vertex can transmit in the same time slot with a vertex from the clique, so any broadcast schedule in G' will require m time slots just for vertices from the clique to transmit.

The remaining vertices in V' (*i.e.*, vertices from V) can transmit according to any proper coloring in G, where different time slots in the resulting broadcast schedule will correspond to different colors in the coloring. Therefore, there is a one-to-one correspondence between proper colorings in G and feasible broadcast schedules in G'. We see that in fact k-coloring reduces in polynomial time to K-BSP, where $K = k + m$. Thus, the proof is complete.

□

4. CONCLUSIONS

In this paper, we introduced the BROADCAST SCHEDULING PROBLEM in TDMA networks. After a brief review of some principles from graph theory we examined the computational complexity of the problem. First we provided a counterexample to an earlier attempt to show that the problem is $\mathcal{NP}$-complete. Finally, we provided a correct proof confirming that the problem is indeed $\mathcal{NP}$-complete.

References

[1] C. Berge. *Graphs and Hypergraphs*. North-Holland Mathematical Library, 6, 1976.

[2] A. Ephremides and T.V. Truong. Scheduling broadcast in multihop radio networks. *IEEE Transactions on Communications*, vol. 38, pages 456-460, 1990.

[3] S. Even, O. Goldreich, S. Moran, and P. Tong. On the NP-completeness of certain network testing problems. *Networks*, 14, 1984.

[4] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Sons, New York, 1979.

[5] L. Kleinrock and J. Silvester. Spatial reuse in multihop packet radio networks. *Proceedings of the IEEE 75*, 1987.

[6] G. Wang and N. Ansari. Optimal broadcast scheduling in packet radio networks using mean field annealing. *IEEE Journal on Selected Areas in Communications*, 15(2), pages 250-260, 1997.

[7] J. Yeo, H. Lee, and S. Kim. An efficient broadcast scheduling algorithm for TDMA ad-hoc networks. *Computers and Operations Research*, 29, pages 1793-1806, 2002.

(S.I. Butenko) Dept. of Industrial and Systems Engineering, Texas A& M University, College Station, TX, 77843 USA.

E-mail address: `butenko@tamu.edu`

(C.W. Commander) Air Force Research Laboratory, Munitions Directorate, and Dept. of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 USA.

E-mail address: `clayton.commander@eglin.af.mil`

(P.M. Pardalos) Center for Applied Optimization, Dept. of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 32611 USA.

E-mail address: `pardalos@ufl.edu`