

A COMBINATORIAL ALGORITHM FOR THE TDMA MESSAGE SCHEDULING PROBLEM

CLAYTON W. COMMANDER AND PANOS M. PARDALOS

ABSTRACT. In this paper, we introduce a combinatorial algorithm for the message scheduling problem on Time Division Multiple Access (TDMA) networks. In TDMA networks, time is divided in to slots in which messages are scheduled. The total number of slots required for all stations to broadcast without message collisions is called the frame length. The objective is to provide a broadcast schedule of minimum frame length which also provides the maximum throughput. This problem is known to be \mathcal{NP} -hard, thus efficient heuristics are needed to provide solutions to real-world instances. We present a two-phase algorithm which exploits the combinatorial structure of the problem in order to provide high quality solutions. The first phase finds a feasible frame length in which the throughput is maximized in phase two. Computational results are provided and compared with other heuristics in the literature as well as to the optimal solutions found using a commercial integer programming solver. Experiments on 63 benchmark instances show that the proposed method is able to provide optimal frame lengths for all cases with near optimal throughputs.

1. INTRODUCTION

Wireless mesh networks (WMNs) have become an important means of communication in recent years. In these networks, a shared radio channel is used in conjunction with some packet switching protocol to provide high-speed communication between many users. The stations in the network act as transmitters and receivers, and are thus capable of utilizing a multi-hop transmission procedure. The advantage of this is that several stations can be used as relays to forward messages to the intended recipient, allowing for beyond line of sight communication for stations that are geographically disbursed and potentially mobile [3].

Mesh networks have increased in popularity in the recent years and the number of applications is steadily increasing [16]. As mentioned in [1], WMNs allow users to integrate various networks, such as Wi-Fi, the internet and cellular systems. WMNs can also be utilized in a military setting in which tactical datalinks network various communication, intelligence, and weapon systems allowing for streamlined communication between several different entities. For a survey of wireless mesh networks, the reader is referred to [1].

In WMNs, the critical problem involves efficiently utilizing the available bandwidth to provide collision free message transmissions. Unfettered transmission by the network stations over the shared channel will lead to message collisions. Therefore, some medium access control (MAC) scheme should be employed to schedule message transmissions so that collisions are prevented. The time division multiple access (TDMA) protocol is a MAC scheme introduced by Kleinrock in 1987 which was shown to provide collision free broadcast schedules [11]. In a TDMA network, time is divided into frames with each frame consisting of a number of unit length slots in which the messages are scheduled. Stations scheduled in the same slot broadcast simultaneously. Thus, the goal is to schedule as many stations as possible in the same slot so long as there are no message collisions.

When considering the message scheduling problem on TDMA networks, there are two optimization problems which must be addressed [21]. The first involves finding the minimum frame length, or the number of slots required to schedule all stations at least once. The second problem is that of maximizing the number of stations scheduled within each slot, thus maximizing the throughput. Both of these problems however, are known to be \mathcal{NP} -hard [3]. Thus, efficient heuristics are needed to quickly provide high quality solutions to real-world instances.

In this paper, we introduce a new heuristic which exploits the combinatorial nature of the problem to provide good solutions for the message scheduling problem. The method first computes a feasible frame length and then attempts to maximize the throughput within the frame. The organization of the paper is as follows. In the next section we formally define the message scheduling problem on TDMA networks. Using a graph theoretical description, we are then able to design a heuristic for quickly providing excellent broadcast schedules. In Section 4, we provide computational results on over 60 benchmark instances comparing the proposed method to several heuristics from the literature. We also compare our solutions to the optimal solutions found using Xpress-MP[®] by Dash Optimization [10]. Conclusions are given in Section 5.

2. PROBLEM DESCRIPTION

A TDMA network can be conveniently described as a graph $G = (V, E)$ where the vertex set V represents the stations and the set of edges E represents the set of communication links between adjacent stations. There are two types of message collisions which must be avoided when scheduling in TDMA networks. The first, called a direct collision occurs between *one-hop neighboring stations*, or those stations $i, j \in V$ such that $(i, j) \in E$. One-hop neighbors which broadcast during the same slot cause a direct collision. Further, if $(i, j) \notin E$, but $(i, k) \in E$ and $(j, k) \in E$, then i and j are called *two-hop neighbors*. Two-hop neighbors transmitting in the same slot cause a hidden collision [3].

Assume that there are M slots per frame. Further, assume that packets sent at the beginning of each time slot and are received in the same slot in which they are sent. Let $x : M \times V \rightarrow \{0, 1\}$, be a function where

$$x_{mn} = \begin{cases} 1, & \text{if station } n \text{ scheduled in slot } m, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Also, let $c : E \rightarrow \{0, 1\}$ return 1 if i and j are one-hop neighbors, i.e., if $(i, j) \in E$ and $i \neq j$.

As mentioned above, there are two problems which have to be solved in order to obtain optimal broadcast schedules using the TDMA protocol. The first is the FRAME LENGTH MINIMIZATION PROBLEM (FLMP) and the second is the THROUGHPUT MAXIMIZATION PROBLEM (TMP). Using the aforementioned definitions and assumptions, we can now formulate the MESSAGE SCHEDULING PROBLEM ON TDMA NETWORKS (MSP-TDMA) as the following multiobjective optimization problem:

Minimize M

$$\text{Maximize } \sum_{i=1}^M \sum_{j=1}^{|V|} x_{ij}$$

subject to:

$$\sum_{m=1}^M x_{mn} \geq 1, \quad \forall n \in V, \quad (2)$$

$$c_{ij} + x_{mi} + x_{mj} \leq 2, \quad \forall i, j \in V, i \neq j, m = 1, \dots, M, \quad (3)$$

$$c_{ik}x_{mi} + c_{kj}x_{mj} \leq 1, \quad \forall i, j, k \in V, i \neq j, j \neq k, k \neq i, m = 1, \dots, M, \quad (4)$$

$$x_{mn} \in \{0, 1\}, \quad \forall n \in V, m = 1, \dots, M, \quad (5)$$

$$M \in \mathbb{Z}^+. \quad (6)$$

The objective provides a minimum frame length with maximum bandwidth utilization, while constraint (2) ensures that all stations broadcast at least once. Constraints (3) and (4) prevent direct and hidden collisions, respectively. We note here that will not be attempting to solve this problem by using the typical multiobjective optimization approach, in which one combines the multiple objectives into one scalar objective whose optimal value is a Pareto optimal solution to the original problem. Instead we will decouple the objectives and handle each independently. This is done because for the MSP-TDMA, frame length minimization usually takes precedence over the utilization maximization problem. This is the usual modus operandi used by other heuristics in the literature [19, 21, 18].

Suppose that we relax the MSP-TDMA and only consider the first objective function. This is referred to as the FRAME LENGTH MINIMIZATION PROBLEM (FLMP) and is given by the following integer program: $\min\{M : (2) - (6)\}$. Clearly any feasible solution to this problem is feasible for BSP-TDMA. Now, consider a graph $G' = (V, E')$ where V follows from the original communication graph G , but whose edge set is given by $E' = E \cup \{(i, j) : i, j \text{ are two-hop neighbors}\}$. Then using this augmented graph, we can formulate the following theorem.

Theorem 1. *The FRAME LENGTH MINIMIZATION PROBLEM on $G = (V, E)$ is equivalent to finding an optimal coloring of the vertices of $G'(V, E')$.*

Proof. Recall that in order for a message schedule to be feasible, all stations must broadcast at least once and no collisions occur, either hidden or direct. Notice now that E' contains both one-hop and two-hop neighbors, and in any feasible solution, neither of these can transmit in the same slot. Thus, there is a one-to-one correlation between time slots in G and vertex colors in G' . Hence, a minimum coloring of the vertices of G' provides the minimum required slots needed for a collision free broadcast schedule on G . \square

After one has successfully solved the FLMP yielding an optimal frame length M^* , then the THROUGHPUT MAXIMIZATION PROBLEM (TMP) given as follows $\max\{\sum_{i=1}^{M^*} \sum_{j=1}^{|V|} x_{ij} : (2) - (6)\}$ can be solved, where M is replaced by M^* in (2) – (6). It turns out that both the FLMP and the TMP have been shown to be \mathcal{NP} -hard [3, 5]. Thus it is unlikely that a polynomial algorithm exists for finding the optimal broadcast schedule [6]. It is interesting to note however, that if we ignore constraint (4) which prevents two-hop neighbors from transmitting simultaneously, then the resulting problem is in \mathcal{P} , and a polynomial time algorithm is provided in [7].

3. COMBINATORIAL ALGORITHM FOR TDMA MESSAGE SCHEDULING

The inherent intractability of the problem motivates the need for efficient heuristics to quickly provide good solutions for non-trivial instances. In this section, we describe a new algorithm for the MESSAGE SCHEDULING PROBLEM IN TDMA NETWORKS. Our heuristic is a two-phase iterative procedure for which pseudocode is provided in Figure 1. First, we concentrate primarily on the frame length minimization portion of the MSP-TDMA by using a greedy heuristic for graph coloring which computes near optimal

solutions for the FLMP. Since this solution will only have each station transmitting exactly once, a local improvement method is then applied which attempts to maximize the throughput within the derived frame length. To increase the efficiency of the procedure, the BurstMaximization procedure is only entered if the current frame length M is at least as small as the current best value M^* . After some specified number of iterations, the algorithm terminates returning the best overall solution, which consists of the frame length M^* , the total number of bursts X^* , and the schedule of slot assignments V^* .

```

procedure ComAlgBSP( $G'$ )
1   $M^* \leftarrow |V|$ 
2   $X^* \leftarrow 0$ 
3  for  $i = 1$  to MaxIter do
4     $M \leftarrow \text{SlotMinimization}(G', \alpha, \text{SlotIter})$ 
5    if  $M \leq M^*$  then
6       $M^* \leftarrow M$ 
7       $X \leftarrow \text{BurstMaximization}(G', M^*, V^*)$ 
8    end
9    if  $X \geq X^*$  then
10      $X^* \leftarrow X$ 
11   end
12 end
13 return ( $M^*, X^*, V^*$ )
end procedure ComAlgBSP

```

FIGURE 1. Pseudocode of the proposed heuristic for BSP-TDMA.

3.1. Frame Length Minimization. For the first phase of the algorithm, we apply a greedy construction heuristic to determine the value for M , the number of time slots required for all stations to transmit. As a result of Theorem 1, the method is based on the construction phase of the Greedy Randomized Adaptive Search Procedure (GRASP) [17] for coloring sparse graphs proposed by Laguna and Martí in [13]. This particular method was chosen because it is able to quickly provide excellent solutions for the frame length. That being said, any other coloring heuristic would work fine for the frame length minimization phase. In fact, in [21] a method based on Sequential Vertex Coloring was used to determine the value of M . However, the randomized approach of the chosen method allows us to explore the search space more thoroughly and provides several feasible solutions to work with in the throughput maximization phase. This is because different optimal colorings will yield different solutions in the second phase. Furthermore since sparse graphs usually contain an exponential number of optimal colorings [12], the chosen method leads to a variety of solutions to explore in phase two. Pseudocode for this routine is given in Figure 2.

Our implementation of the frame length minimization heuristic is exactly as described in [13]. The procedure takes the augmented graph G' , a proportional parameter α , and a value SlotIter as input and creates an initial broadcast schedule one slot at a time. The value $\alpha \in [0, 1]$ determines the amount of randomness, or conversely greediness that the procedure uses. SlotIter is the number of candidate schedules for a particular slot from which the best is chosen.

Initially, the frame length M is initialized to 0 and V' , the set of unscheduled stations is initialized to V . The initial schedule is created in the **while** loop from lines 3 – 27. After incrementing the frame length, the **for** loop from lines 6 – 25 is entered. In this loop, SlotIter candidate schedules are created for the current slot V_M . First \hat{V} , the set of admissible unscheduled stations is initialized to V' and U , the set of inadmissible scheduled vertices is initialized to the empty set. S , the set of stations scheduled in the

```

procedure SlotMinimization( $G', \alpha, \text{SlotIter}$ )
1   $M \leftarrow 0$ 
2   $V' \leftarrow V$ 
3  while  $V' \neq \emptyset$  do
4     $M \leftarrow M + 1$ 
5     $Ecount \leftarrow \infty$ 
6    for  $j = 1$  to  $\text{SlotIter}$  do
7       $\hat{V} \leftarrow V', U \leftarrow \emptyset, S \leftarrow \emptyset$ 
8      while  $\hat{V} \neq \emptyset$  do
9        if  $U = \emptyset$  then
10          $\text{RCL} \leftarrow \{(1 - \alpha)100\% \text{ stations of max degree in } \hat{V}\}$ 
11        else
12          $\text{RCL} \leftarrow \{(1 - \alpha)100\% \text{ stations of max degree in } \hat{V} \cap U\}$ 
13        end if
14         $s \leftarrow \text{randSelect}(\text{RCL})$ 
15         $S \leftarrow S \cup \{s\}$ 
16         $N(s) \leftarrow \{w | (s, w) \in E'\}$ 
17         $\hat{V} \leftarrow \hat{V} / (\{s\} \cup N(s))$ 
18         $U \leftarrow U \cup N(s)$ 
19      end while
20       $E^* \leftarrow \{(u, v) \in E' | u, v \in V' / S\}$ 
21      if  $|E^*| < Ecount$  then
22         $V_i \leftarrow S$ 
23         $Ecount \leftarrow |E^*|$ 
24      end if
25    end for
26     $V' \leftarrow V' / V_i$ 
27  end while
28  return  $(M, V^* = \{V_1, V_2, \dots, V_M\})$ 
end procedure SlotMinimization

```

FIGURE 2. Greedy randomized heuristic for frame length minimization [13].

current slot, is also set to \emptyset . From lines 9 – 11 a so-called Restricted Candidate List (RCL) is constructed and contains the $(1 - \alpha)100\%$ admissible stations of maximum degree. It is now clear how the particular value of α controls the amount of randomness that is used by the algorithm. A value of $\alpha = 0$ would result in a simple random search, while $\alpha = 1$ would yield a pure greedy search [15]. After the construction of the RCL, an element $s \in \text{RCL}$ is chosen at random and scheduled in the current slot in line 15. The sets \hat{V} and U are updated and the loop continues. After the slot capacity is maximal, the set E^* is computed which contains the set of edges remaining in the graph induced by the yet unscheduled stations. If $|E^*|$ is less than the current minimum value $Ecount$, then the current candidate slot schedule is saved in V_M in line 22. In line 26 after SlotIter samples, the best slot schedule is removed from the graph and the main loop repeats [13]. Finally, the frame length M and the final slot schedule $V^* = \{V_1, V_2, \dots, V_M\}$ are returned to the main procedure.

The result of this procedure is a feasible solution for MSP-TDMA in which each station is scheduled to broadcast in exactly one slot during the frame. This follows directly from the result proven in Theorem 1. For a discussion of the computational complexity of the proposed procedure, the reader is referred to [13].

3.2. Throughput Maximization Phase. The second phase of the proposed method attempts to maximize the throughput beginning with the feasible solution found in the frame length minimization phase. Clearly, the solution from the first phase will not provide an

```

procedure BurstMaximization( $G', M^*, V^*$ )
1   $X \leftarrow |V|$ 
2  for  $i = 1$  to  $M$  do
3     $T \leftarrow V$ 
4    while  $T \neq \emptyset$  do
5       $T \leftarrow \{v | v \notin V_i \text{ and } \forall s \in V_i, (v, s) \notin E'\}$ 
6       $s \leftarrow \text{randSelect}(T)$ 
7       $V_i \leftarrow V_i \cup \{s\}$ 
8       $X \leftarrow X + 1$ 
9    end while
10 end for
11 return  $(X, V^* = \{V_1, V_2, \dots, V_M\})$ 
end procedure BurstMaximization

```

FIGURE 3. Throughput maximization pseudocode.

optimal throughput in general, because each station will only be scheduled to transmit once in the frame. Therefore, we use a randomized local improvement method to schedule each station as many times as possible in the frame.

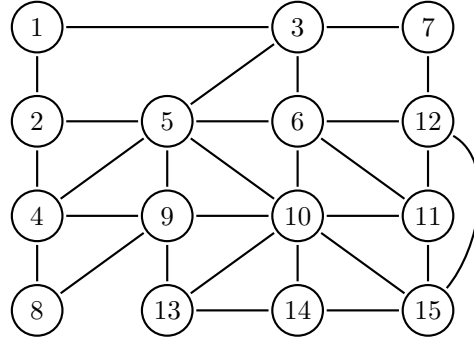
Pseudocode for the throughput maximization heuristic is provided in Figure 3, and the method proceeds as follows. Since each station is only scheduled once, the total number of bursts, X is set to $|V|$. The main loop from lines 2 – 10 locally optimizes each slot in the frame. First, the set of stations which can transmit along with those stations already scheduled in the current slot, namely T is initialized to V . T is then updated and contains those stations v which are not already scheduled in the current slot and are not adjacent to any station s which is scheduled in the current slot. An element of T is then selected randomly and added to the current slot. In line 8 the total number of bursts is incremented, and the loop repeats. The method proceeds to the next slot when there are no stations which can transmit with those currently scheduled, i.e., when $T = \emptyset$. The method returns the total number of bursts, X and the updated broadcast schedule V^* in line 11.

4. COMPUTATIONAL RESULTS

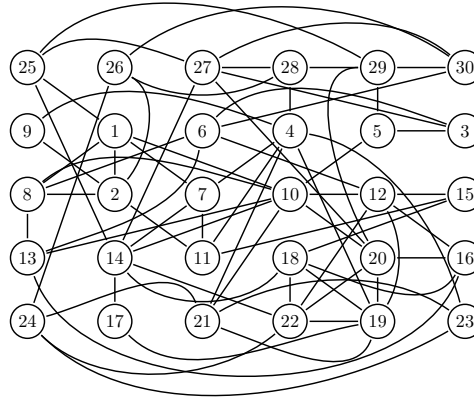
The proposed heuristic was coded in the C++ programming language and compiled using Microsoft® Visual C++ 6.0. The test machine was a PC equipped with a 1700MHz Intel® Pentium® M processor and 1GB of RAM operating under the Microsoft® Windows® XP environment. The heuristic was tested on three classical instances as well as a set of 60 random unit disk graphs [2] with varying densities, 20 graphs each having 50, 75, and 100 nodes. The graphs are those which were used by Butenko et al. in a prior MSP-TDMA study [3, 4].

We compared our results to those found by several heuristics from the literature, all of which were tested on the same PC described above. As for the parameters of our method, α was set to 0.1 and SlotIter to 5. In addition, we have implemented the integer programming (IP) model for the THROUGHPUT MAXIMIZATION PROBLEM using the Xpress-MP™ optimization suite from Dash Optimization [10]. Xpress-MP contains an implementation of the simplex method [8], and uses a branch and bound algorithm [20] together with advanced cutting-plane techniques [9, 14]. Thus not only are we able to compare our heuristic to those in the literature, but we can also see how the heuristics compare with the optimal solutions.

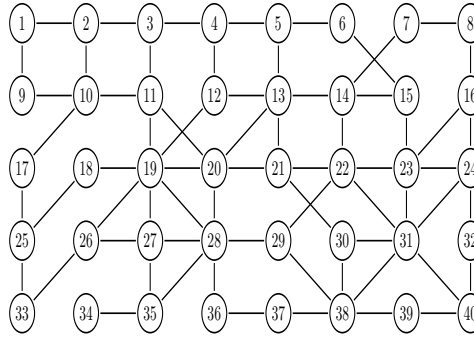
Though finding the optimal frame length is \mathcal{NP} -hard, we can use the IP model for the TMP to confirm whether a frame length is optimal or not. Consider an instance of MSP-TDMA and let M^* be the optimal frame length. Then if we set $M = M^* - 1$ in the integer programming model for the TMP, the resulting IP will be not yield any feasible



(a)



(b)



(c)

FIGURE 4. (a) 15 station network. (b) 30 station network. (c) 40 station network.

integer solutions. In fact, the linear programming relaxation could also be infeasible; thus implying the particular instance of the TMP is also infeasible. The proposed heuristic was first tested using three examples first introduced by Wang et al. in [19] which have become the de facto test cases for TDMA broadcast scheduling algorithms. These examples include networks of varying densities with 15, 30, and 40 stations. The graphs of the networks can be seen in Figure 4.

| Stations | Optimal Soln | ComAlg | GRASP | HNN-GA | MFA | SVC |
|----------|--------------|---------|---------|---------|---------|---------|
| 15 | (20,8) | (20,8) | (20,8) | (20,8) | (18,8) | (18,8) |
| 30 | (36,10) | (36,10) | (36,10) | (35,10) | (39,12) | (37,11) |
| 40 | (69,8) | (69,8) | (65,8) | (67,8) | (71,9) | (60,8) |

TABLE 1. Comparison of solutions for the benchmark instances from [19].

Table 1 provides the optimal solutions for the three aforementioned networks as well as the heuristic solutions found by our combinatorial algorithm (ComAlg), the GRASP from [3], the Mixed Neural-Genetic Algorithm (HNN-GA) proposed in [18], the Mean Field Annealing (MFA) method from Wang and Ansari [19], and the Sequential Vertex Coloring (SVC) heuristic from [21]. The solutions are reported as (X, M) . Notice that the proposed algorithm found the optimal solution for each of the three instances. The average computation time required for these instances by our method was 1.375s. The average time required by Xpress-MP to compute the optimal solutions was 3411.4 seconds, with the 30 station network taking 10212 seconds. Next, in order to test the scalability of the new method and evaluate its performance for general networks, we tested the algorithms on the 60 random graphs from [3].

The comparative results of the proposed algorithm against the best solutions computed by Xpress-MP after 3600 seconds, as well as the aforementioned heuristics¹ on the 50 station graphs from [3] are given in Table 2. The first column represents the instance name followed by the density of the graph G' . Notice that the solutions from the new method (ComAlg) are at least as good as any other heuristic for all of these instances. Specifically, the new method provides better solutions for 15 of the 20 instances. The asterisk implies that the reported solution is optimal. For these instances, the new algorithm found optimal solutions for 40% of the test cases. The average frame utilization is also reported at the bottom of the table. The utilization ρ , provides a measure of the efficiency of a broadcast schedule and is computed as follows

$$\rho = \frac{X}{M \cdot |V|}. \quad (7)$$

We see that for the 50 station networks, the proposed algorithm has an average channel utilization that is 10.96% greater than the other heuristics. The average optimality gap for the throughput maximization phase was 1.921%. The average computation time for our algorithm on these instances was 2.8 seconds.

The comparative solutions for the 75 station networks are given in Table 3. Notice that our method outperforms the other heuristics in the literature on every instance. For these networks, the proposed algorithm has an average channel utilization that is 8.66% greater than the other methods. The heuristic required on average 6.62 seconds to find the target solution, and as with the 50 station networks, optimal frame lengths are achieved for all instances. Furthermore, the solutions from our method are always within 10% of optimal solutions, with an average gap of 5.8%.

Finally, the solutions for the 100 station networks are given in Table 4. Once more, the new algorithm finds solutions which are superior to the other heuristics for each instance. The utilization was an average of 10.17% higher than the other algorithms. The average computation time was 12.17 seconds, with reported gaps of less than 10% of the best solution found by Xpress-MP after 3600 seconds. Notice also that Xpress-MP was unable to compute a solution superior to the proposed heuristic for 100r50i3 and 100r50i6.

¹The MFA algorithm of [19] was not available to the authors for testing.

| Instance | Density | Xpress-MP | ComAlg | GRASP | HNN-GA | SVC |
|----------|---------|----------------------|----------------------|--------------|--------------|-------------|
| 50r20i6 | 0.1136 | (146,10) | (145,10) | (143,10) | (145,10) | (111,10) |
| 50r20i2 | 0.0824 | (86,6) | (86,6)* | (84,6) | (86,6)* | (82,6) |
| 50r20i3 | 0.1040 | (85,6) | (85,6)* | (83,6) | (85,6)* | (60,7) |
| 50r20i7 | 0.0872 | (90,6) | (89,6) | (87,6) | (89,6) | (52,6) |
| 50r20i5 | 0.0968 | (107,7) | (107,7)* | (107,7)* | (105,7) | (64,8) |
| 50r30i1 | 0.1728 | (78,8) | (76,8) | (74,8) | (75,8) | (54,9) |
| 50r30i2 | 0.2122 | (77,9) | (75,9) | (81,10) | (70,9) | (73,10) |
| 50r30i3 | 0.1960 | (84,9) | (84,9)* | (78,9) | (78,9) | (78,10) |
| 50r30i4 | 0.2048 | (74,8) [†] | (71,8) | (67,8) | (67,8) | (60,10) |
| 50r30i5 | 0.2096 | (82,9) | (79,9) | (76,9) | (84,10) | (89,11) |
| 50r40i1 | 0.3048 | (76,12) | (74,12) | (73,12) | (71,12) | (58,14) |
| 50r40i2 | 0.3680 | (83,14) [†] | (80,14) | (77,14) | (77,14) | (83,16) |
| 50r40i3 | 0.3408 | (76,12) [†] | (76,12)* | (80,13) | (77,13) | (56,15) |
| 50r40i4 | 0.3712 | (81,15) | (81,15)* | (80,15) | (76,15) | (81,17) |
| 50r40i5 | 0.3208 | (71,12) | (70,12) | (67,12) | (65,12) | (55,14) |
| 50r50i1 | 0.4280 | (72,17) | (72,17)* | (71,17) | (75,18) | (61,19) |
| 50r50i2 | 0.4640 | (61,15) [†] | (61,15) [†] | (65,16) | (68,17) | (55,17) |
| 50r50i3 | 0.4480 | (66,15) [†] | (66,15) [†] | (64,15) | (65,16) | (56,17) |
| 50r50i4 | 0.4376 | (70,15) | (70,15)* | (72,16) | (72,16) | (79,18) |
| 50r50i5 | 0.4088 | (55,14) [†] | (55,14) [†] | (58,15) | (56,15) | (61,18) |
| Avg Soln | 0.2603 | (80.1,10.95) | (80.1,10.95) | (79.35,11.2) | (79.3,11.35) | (68.4,12.6) |
| Avg Util | – | 0.1463 | 0.1463 | 0.1417 | 0.1397 | 0.1086 |

TABLE 2. Comparison of optimal and heuristic solutions for graphs with $|V| = 50$ stations. An * indicates that the solution is optimal, while a [†] indicates the solution is the best found by Xpress-MP after 3600s. Solutions are reported as (X, M) .

Each of these instances were ran for 10000 seconds and Xpress was unable to compute a feasible solution in the frame length achieved by our method.

Experimental analysis shows that our algorithm is superior to the other heuristics in the literature. For all 63 instances tested, the method found solutions at least as good as any of the other algorithms from the literature for all of the networks, outperforming them on 56 cases. Also, we see that attempting to solve large-scale instances optimally is impractical. However, our heuristic required only 7.49 seconds on average to find solutions that are within 4.18% of the average best solution found by the commercial IP solver in 3600 seconds.

5. CONCLUSION

In this paper, we described and implemented a new heuristic for the MESSAGE SCHEDULING PROBLEM ON TDMA NETWORKS. The MSP-TDMA is an important problem that occurs in wireless mesh networks regarding efficiently scheduling collision free broadcasts for the network stations. The objective of the MSP-TDMA is two-fold. First, the number of slots required to schedule all stations is minimized. Then the throughput is to be maximized by scheduling as many stations as possible with the determined frame length. The algorithm exploits the combinatorial structure of the problem in order to quickly find high-quality solutions.

The proposed algorithm is executed in two phases, each handling one of the objectives in the IP formulation. The first phase finds a minimal feasible frame length, and the second phase maximizes the throughput within this frame. Experimental results indicate that our heuristic outperforms several other methods from the literature. In addition, the method is robust in that it is able to find good solutions for a wide variety of instances. The efficiency

| Instance | Density | Xpress-MP | ComAlg | GRASP | HNN-GA | SVC |
|----------|---------|-----------------------|---------------|----------------|------------|---------------|
| 75r20i1 | 0.0988 | (145,8) | (139,8) | (135,8) | (136,8) | (161,10) |
| 75r20i2 | 0.1038 | (122,8) [†] | (119,8) | (113,8) | (112,8) | (79,10) |
| 75r20i3 | 0.1159 | (113,7) [†] | (108,7) | (139,9) | (121,8) | (150,10) |
| 75r20i4 | 0.0946 | (116,7) | (114,7) | (109,7) | (111,7) | (84,8) |
| 75r20i5 | 0.0988 | (145,8) | (138,8) | (131,8) | (135,8) | (161,10) |
| 75r30i1 | 0.1927 | (114,12) | (110,12) | (117,13) | (117,13) | (91,13) |
| 75r30i2 | 0.1867 | (110,11) | (105,11) | (109,12) | (101,11) | (94,12) |
| 75r30i3 | 0.2190 | (140,15) | (133,15) | (132,15) | (132,15) | (81,17) |
| 75r30i4 | 0.2009 | (142,13) [†] | (133,13) | (127,13) | (128,13) | (144,15) |
| 75r30i5 | 0.1927 | (119,12) [†] | (111,12) | (106,12) | (108,12) | (89,12) |
| 75r40i1 | 0.3328 | (105,17) [†] | (103,17) | (104,18) | (113,19) | (79,20) |
| 75r40i2 | 0.2980 | (108,16) [†] | (106,16) | (109,17) | (115,18) | (86,19) |
| 75r40i3 | 0.3403 | (112,19) [†] | (109,19) | (105,19) | (103,19) | (87,20) |
| 75r40i4 | 0.3492 | (126,20) [†] | (118,20) | (124,21) | (119,21) | (79,24) |
| 75r40i5 | 0.3143 | (104,16) [†] | (97,16) | (100,17) | (109,18) | (114,20) |
| 75r50i1 | 0.4587 | (110,23) [†] | (106,23) | (107,24) | (108,25) | (123,29) |
| 75r50i2 | 0.4622 | (102,23) [†] | (97,23) | (99,24) | (104,25) | (108,27) |
| 75r50i3 | 0.4807 | (106,24) [†] | (102,24) | (104,25) | (111,27) | (114,29) |
| 75r50i4 | 0.4750 | (121,26) [†] | (115,26) | (112,26) | (110,26) | (102,28) |
| 75r50i5 | 0.5088 | (106,25) [†] | (104,25) | (111,27) | (107,27) | (106,28) |
| Avg Soln | 0.2686 | (118.45,15.5) | (113.25,15.5) | (114.65,16.15) | (115,16.4) | (106.6,18.05) |
| Avg Util | — | 0.1019 | 0.0974 | 0.0947 | 0.0935 | 0.0787 |

TABLE 3. Comparison of optimal solver and heuristic solutions for the 75 station networks.

| Instance | Density | Xpress-MP | ComAlg | GRASP | HNN-GA | SVC |
|----------|---------|-----------------------|---------------|---------------|---------------|---------------|
| 100r20i1 | 0.1006 | (204,11) [†] | (184,11) | (179,11) | (181,11) | (110,11) |
| 100r20i2 | 0.1028 | (168,9) [†] | (159,9) | (169,10) | (149,9) | (131,11) |
| 100r20i3 | 0.1138 | (208,12) [†] | (187,12) | (177,12) | (182,12) | (101,14) |
| 100r20i4 | 0.0992 | (172,9) | (160,9) | (171,10) | (153,9) | (161,10) |
| 100r20i5 | 0.1028 | (183,10) [†] | (170,10) | (183,11) | (164,10) | (150,12) |
| 100r30i1 | 0.1942 | (138,14) [†] | (132,14) | (148,16) | (137,15) | (119,16) |
| 100r30i2 | 0.1956 | (158,15) [†] | (147,15) | 1 (152,16) | (135,15) | (145,17) |
| 100r30i3 | 0.2270 | (168,16) [†] | (157,16) | (164,18) | (159,17) | (138,18) |
| 100r30i4 | 0.2172 | (156,16) [†] | (143,16) | (140,17) | (137,16) | (102,17) |
| 100r30i5 | 0.2088 | (139,14) [†] | (130,14) | (151,17) | (132,16) | (150,17) |
| 100r40i1 | 0.3552 | (103,21) [†] | (121,21) | (136,24) | (141,25) | (142,26) |
| 100r40i2 | 0.3300 | (157,23) [†] | (145,23) | (149,24) | (145,24) | (141,25) |
| 100r40i3 | 0.3480 | (192,28) [†] | (183,28) | (178,28) | (171,28) | (115,31) |
| 100r40i4 | 0.3164 | (168,24) [†] | (156,24) | (148,24) | (143,24) | (153,26) |
| 100r40i5 | 0.3178 | (151,22) [†] | (141,22) | (149,24) | (147,24) | (134,25) |
| 100r50i1 | 0.4826 | (130,29) [†] | (127,29) | (142,33) | (146,34) | (104,36) |
| 100r50i2 | 0.4604 | (142,32) [†] | (138,32) | (137,32) | (145,34) | (148,36) |
| 100r50i3 | 0.5076 | (109,35) [†] | (148,34) | (149,35) | (147,36) | (138,38) |
| 100r50i4 | 0.4690 | (129,30) [†] | (125,30) | (128,32) | (133,33) | (126,33) |
| 100r50i6 | 0.5012 | (145,32) [†] | (129,30) | (149,34) | (150,35) | (125,35) |
| Avg Soln | 0.2825 | (156, 20.57) | (149.1,19.95) | (155.95,21.4) | (150.3,21.35) | (131.65,22.7) |
| Avg Util | — | 0.0758 | 0.0747 | 0.0729 | 0.0704 | 0.0580 |

TABLE 4. Comparison of optimal solver and heuristic solutions for networks with $|V| = 100$ stations.

of the method is demonstrated by comparing the run times to that required by a commercial integer programming solver.

REFERENCES

- [1] I.F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.
- [2] B.N. Clark, C.J. Colbourn, and D.S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.
- [3] C.W. Commander, S.I. Butenko, and P.M. Pardalos. On the performance of heuristics for broadcast scheduling. In D. Grundel, R. Murphey, and P. Pardalos, editors, *Theory and Algorithms for Cooperative Systems*, pages 63–80. World Scientific, 2004.
- [4] C.W. Commander, S.I. Butenko, P.M. Pardalos, and C.A.S. Oliveira. Reactive grasp with path relinking for the broadcast scheduling problem. In *Proceedings of the 40th Annual International Telemetry Conference*, pages 792–800, 2004.
- [5] A. Ephremides and T.V. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, 38(4):456–460, 1990.
- [6] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- [7] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34:910–918, 1988.
- [8] F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*. McGraw Hill, 2001.
- [9] R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to Global Optimization*, volume 3 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, 1995.
- [10] Dash Optimization Inc. *Xpress-Optimizer Reference Manual*, 2003.
- [11] L. Kleinrock and J. Silvester. Spatial reuse in multihop packet radio networks. In *Proceedings of the IEEE* 75, 1987.
- [12] M. Krivelevich. Sparse graphs usually have exponentially many optimal colorings. *Electronic Journal of Combinatorics*, 9:8pp, 2002.
- [13] M. Laguna and R. Martí. A grasp for coloring sparse graphs. *Computational Optimization and Applications*, 19(2):165–178, 2001.
- [14] C.A.S. Oliveira, P.M. Pardalos, and T.M. Querido. A combinatorial algorithm for message scheduling on controller area networks. *Int. Journal of Operations Res.*, 1(1/2):160–171, 2005.
- [15] L.S. Pitsoulis and M.G.C. Resende. Greedy randomized adaptive search procedures. In M.G.C. Resende and P.M. Pardalos, editors, *Handbook of Applied Optimization*, pages 168–183. Oxford University Press, 2002.
- [16] M.G.C. Resende and P.M. Pardalos. *Handbook of Optimization in Telecommunications*. Springer, 2006.
- [17] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer Academic Publishers, 2003.
- [18] S. Salcedo-Sanz, C. Busoño-Calzón, and A.R. Figueiral-Vidal. A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Transactions on Wireless Communications*, 2(2):277–283, 2003.
- [19] 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):250–260, 1997.
- [20] L. Wolsey. *Integer Programming*. Wiley, 1998.
- [21] J. Yeo, H. Lee, and S. Kim. An efficient broadcast scheduling algorithm for tdma ad-hoc networks. *Computers and Operations Research*, 29:1793–1806, 2002.

(C.W. COMMANDER) AIR FORCE RESEARCH LABORATORY, MUNITIONS DIRECTORATE, AND, DEPT. OF INDUSTRIAL AND SYSTEMS ENGINEERING, UNIVERSITY OF FLORIDA, GAINESVILLE, FL 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, USA.
E-mail address: pardalos@ufl.edu