A Greedy Randomized Adaptive Search Procedure for the Broadcast Scheduling Problem

Clayton W. Commander∗️ Sergiy I. Butenko†
Panos M. Pardalos∗️

Abstract

In the Broadcast Scheduling Problem (BSP), a finite set of wireless stations are to be scheduled in a time division multiple access frame. The objective is to provide a collision free broadcast schedule which minimizes the total frame length and maximizes the slot utilization within the frame. We present a Greedy Randomized Adaptive Search Procedure (GRASP) for the BSP. GRASP is a two-phase metaheuristic for combinatorial optimization. In the first phase, a greedy randomized initial feasible solution is created. The initial solution is then improved by the application of a local search procedure. Numerical examples are given and results of the GRASP are compared with those of other scheduling algorithms.

1 Introduction

In recent years there has been an explosion of research in the area of wireless communication. This is due to improved technology and increasing demands. Of particular interest are so-called ad-hoc networks. Having no fixed infrastructure, these networks are particularly useful in areas such as mobile commerce, combat search and rescue, and other battlefield scenarios, to name a few [3]. There are however inherent difficulties with ad-hoc networks, mainly concerning message scheduling and routing.

Since all stations in the network share the transmission channel, they must be scheduled to transmit messages in such a manner that prevents destructive interference, or message collision. There are two types of message collision. The first, referred to as direct collision occurs when two neighboring stations transmit during the same time slot. The second, known as

∗️Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida. 303 Weil Hall, Gainesville, FL 32611-6595. Email: {clayton8,pardalos}@ufl.edu
†Department of Industrial Engineering, Texas A&M University. 236E Zachry Engineering Center, College Station, FL 77840-3131. Email: butenko@tamu.edu
hidden collision results when two non-neighboring stations transmit simultaneously to a station that can receive messages from both senders. The desired result is a schedule that is guaranteed to produce collision free transmissions [24].

In [14], it is shown that the time division multiple access (TDMA) protocol can be used to provide a collision free scheduling procedure. In a TDMA network, time is divided into frames, each frame consisting of a number of fixed length slots. It is acceptable, and in fact highly desirable for multiple stations to transmit during the same time slot provided that they do not cause any collision, either hidden or direct. Exact transmission criteria will be defined in Section 2.

In this paper, we present a Greedy Randomized Adaptive Search Procedure (GRASP) for the Broadcast Scheduling Problem (BSP). Our GRASP attempts to minimize the total frame length of the broadcast schedule and maximize the throughput within the frame. Such a schedule will minimize the overall delay of the system. The organization is as follows: In Section 2, we will formally define the problem statement. Section 3 will contain a brief review of GRASP and a description of the GRASP for the BSP. Computational results are compared with other algorithms in Section 4. Finally, some concluding remarks are given in Section 5.

2 The Broadcast Scheduling Problem

An ad-hoc network can be 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. Two stations $i, j \in V$ are said to be one-hop neighbors if and only if (iff) they can directly communicate. That is, stations $i$ and $j$ are one-hop neighbors iff there exists an undirected edge $(i, j) \in E$. One-hop neighbors transmitting in the same slot will result in a direct collision. If $(i, j) \notin E$ but there exists an intermediate node $k \in V$ such that $(i, k) \in E$ and $(k, j) \in E$, then we say that stations $i$ and $j$ are two-hop neighbors. Two-hop neighboring stations which transmit in the same time slot will cause a hidden collision.

The topology of the network can then be described by an $N \times N$ symmetric binary matrix $C$, where $N = |V|$. $C = \{c_{ij}\}$ is known as the connectivity (adjacency) matrix and is defined as follows:

$$c_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E \text{ and } i \neq j, \\ 0, & \text{otherwise.} \end{cases}$$

We assume that there are $M$ time slots per frame, and that one slot length in time is equal to the time required to transmit one packet of information. We also assume that packets are transmitted at the beginning of each slot, and that the message is received during the same slot in which it
is sent. It is now possible to represent the broadcast schedule as a binary 
$M \times N$ matrix $S = \{s_{mn}\}$, where

$$s_{mn} = \begin{cases} 1, & \text{if station } n \text{ is scheduled to transmit in slot } m, \\ 0, & \text{otherwise.} \end{cases}$$

In order to perform some analysis on the efficiency of a schedule, it is 
helpful to know what percentage of the available slots are being assigned in 
a transmission frame. Let $\rho_n$ be the slot utilization for station $n$. Then,

$$\rho_n = \frac{\text{the number of slots assigned to station } n}{\text{frame length}} = \frac{\sum_{m=1}^{M} s_{mn}}{M}.$$

It follows that $\rho$, the total slot utilization of the network is as follows:

$$\rho = \frac{\sum_{n=1}^{N} \rho_n}{N} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} s_{mn}}{NM}.$$

With this, we may define the Broadcast Scheduling Problem as follows:

Minimize $M$ and Maximize $\rho$

subject to:

$$\sum_{m=1}^{M} s_{mn} \geq 1, \quad \forall n, \quad (1)$$

$$c_{ij} + s_{mi} + s_{mj} \leq 2, \quad \forall i, \forall j, \text{ and } \forall m, i \neq j, \quad (2)$$

$$c_{ik}s_{mi} + c_{kj}s_{mj} \leq 1, \quad \forall i, \forall j, \forall k, \text{ and } \forall m, i \neq j, j \neq k, k \neq i. \quad (3)$$

The first constraint implies that each station should be able to transmit 
at least once per frame. Constraint (2) ensures that one-hop neighbors will 
not transmit in the same slot. Finally, constraint (3) ensures that no two-hop 
neighbors will transmit in the same slot [24].

The recognition version of the BSP was proven to be $NP$-complete by 
the current authors in [4]. This implies that an algorithm which optimally 
solves the problem in polynomial time is unlikely to exist. Thus the need 
for heuristics which provide high quality solutions within reasonable com-
putation times arises.

It is possible to establish some lower bound for $M$ which can provide some 
insight about the minimum number of slots which will be required for a given 
broadcast schedule. In [23], Wang and Ansari propose a lower bounding
lemma based on the degrees of the vertices in the graph. Specifically, for a given network, \( G = (V, E) \), define the degree of a given vertex \( v \in V \), denoted \( \text{deg}(v) \), to be the number of edges incident to \( v \). Then, the frame length \( M \) satisfies the following inequality:

\[
M \geq \delta(G) + 1, \tag{4}
\]

where \( \delta(G) = \max_{v \in V} \text{deg}(v) \).

Though the bound in (4) is relatively easy to calculate, it does not provide a tight lower bound for \( M \). In [13], another bounding procedure is given which produces tighter bounds on \( M \). Consider the network \( G = (V, E) \) as previously defined. That is, \( V \) is the set of stations, and \( E \) is the set of direct collisions. Suppose now that \( E \) is expanded to also include hidden collisions, and call this new set \( E' \). We now have a new graph, namely \( G' = (V, E') \).

Recall from graph theory that a clique is a subset \( C \subseteq V \) such that any two vertices in \( C \) are adjacent. That is, \( C \) is a complete subgraph. With that, Jungnickel shows that

\[
M \geq \omega(G'), \tag{5}
\]

where \( \omega(G') \) is the maximal cardinality of a clique in \( G' \). This bound will be used later to show that GRASP obtains optimal frame lengths for all test cases examined.

3 A GRASP for the BSP

Greedy Randomized Adaptive Search Procedure (GRASP), originally introduced by Feo and Resende in [7], is a two-phase multi-start metaheuristic for hard combinatorial problems [8, 9, 20]. In the first phase, known as the construction phase, an initial solution is built by means of an adaptive greedy function. However, since a construction phase solution is not guaranteed to be locally optimal, phase two applies a local search for improvement. The best solution out of all GRASP iterations is returned. GRASP has been applied to many combinatorial problems such as quadratic assignment [16, 17], job shop scheduling [2, 1], and most recently the uncapacitated facility location problem [21].

3.1 Construction Phase

In the construction phase, a basic feasible solution is constructed iteratively, one element at a time. First, the stations in the network are sorted in descending order of the number of one-hop and two-hop neighbors. Then that station which has the most neighbors is assigned. After this greedy selection is made, a restricted candidate list (RCL) is formed consisting of those elements which may transmit simultaneously with the station assigned
by the greedy function. Next, a station is selected at random from the RCL
and is assigned in the same slot as the greedy assigned station. A new RCL
is then created, and another station is selected at random. This process
continues until RCL = ∅. At this point, the GRASP moves to the next slot
and restarts the process beginning with another biased greedy assignment.
This bias is towards those stations which were not previously selected at
random from the RCL.

3.2 Local Search Phase

For the local search phase of the GRASP for the BSP, we implement a simple
swap-based procedure. This is an adaptation of the method introduced by
Laguna and Martí in [15]. Using the initial solution produced by the greedy
randomized constructor as a guide, a new broadcast schedule is created.
First, the slots are sorted in descending order of the number of transmissions.
The two slots with the fewest number of bursts are then combined and the
number of slots becomes \( k = m - 1 \). Denote the new broadcast schedule as
\( S_{m',n} \). Now, let the function

\[
f(s) = \sum_{i=1}^{k} E(m'_i),
\]

where \( E(m'_i) \) is the set of collisions in slot \( m'_i \). \( f(s) \) is then minimized
by the application of a local search procedure as follows. A colliding station in
the combined slot is chosen randomly and every attempt is made to swap
this station with another from the remaining \( k - 1 \) slots. After a swap is
made, \( f(s) \) is re-evaluated. If the result is better, that is if \( f(s) \) has a lower
value than before the swap, the swap is kept and the process repeated with
the remaining colliding stations. If after every attempt to swap a colliding
station the result is unimproved, a new colliding station is chosen and the
swap routine is attempted. This continues until either a succesful swap is
made or for some specified number of iterations.

If a solution is improved such that \( f(s) = 0 \), then the frame length has
been successfully decreased by one slot. The value of \( k \) is then decremented
and the process is repeated beginning with the combination of the two small-
est slots. If the procedure ends with \( f(s) > 0 \), then no improved solution
was found.

If the procedure is able to reduce the frame length to some new slot count
\( k < m \), then the main objective of minimizing the frame length has been
achieved. However, there is still a desire to have an increased throughput.
Furthermore, there still remains a possibility that more stations can be
scheduled in the reduced slots. Therefore, after the completion of the local
search
4 Computational Results

The GRASP was tested using three examples first introduced by Wang and Ansari in [23] which have become the de facto test cases for broadcast scheduling algorithms. These examples include 15, 30, and 40 station networks with varying densities. The graphs of the networks can be seen in Figure 1.

4.1 Average Time Delay

In [23], the average time delay of a network and the channel utilization are used as ways of rating the performance of broadcast scheduling heuristics. These methods were also adopted by Yeo, et al. in [24]. For the sake of comparison, we will also use these parameters as a means of evaluating the effectiveness of the GRASP. The following assumptions were made before deriving the average time delay:

1. The interarrival time for each station \( i \) is statistically independent from other stations, and packets arrive according to a Poisson process with a rate of \( \lambda_i \) (packets/slot). The total traffic in station \( i \) consists of its own traffic and the traffic incoming from other stations. Packets are stored in buffers in each station and the buffer size is infinite.

2. The probability distribution of the service time of station \( i \) is deterministic. Define the service rate of station \( i \) to be \( \mu_i \) (packets/slot).

Using these assumptions and those from Section 2, an ad-hoc network can be modeled as a system of \( N \) M/D/1 queues, where \( N \) is the number of stations in the network. The Pollaczek-Khinchin (P-K) formula is used to determine the average time delay of each queue. Letting \( D_i \) represent the average time delay for each station \( i \), then by P-K, we have:

\[
D_i = \frac{1}{\mu_i} + \frac{\lambda_i/\mu_i^2}{2(1 - \rho_i)},
\]

where \( \mu_i = \sum_{m=1}^{M} s_{mi}/M \), and \( \rho_i = \lambda_i/\mu_i \). Thus, the total time delay is given by

\[
D = \frac{\sum_{i=1}^{N} \lambda_i D_i}{\sum_{i=1}^{N} \lambda_i}.
\]

4.2 Numerical Results

The results of the GRASP can be seen in the broadcast schedules shown in Figure 2. Specific information about these schedules is given in Figure 4. Note that in all examples, GRASP achieves schedules with optimal frame lengths of 8, 10, and 8 slots respectively. When compared to the algorithms
presented in [24] and [23], GRASP outperforms them both in terms of total channel utilization for all cases. Notice also from the graphs in Figure 3 that GRASP results in less time delay of queued messages. This would suggest that GRASP is the more efficient heuristic.
Figure 2: GRASP broadcast schedules from Figure 1: (a) 15 station network, (b) 30 station network, (c) 40 station network.
Figure 3: Comparison of average time delays for example networks: (a) 15 station network, (b) 30 station network, (c) 40 station network.
<table>
<thead>
<tr>
<th>Stations</th>
<th>LB</th>
<th>Frame Length</th>
<th>Channel Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GRASP</td>
<td>SVC</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>30</td>
<td>10</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>40</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure 4: Comparison of frame length and utilization achieved by GRASP, Sequential Vertex Coloring (SVC) [24], and Mean Field Annealing (MFA) [23]. The lower bounds (LB) were calculated using the method from [13] described in Section 2.

5 Conclusion

In this paper, we presented a Greedy Randomized Adaptive Search Procedure for the Broadcast Scheduling Problem. We showed that our results compete very well with other heuristics, producing broadcast schedules for several networks with optimal frame lengths. There is still a great deal of work which can be done to intensify the GRASP.

One procedure known as path relinking [11] is used to introduce a so-called memory to the GRASP. Path relinking is a post-optimization step, that enables GRASP to recall certain characteristics about “good” and “bad” solutions and uses these characteristics to build better solutions in future iterations. GRASP has also been implemented using parallel processors [18, 19], which usually results in improved solutions. Though the solution quality tends to increase when such intensifications are applied, it is usually the case of trading between solution quality and computation time. It is up to the designer to decide if such improvement methods are profitable for their individual problems.

References


