Chapter 1

ON THE PERFORMANCE OF HEURISTICS FOR BROADCAST SCHEDULING

Clayton W. Commander
Department of Industrial and Systems Engineering,
University of Florida
Gainesville, FL
clayton8@ufl.edu

Sergiy I. Butenko
Department of Industrial Engineering
Texas A&M University
College Station, TX
butenko@tamu.edu

Panos M. Pardalos
Center for Applied Optimization
Department of Industrial and Systems Engineering
University of Florida
Gainesville, FL
pardalos@ufl.edu

Abstract In the Broadcast Scheduling Problem (BSP), a finite set of stations are to be scheduled in a time division multiple access (TDMA) frame. In a TDMA frame, time is divided into equal length transmission slots. Unconstrained message transmission can result in a collision of messages, rendering them useless. Therefore, the objective of the BSP is to provide a collision free broadcast schedule which minimizes the total frame length and maximizes the slot utilization within the frame. In this chapter, we introduce the BSP, show that it is *NP*-complete, and discuss several heuristics which have been applied to the problem. The heuristics are tested on over 60 networks of varying sizes and densities and the results are compared.

Keywords: Broadcast Scheduling Problem, Ad-hoc Networks, Combinatorial Optimization, *NP*-complete, Heuristics

1. Introduction

In recent years, ad-hoc networks have become increasingly popular among many researchers. These networks are used to apply a packet switching technique over a shared radio channel and provide high-speed communications between a large number of potentially mobile stations which may be geographically disbursed. Each station 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 in the network share the transmission channel, stations must be scheduled to transmit messages in such a manner that prevents message collisions [32]. 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. What is desired is a broadcast schedule that is guaranteed to produce collision free transmissions.

In [16], 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 unit length slots. The goal is to schedule as many stations as possible in the same time slot provided that simultaneous broadcasts will not cause any collision, either hidden or direct. Specific transmission criteria will be defined in Section 2.

In general, two optimization objectives are the focus of most broadcast scheduling research [32]. The first objective is to minimize the total number of slots in each frame. The other objective is, for a given frame length, to maximize the slot usage per frame, which will maximize the throughput [30, 32].

In this chapter, we introduce and formally define the BSP. Our primary objective is to minimize the frame length. In Section 2, we prove that the recognition version of this problem is NP-complete and discuss some lower bounding techniques. Section 4 will be a review of several heuristics which have been applied to the problem. Finally, some concluding remarks will be given in Section 5.

2. The Broadcast Scheduling Problem

An ad-hoc TDMA network can be described as a graph $G = (V, E)$ where the vertices $V = \{1, \ldots, n\}$ represent the stations in the network and the edge set E represents the set of transmission links between the vertices. 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 which transmit in the same time slot will cause a direct collision. Now suppose that $(i, j) \notin E$ but there exists some intermediate node $k \in V$ such that $(i, k) \in E$ and $(k, j) \in E$, then we will say that stations i and j are *two-hop neighbors.* Two-hop neighboring stations transmitting in the same slot will result in a hidden collision. Let us donote by $C = \{c_{ij}\}$ the *adjacency (connectivity) matrix* of G, which 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 packet length in time is equal to the time required to transmit one packet of information. Furthermore, we assume that packets are transmitted at the beginning of each time slot, and that each packet is received during the same slot in which it is sent. With these assumptions, we now represent the broadcast schedule as a binary $M \times N$ matrix $S = \{s_{mn}\}$, where

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

When performing analysis on the efficiency of a broadcast schedule it is helpful to know what percentage of the available slots are utilized in each frame. Let ρ_n be the slot utilization for station n. Then,

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

Hence, the total slot utilization of the entire network, ρ, is given by

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

We now define the Broadcast Scheduling Problem as follows:

$$\text{Minimize } M \text{ and Maximize } \rho$$

subject to:

$$\sum_{m=1}^{M} s_{mn} \geq 1, \qquad \forall\, n, \tag{1}$$

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

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

The first constraint allows each station to broadcast at least once per frame. Constraint (2) ensures that one-hop neighbors do not transmit in the same slot. Lastly, constraint (3) ensures that no two-hop neighbors transmit in the same slot [32].

2.1. Computational Complexity

The BSP belongs to a class of computationally difficult problems known as *NP*-complete [9]. Belonging to this class suggests that an algorithm which solves the problem to optimality in polynomial time is unlikely to exist [9]. It was first noted that BSP is *NP*-complete by Wang & Ansari in [30]. However, their proof of NP-completeness of the recognition version of BSP was incorrect. Complexity results for other related problems can be found in [5] and [4]. Next we prove the NP-completeness of the recognition version of BSP. We consider the following problem (K-BSP): Given a graph $G = (V, E)$ and an integer K, does there exist a broadcast schedule with frame length $\leq K$? To show that K-BSP is NP-complete, we need to show that (1) K-BSP $\in NP$; (2) Some NP-complete problem reduces to K-BSP in polynomial time. Denote by $n = |V|$, $m = |E|$. Without loss of generality, we assume that G is connected (if it is not, we can consider each connected component separately).

(1) K-BSP $\in NP$ since a given broadcast schedule with frame length $k \leq K$ can be verified for validity in $O(n^3)$ time. Indeed, the verification of validity consists of checking, for each vertex i, 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 repeated elements. This can be done using the sorting of time slot numbers in L_i in $O((|L_i|+1)\log(|L_i|+1))$ time for vertex i, therefore the total run time will be

$$O\left(\sum_{i=1}^{n}(|L_i|+1)\log(n)\right) = O((m+n)\log n) = O(n^3).$$

(2) We will show that the graph k-coloring problem can be reduced to K-BSP in polynomial time. Recall that k-coloring problem 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 NP-complete problem [9].

Given a graph $G=(V,E)$, we will construct the corresponding graph $G'=(V',E')$, where $V'=V\bigcup E$ and $E'=\{[i,(i,j)] : (i,j)\in E, i,j\in V\}\bigcup\{(e_1,e_2) : e_1,e_2\in E\}$. 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', $(v_1,v_2)\in E$ *iff* v_1 and $v_2\in V$ 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 from 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'. Thus k-coloring reduces in polynomial time to K-BSP, where $K=k+m$.

2.2. Lower Bounds on the Frame Length

Establishing a lower bound on the frame length M is possible by means of a few lemmata provided in the literature, three of which will be described here. In [30], Wang and Ansari propose a lower bounding lemma based on the degrees of vertices in the graph. Specifically, for a given network, $G=(V,E)$, define the *degree* of a given vertex $v\in V$, denoted $\deg(v)$, 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 $\delta(G) = \max_{v \in V} \deg(v)$.

Though the bound in (4) is relatively simple to calculate, it does not provide a tight lower bound on M. In [14], another bounding lemma is given which produces tighter bounds on M. Consider the network $G = (V, E)$ as we have previously defined. That is, V is the set of stations, and E is the set of direct collisions. Suppose that the edge set E is expanded to also include hidden collisions, and we call this new set E'. We now have a new graph, namely $G' = (V, E')$ which is known as the *augmented* graph of G. Then finding a broadcast schedule in G which would minimize the frame length is equivalent to finding a proper coloring of vertices in G' (*i.e.* such that any two neighbors are assigned different colors) with a minimum number of colors.

Recall that a *clique* in a graph $G' = (V, E')$ is a subset of V such that any two vertices in that subset are adjacent. Then

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

where $\omega(G')$ is the clique number, which is the maximal cardinality of a clique in G'. The latter provides better bounds on the frame length [29]; however, computing the clique number of a graph is *NP*-hard in general.

The third and final bound to be discussed is a method based on semidefinite programming and is the value of the so-called Lovász theta function. Lovász first introduced the θ-function in [20]. Given a graph $G = (V, E)$, a subset of vertices $V' \subseteq V$ is said to be an *independent set* if no two members of V' are connected by an edge in E. Lovász showed that the θ-function has the property that

$$\alpha(G) \leq \theta(G) \leq \chi(\bar{G}),$$

where $\alpha(G)$ is the independence number (cardinality of a maximum independent set) in G and $\chi(\bar{G})$ is the chromatic number of the complement of G. The θ-function can be computed in polynomial time with arbitrary accuracy [11], whereas computing $\alpha(G)$ is *NP*-hard in general. We can use $\theta(\bar{G}')$, where $\bar{G}'$ is the complement of G', to compute a bound on M which is in general tighter than the bound given by $\omega(G')$:

$$\omega(G') = \alpha(\bar{G}') \leq \theta(\bar{G}') \leq \chi(G') = M. \tag{6}$$

Even though the bound based on θ-number of $\bar{G}'$ is polynomially computable in theory, the CPU time required to solve the corresponding semidefinite program is usually unacceptably high for large-scale instances and approximate algorithms are used.

3. Heuristics

In this section, we will introduce and discuss four heuristics which have been applied to the BSP with varying degrees of success. The algorithms are as follows:

- ◇ Greedy Randomized Adaptive Search Procedure (GRASP) [3],
- ◇ Sequential Vertex Coloring (SVC) [32],
- ◇ Mean Field Annealing (MFA) [30], and
- ◇ Mixed Neural-Genetic Algorithm (HNN-GA) [29].

3.1. GRASP

Greedy Randomized Adaptive Search Procedure (GRASP), originally introduced by Feo and Resende in [6], is a two-phase iterative meta-heuristic for combinatorial optimization [7, 8, 27]. In the first phase, referred to as the construction phase, a greedy randomized initial feasible solution is created. Then in the second phase, the initial solution is improved by the application of a local search procedure. The solution which is best out of all GRASP iterations is returned. GRASP has been applied to many combinatorial problems such as quadratic assignment [19, 21], job shop scheduling [2, 1], private virtual circuit routing [25], and satisfiability [28]. GRASP was successfully applied to the BSP by the current authors in [3].

3.1.1 Construction Phase. The construction phase for the GRASP constructs a solution iteratively from a partial broadcast schedule which is initially empty. The stations are first sorted in descending order of the number of one-hop and two-hop neighbors. Next, a so-called *Restricted Candidate List* (RCL) is created and consists of those greedily selected stations which may broadcast simultaneously with the stations previously assigned to the current slot. From this RCL a station is randomly chosen and assigned in the current slot. A new RCL is created and another station is randomly selected. This process continues until there are no stations to put in the RCL, at which time the slot number is incremented and the procedure is repeated recursively for the subgraph induced by the set of all vertices whose corresponding stations have not yet been assigned to a time slot.

3.1.2 Local Search. The local search phase used is a swap-based procedure which is adapted from a similar method for graph coloring implemented by Laguna and Martí in [17]. First, the two slots with the fewest number of scheduled transmissions are combined and the total number of slots is now given as $k = m - 1$, where m is the frame length of the schedule computed in the construction phase. Denote the new broadcast schedule as $\{s_{m',n} : m' = 1, \ldots, k,\, n = 1, \ldots, N\}$. 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 successful 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 "smallest" slots. If the procedure ends with $f(s) > 0$, then no improved solution was found.

3.2. Sequential Vertex Coloring

In [32], Yeo et al. take a multi-objective optimization approach to solving the BSP. They implement a two-phase heuristic based on the idea of sequential vertex coloring (SVC). In the first phase, they only consider the problem of minimizing the frame length. Then in phase 2, the frame length is fixed with the solution from phase 1 and the utilization within the frame is maximized.

3.2.1 Frame Length Minimization. For this phase, the frame length minimization in BSP is attacked by solving the graph coloring problem in the augmented graph. More specifically, an algorithm based on the sequential vertex ordering method is used to solve this problem. This is done by first ordering the stations in descending order of the number of one-hop and two-hop neighbors. The first vertex is colored and the list of the other $N - 1$ vertices are scanned downward. The remaining vertices are colored with the smallest color which has not already been assigned to a one-hop neighboring station. The process is continued until all vertices have been assigned a color.

3.2.2 Utilization Maximization. Beginning with this initial schedule, phase 2 attempts to maximize the throughput in the TDMA frame. To maximize the utilization within the frame whose length was determined in phase 1, an ordering method of the sequential vertex coloring algorithm is applied. The stations are now ordered in ascending order of the the number of one-hop and two-hop neighbors. The first ordered station is then assigned to any slots in which it can simultaneously broadcast with the previously assigned stations. This process is repeated with every station in the ordered list.

3.3. Mean Field Annealing

In 1997, Wang and Ansari [30] proposed a heuristic for the BSP based on Mean Field Annealing (MFA). In statistical mechanics, the physical process of annealing is used to relax a system to the state of minimal energy. This is done by heating the solid until it melts and then cooling it slowly so that at each temperature the particles randomly arrange themselves until reaching thermal equilibrium.

In [15], Kirkpatrick et al. introduced a method for combinatorial problems known as *simulated annealing*. Based on the theory of the physical process, simulated annealing was shown to asymptotically converge to the global minimum after performing a number of so-called transitions at decreasing temperatures.

Though simulated annealing is guaranteed to converge to the global optimal solution, this process is quite often computationally expensive. Mean field annealing, a heuristic which mimics the idea of mean field approximation from statistical physics [23] can be employed instead. In MFA, the stochastic process in simulated annealing is replaced by a set of deterministic equations. Though MFA does not guarantee convergence to a global optimal solution, it can provide an excellent approximation to an optimal solution and is much less expensive computationally.

3.4. Mixed Neural-Genetic Algorithm

As in the algorithm presented by Yeo et al. in [32], Salcedo-Sanz et al. [29] introduced a two-phase heuristic based on combining both Hopfield neural networks [13] and genetic algorithms as in [31]. As with the vertex coloring algorithm, phase one of the mixed neural-genetic algorithm minimizes framelength and phase two attempts to maximize the utilization within the slot.

3.4.1 Frame Length Minimization. The frame length minimization problem presented in [29] is the same as described above. For the solution, a discrete-time binary Hopfield neural network (HNN) is used. As described in [29], the HNN can be represented as a graph whose vertices are the neurons (stations) and whose edges are the direct collisions. The graph is then mapped to the schedule matrix S as defined in Section 2 above. The neurons are updated one at a time after a randomized initialization until the system converges.

3.4.2 Utilization Maximization. In this phase, a genetic algorithm is used to maximize the channel utilization within the frame length that was determined in phase one. A HNN is also used to ensure that all constraints are satisfied. Genetic algorithms receive their name from an explanation of the way they behave. Not surprisingly, they are based on Darwin's Theory of Natural Selection. Genetic algorithms store a set of solutions and then work to replace these solutions with better ones based on certain fitness criterion represented by the objective function value.

4. Computational Results

All the aforementioned heuristics were tested using three examples introduced by Wang and Ansari in [30] 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 these networks can be seen in Figure 1.1.

Though these examples provide reasonable insight into the quality of a given heuristic, they are relatively trivial networks. Notice in Figure 1.3 that the lower bound calculations are trivial for these test cases. Therefore, in order to get a better comparison of the various heuristics, the authors generated 60 random unit disc graphs having varying radii and stations. Twenty graphs of networks having 50, 75, and 100 stations were generated with each broken up into four sets of five graphs with radii of 20, 30, 40, and 50 units. Unit disc graphs are popular models. In such a graph, each node has the same transmission range equal to the radius of a disc centered at the node.

4.1. Average Time Delay

Evaluating the average time delay of packets as the utilization of the network increases is an effective means of evaluating a broadcast scheduling heuristic. The following assumptions were made before deriving the average time delay [30]:

1 Packets have a fixed length, and the length of a time slot is equal to the time required to transmit a packet.

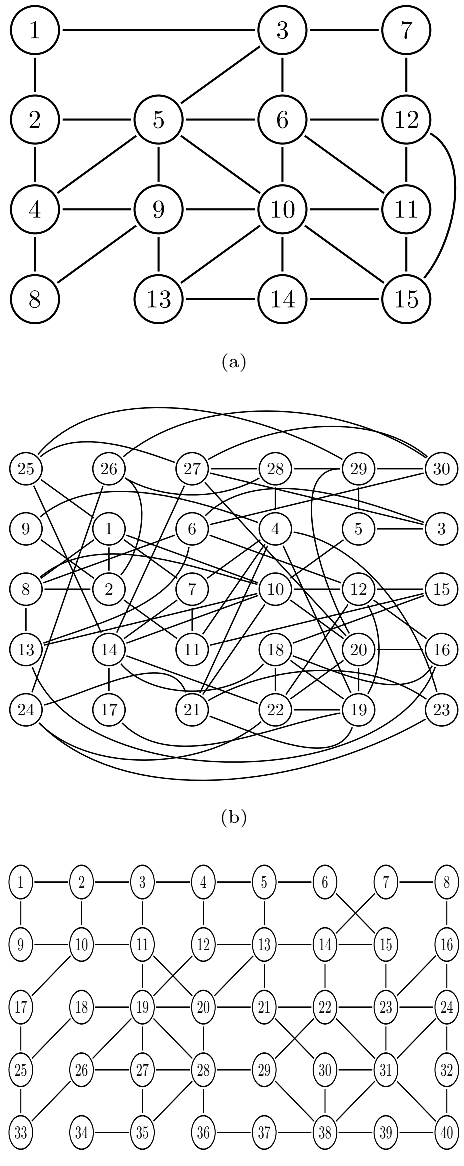


Figure 1.1. (a) 15 station network. (b) 30 station network. (c) 40 station network.

2 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 λ_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.

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

4 Packets may be transmitted only at the beginning of each time slot.

Using the above assumptions, 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* [12] 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:

$$D_i = \frac{1}{\mu_i} + \frac{\lambda_i/\mu_i^2}{2(1-\rho_i)}, \tag{7}$$

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

$$D = \frac{\sum_{i=1}^{N} \lambda_i D_i}{\sum_{i=1}^{N} \lambda_i}. \tag{8}$$

Notice that the average time delay is most affected by the number of time slots in the frame. The second dominating factor is the total utilization within the frame. As expected, those heuristics which result in schedules with fewer slots and greater throughput experience a less significant delay as the arrival rate increases. Note that some delay plots may be indistinguishable due to the overlapping with others with similar slots and utilizations. Graphs of the average time delay for the 15, 30 and 40 station networks can be seen in Figure 1.2.

4.2. Numerical Results

Comparative results for the aforementioned broadcast scheduling heuristics are shown in Figures 1.3 and 1.4. Due to lack of availability, the MFA algorithm was not considered for the larger instances whose results are given in Figure 1.4. Note that in the 15, 30, and 40 station examples, GRASP and the mixed neural-genetic algorithm all achieve schedules with optimal frame lengths of 8, 10, and 8 respectively. Examples of typical broadcast schedules for these example networks can be seen in Figure 1.5 [3].

With the results from the three examples shown in Figure 1.3 it appears as if all the routines perform well. However, the results of the randomly generated unit disc graphs given in Figure 1.4 provide a better insight into the abilities of the various heuristics against more formidable instances.

Notice that the sequential vertex coloring algorithm tends to decline in performance rather rapidly whereas the GRASP and mixed neural-genetic algorithm fare quite well against the tougher cases. For all cases, GRASP attains a frame length less than or equal that of the other heuristics with quite comparable utilizations. The mixed neural-genetic algorithm also yields good results with respect to utilization and frame length.

Recall that it is the total number of slots that most greatly affects the average time delay of packets in the network. Thus, for the tested networks, it appears as if GRASP would be the heuristic of choice.

5. Conclusion

In this chapter, we introduced the Broadcast Scheduling Problem and proved its *NP*-completeness. We formally defined the problem and discussed several heuristics which have been applied to the BSP, all with competitive results. Over 60 different networks of varying size and densities were tested and the results compared. Some heuristics approached the problem of minimizing the total slots per frame, while others partitioned the BSP into two problems and used varying methods to arrive at a final answer. As research on ad-hoc networks increases, so too will applications of the BSP and other exciting problems which are constantly arising in research on optimization in telecommunications.

6. Acknowledgements

The authors would like to thank Dr. Sancho Salcedo-Sanz for donating the source code from the mixed neural-genetic algorithm [13] described in Section 3.4 to be tested on the unit disk graphs described previously. The SVC algorithm was coded by the current authors for testing.

References

[1] R.M. Aiex, S. Binato, and M.G.C. Resende. Parallel GRASP with path-relinking for job shop scheduling, *Parallel Computing*, vol. 29, Pages 393-430, 2003.

[2] S. Binato, W.J. Hery, D.M. Loewenstern, and M.G.C. Resende. A GRASP for job shop scheduling, *Essays and Surveys on Metaheuristics*, C.C. Ribeiro and P. Hansen, Eds., Kluwer Academic Publishers, Pages 58-79, 2002.

[3] C.W. Commander, S.I. Butenko, and P.M. Pardalos. A Greedy Randomized Adaptive Search Procedure for the Broadcast Scheduling Problem, submitted, 2004.

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

[5] S. Even, O. Goldreich, S. Moran, and P. Tong. On the NP-completeness of Certain Network Testing Problems. *Networks*, 14, 1984.

[6] T.A. Feo and M.G.C. Resende. A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem. *Operations Research Letters*, 8, Pages 67-71, 1989.

[7] T.A. Feo and M.G.C. Resende. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6, Pages 109-133, 1995.

[8] P. Festa and M.G.C. Resende. GRASP: An Annotated Bibliography. In C.C. Ribeiro and P. Hansen, editors, *Essays and Surveys on Metaheuristics*, Pages 325-367. Kluwer Academic Publishers, 2001.

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

[10] F. Glover. Tabu Search and Adaptive Memory Programming - Advances, Applications, and Challenges. In R.S. Barr, R.V. Helgason, and J.L. Kenngington, editors, *Interfaces in Computer Science and Operations Research*, Pages 1-75. Kluwer Academic Publishers, 1996.

[11] M. Grotschel, L. Lovász, and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. *Springer-Verlag, Berlin*, 1987

[12] F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research.* New York: McGraw Hill, 2001.

[13] J.J. Hopfield and D.W. Tank. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Science*, Pages 2541-2554, 1982.

[14] D. Jungnickel, *Graphs, Networks and Algorithms.* Berlin, Germany: Springer-Verlag, 1999.

[15] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by Simulated Annealing. *Science*, 220, Pages 671-680, 1983.

[16] L. Kleinrock, and J. Silvester. Spatial Reuse in Multihop Packet Radio Networks. *Proceedings of the IEEE 75*, 1987.

[17] M. Laguna and R. Martí. A GRASP for Coloring Sparse Graphs. *Computational Optimization and Applications*, vol. 19, no. 2, Pages 165-178, 2001.

[18] M. Laguna and R. Martí. GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. INFORMS Journal on Computing, 11, Pages 44-52, 1999.

[19] Y. Li, P.M. Pardalos, and M.G.C. Resende. A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem, *Quadratic Assignment and Related Problems*, P.M. Pardalos and H. Wolkowicz, eds., *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, vol. 16, Pages 237-261, 1994.

[20] L. Lovász. On the Shannon Capacity of a Graph. *IEEE Transactions on Information Theory*, vol. IT-25, no. 1, Pages 1-7, 1979.

[21] C.A.S. Oliveira, P.M. Pardalos, and M.G.C. Resende. GRASP with Path-Relinking for the QAP, *5th Metaheuristics International Conference*, Pages 57.1-57.6, 2003.

[22] P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. A Parallel GRASP Implementation for the Quadratic Assignment Problem, *Parallel Algorithms for Irregular Problems*, A. Ferreira and J. Rolim, eds, Kluwer Academic Publishers, Pages 111-130, 1995.

[23] C. Peterson and B. Soderberg. A New Method for Mapping Optmization Problems onto Neural Networks, *International Journal of Neural Systems*, vol. 1, no. 1, Pages 3-22, 1989.

[24] L.S. Pitsoulis, P.M. Pardalos, and M.G.C. Resende. A Parallel GRASP for MAX-SAT, *Lecture Notes in Computer Science*, vol. 1180, Pages 575-585, 1996.

[25] M.G.C. Resende and C.C. Ribeiro. A GRASP with Path-relinking for Private Virtual Circuit Routing, *Networks*, vol. 41, Pages 104-114, 2003.

[26] M.G.C. Resende and C.C. Ribeiro. GRASP with Path-relinking: Recent Advances and Applications, *Metaheuristics: Progress as Real Problem Solvers*, T. Ibaraki, K. Nonobe and M. Yagiura, (Eds.), Kluwer Academic Publishers, 2005.

[27] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures, in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, eds., Kluwer Academic Publishers, Pages 219-249, 2003.

[28] M.G.C. Resende and T.A. Feo. A GRASP for Satisfiability. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, David S. Johnson and Michael A. Trick ,Eds., *DIMACS Se-*

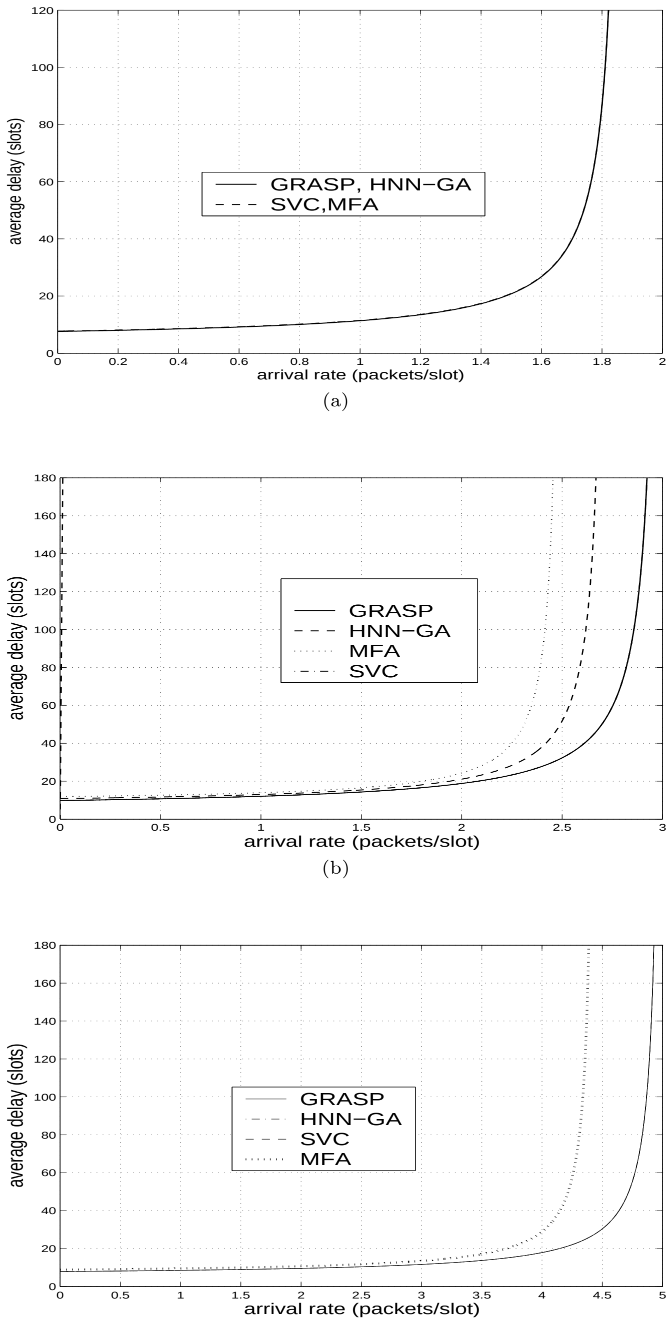
ries on Discrete Mathematics and Theoretical Computer Science, vol. 26, Pages 499-520, American Mathematical Society, 1996.

[29] 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*, vol. 2, no. 2, 2003.

[30] G. Wang and N. Ansari. Optimial Broadcast Scheduling in Packet Radio Networks Using Mean Field Annealing. *IEEE Journal on Selected Areas in Communications*, 15(2), 1997.

[31] Y. Watanabe, N. Mizuguchi, and Y. Fujii. Solving Optimization Problems by Using a Hopfield Neural Network and Genetic Algorithm Combination. *Syst. Comput. Japan* vol. 29, no. 10, Pages 68-73, 1998.

[32] 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.



(c)

Figure 1.2. Comparison of average time delays for example networks: (a) 15 station network, (b) 30 station network, (c) 40 station network.

Stations	LB	Frame Length				Channel Utilization			
		GRASP	HNN-GA	SVC	MFA	GRASP	HNN-GA	SVC	MFA
15	8	8	8	8	8	0.167	0.167	0.15	0.15
30	10	10	10	11	12	0.120	0.117	0.112	0.108
40	8	8	8	8	9	0.203	0.209	0.188	0.197

Figure 1.3. In this table, LB represents the lower bounds of M which were derived using the method in [14]. The results compare the heuristics discussed in Section 4.

		GRASP	HNN-GA	SVC
	r = 20	n = 7, ρ = .288	n = 7, ρ = .2926	n = 7.4, ρ = .1997
n = 50	r = 30	n = 8.8, ρ = .1709	n = 9, ρ = .1985	n = 10, ρ = .1416
	r = 40	n = 13.2, ρ = .1142	n = 13.2, ρ = .1142	n = 15.2, ρ = .0876
	r = 50	n = 15.8, ρ = .0835	n = 16, ρ = .0845	n = 17.8, ρ = .0701
	r = 20	n = 8, ρ = .209	n = 8, ρ = .2187	n = 9.6, ρ = .1764
n = 75	r = 30	n = 13, ρ = .1213	n = 13, ρ = .1273	n = 13.8, ρ = .0964
	r = 40	n = 18.4, ρ = .0786	n = 18.4, ρ = .0803	n = 20.6, ρ = .0576
	r = 50	n = 25.2, ρ = .0564	n = 26, ρ = .0583	n = 28.2, ρ = .0523
	r = 20	n = 10.8, ρ = .1628	n = 10.8, ρ = .1966	n = 11.6, ρ = .1126
n = 100	r = 30	n = 16.8, ρ = .0899	n = 16.8, ρ = .0938	n = 17, ρ = .0769
	r = 40	n = 24.8, ρ = .0613	n = 24.8, ρ = .0640	n = 26.6, ρ = .0515
	r = 50	n = 33.2, ρ = .0425	n = 34, ρ = .0435	n = 35.6, ρ = .0360

Figure 1.4. The average frame length and utilization are given for the randomly generated networks consisting of 50, 75, and 100 stations with radii of 20, 30, 40, and 50 units.

slot \ station	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1					B										
2						B		B							
3							B			B					
4	B								B			B			
5			B					B							B
6	B							B			B				
7				B			B							B	
8		B					B						B		

(a)

slot \ station	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1										B	B																			B
2						B			B					B																
3						B	B													B										
4				B									B												B					
5		B										B											B				B			
6					B														B							B				
7		B	B													B					B									
8								B														B						B		
9	B																	B					B							B
10									B				B		B		B							B					B	

(b)

slot \ station	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		B						B							B					B
2	B				B		B										B		B	
3				B						B				B		B				
4	B					B		B			B									
5		B				B		B				B								
6		B				B	B											B		
7	B							B					B				B			
8			B						B						B					

slot \ station	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1												B	B	B				B		
2			B											B					B	
3								B		B			B							
4											B		B	B		B				
5		B			B							B			B		B			
6				B					B					B						
7						B			B					B						B
8	B						B										B			

(c)

Figure 1.5. Example GRASP broadcast schedules for the networks given in Figure 1.1: (a) 15 station network, (b) 30 station network, (c) 40 station network [3].