Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Neural Networks 18 (2005) 765–771
Neural Networks
www.elsevier.com/locate/neunet

2005 Special Issue

Broadcast scheduling in wireless multihop networks using a neural-network-based hybrid algorithm ☆

Haixiang Shi*, Lipo Wang

Software Engineering Laboratory, School of Electrical & Electronic Engineering, Nanyang Technological University, S2.2-B4-04, 50 Nanyang Avenue, 639798, Singapore

Abstract

In wireless multihop networks, the objective of the broadcast scheduling problem is to find a conflict free transmission schedule for each node at different time slots in a fixed length time cycle, called TDMA cycle. The optimization criterion is to find an optimal TDMA schedule with minimal TDMA cycle length and maximal node transmissions. In this paper we propose a two-stage hybrid method to solve this broadcast scheduling problem in wireless multihop networks. In the first stage, we use a sequential vertex-coloring algorithm to obtain a minimal TDMA frame length. In the second stage, we apply the noisy chaotic neural network to find the maximum node transmission based on the results obtained in the previous stage. Simulation results show that this hybrid method outperforms previous approaches, such as mean field annealing, a hybrid of the Hopfield neural network and genetic algorithms, the sequential vertex coloring algorithm, and the gradual neural network.

Keywords: Broadcast scheduling problem; wireless multihop networks; sequential vertex coloring; noisy chaotic neural network; NP-hard

1. Introduction

Wireless multihop networks (WMNs) have been deployed since the 60 s in military, commercial, and academic environments, including the Internet. WMNs provide easy-to-use mobile services including military and disaster-relief communications (Lloyd, 2002). It is a good option for high-speed wireless data communications, especially over a broad geographic region (Leiner, Nielson, & Tobagi, 1987). In WMNs, since all nodes communicate with each other using a shared radio channel, uncontrolled transmissions may lead to time overlaps between two or more packet receptions, which are called conflicts, resulting in damaged packets at the destination. These damaged packets increase the network delay because they must be retransmitted (Yeo, Lee, & Kim, 2002). Effective broadcast scheduling is necessary to avoid any conflict and to use channel resource efficiently. The *time−division multiple−access* (TDMA) protocol has been adopted to obtain conflict-free transmissions. In a TDMA network, time is divided into frames and each TDMA frame is a collection of time slots. The TDMA cycle length is the total number of time slots in a frame cycle. A time slot has a unit time length required for a single packet to be communicated between adjacent nodes. When nodes transmit simultaneously, conflicts will occur if the nodes are in a close range. Therefore, adjacent nodes must be scheduled to transmit in different time slots, while nodes some distance away may be arranged to transmit in the same time slot without causing conflicts (Wang & Ansari, 1997). The goal of the broadcast scheduling problem (BSP) is to find an optimal TDMA frame structure that fulfills the following two objectives: the first is to schedule transmissions of all nodes in a minimal TDMA length without any conflict, the second is to maximize the total conflict-free transmissions in order to maximize channel utilization.

In the previous research, various heuristic methods have been adopted to solve the BSP. We categories these methods into two categories. One is the single-stage method which solves the two objectives in one step (Chakraborty & Hirano, 1998; Ephremides & Truong, 1990; Funabiki & Takefuji, 1993; Wang & Ansari, 1997) and the other is the two-stage method which solves the two objectives in two

☆ An abbreviated version of some portions of this article appeared in (Shi & Wang, 2005), published under the IEEE copyright.

* Corresponding author.

0893-6080/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2005.06.013

separate stages (Funabiki & Kitamichi, 1999; Salcedo-Sanz, no Calzón, & Figueiras-Vidal, 2003; Yeo et al., 2002).

Ephremides and Truong (1990) proposed a distributed greedy algorithm to find a TDMA structure with the maximal number of transmissions. They proved that the optimal scheduling of broadcasts in a radio network is NP-complete. Funabiki and Takefuji (1993) proposed a parallel algorithm based on an artificial neural network to solve the M-slot problem where the number of time slots is pre-defined. They used hill-climbing to help the system escape from local minima. Wang and Ansari (1997) proposed a mean field annealing (MFA) algorithm to find a TDMA cycle with the minimum delay time. In order to find the minimal frame length, they first used MFA to find assignments for all nodes with a lower bound of the frame length. For the unassigned nodes left, they then used another heuristic algorithm, which adds one time slot at each iteration to the node with the highest degree. The heuristic algorithm was run repeatedly until all the nodes left were assigned. After the number of time slots was found, possibilities of making additional feasible assignments to the nodes were checked. They also proved NP-completeness of the BSP by transforming the BSP to the maximum independent set problem. Chakraborty and Hirano (1998) used genetic algorithms with a modified crossover operator to handle large networks with complex connectivity.

Recently, Funabiki and Kitamichi (1999) proposed a binary neural network with a gradual expansion scheme, called a gradual neural network (GNN), to find the minimum frame length and the maximum transmissions through a two-phase process. The performance of their method was demonstrated through the three benchmark instances used in (Wang & Ansari, 1997) and randomly generated geometric graph instances. Yeo et al. (2002) proposed a two-phase algorithm based on sequential vertex coloring (SVC). They showed that their method can find better solutions compared to the method in (Wang & Ansari, 1997). Salcedo-Sanz et al. (2003) proposed a hybrid algorithm HNN-GA which combines a Hopfield neural network for constraint satisfaction and a genetic algorithm for achieving maximal throughput. They compared their results with MFA in (Wang & Ansari, 1997) in the three benchmark problems, and showed that the hybrid algorithm outperformed the MFA.

In this paper, we propose a two-stage hybrid method SVC-NCNN that combines a sequential vertex coloring (SVC) algorithm and the noisy chaotic neural network (NCNN) (Li & Wang, 2001; Wang & Tian, 2000; Wang, Li, Tian, & Fu , 2004; Wang, Li, Wan, & Soong, 2003). We use the SVC to obtain the minimal frame length in the first stage and the noisy chaotic neural network to obtain the maximal node transmissions in the second stage (preliminary results of this work have been reported in (Shi & Wang, 2005)). The rest of paper is organized as follows. In Section 2, we review and formulate the broadcast scheduling problem. In Section 3, we propose our hybrid algorithm. Three benchmark problems and three large network instances are solved and the performance is evaluated in Section 4. Section 5 presents our conclusions.

2. Formulating the broadcast scheduling problem

A wireless multihop network can be represented by a graph $G=(V, E)$, where vertices in $V=\{1,\ldots,N\}$ are network nodes, N being the total number of nodes in the network, and E represents the set of transmission links. Two nodes i and j ($i, j\in V$) are connected by an undirected edge $e_{ij}\in E$ if and only if they can receive each other's transmission. In such a case, the two nodes i and j are said to be one hop away. The connectivity matrix $C=\{c_{ij}\}$, $\{i, j=1,\ldots,N\}$ is defined as follows.

$$c_{ij} = \begin{cases} 1, & \text{if node } i \text{ and } j \text{ are one hop away;} \\ 0, & \text{otherwise.} \end{cases}$$

If $e_{ij}\notin E$, but there is an intermediate node k such that $e_{ik}\in E$ and $e_{kj}\in E$, then nodes i and j are two hops away. The $N\times N$ compatibility matrix $\mathrm{D}=\{d_{ij}\}$ (Funabiki & Kitamichi, 1999) is defined as follows.

$$d_{ij} = \begin{cases} 1, & \text{if node } i \text{ and } j \text{ are within two hops;} \\ 0, & \text{otherwise.} \end{cases}$$

A *primary conflict* occurs when two nodes which are one hop away transmit in the same time slot, as shown in Fig. 1(a). A *secondary conflict* occurs if two nodes are two hops away and transmit in the same time slot, as shown in Fig. 1 (b).

We summarize the constraints in the BSP in the following two categories:

(1) *No-transmission constraint* (Chakraborty & Hirano, 1998): Each node should be scheduled to transmit at least once in a TDMA cycle.
(2) *No-conflict constraint*: It excludes the primary conflict (a node cannot have transmission and reception simultaneously) and the secondary conflict (a node is not allowed to receive more than one transmission simultaneously).

Hence two nodes can transmit in the same time slot without conflicts if and only if they are more than two hops away from each other.

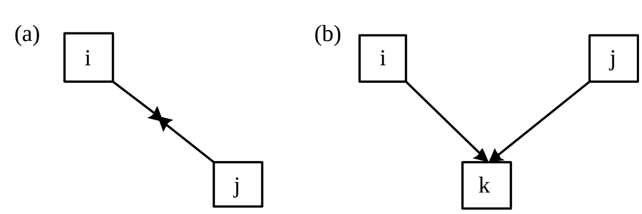

Fig. 1. Situations in which conflicts occur in a wireless multihop network, where i, j, and k represent nodes in the wireless multihop network. (a) Primary conflict. (b) Secondary conflict.

The final optimal solution obtained is a transmission schedule consisting of M time slots. We use an $M \times N$ binary matrix $T=(t_{ij})$ to express a transmission schedule, where

$$t_{ij} = \begin{cases} 1, & \text{if node } i \text{ transmits in slot } j \text{ in a frame;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Channel utilization ρ_j for node j is defined as (Wang & Ansari, 1997)

$$\rho_j = \frac{\text{the number of time slots assigned to node } j}{\text{TDMA cycle length}} = \frac{\sum_{i=1}^{M} t_{ij}}{M}. \tag{2}$$

The total channel utilization for the entire network, ρ, is given by (Wang & Ansari, 1997):

$$\rho = \frac{1}{N}\sum_{j=1}^{N}\rho_j = \frac{1}{NM}\sum_{j=1}^{N}\sum_{j=1}^{M} t_{ij}. \tag{3}$$

The goal of the BSP is to find a transmission schedule with the shortest TDMA frame length (i.e. smallest M) which satisfies the above constraints, and the total transmissions are maximized. Based on the above description of the BSP, we formulate it as follows:

BSP

(1) Minimize the frame cyclelength M;
(2) Maximize channel utilization ρ

$$\text{Subject to}: \quad \sum_{i=1}^{M} t_{ij} \geq 1, \quad j = 1, 2, \ldots, N, \tag{4}$$

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1, k\neq j}^{N} d_{jk} t_{ij} t_{ik} = 0. \tag{5}$$

The no-transmission constraint is formulated in Eq. (4), which means each node in the network must transmit at least once in a frame. The no-conflict constraint in Eq. (5) indicates that every pair of nodes within one hop or two hops cannot be scheduled in the same time slot.

3. Two stage optimization using a neural-network-based hybrid method

3.1. Stage I: Minimize TDMA frame length using SVC

In this stage, we minimize the TDMA frame length, which can be reduced to a variation of the classical vertex coloring problem (VCP) (Yeo et al., 2002). If we regard the colors in the VCP as the time slots in the BSP and vertices in the VCP as nodes in the BSP, the frame length minimization problem becomes the same as the VCP if we require that different slots (colors) must be assigned to one-hop-away and two-hop-away nodes (Yeo et al., 2002). In this paper, the phrases of color and time slot are interchangeable, so are node and vertex. The sequential coloring algorithm with vertex ordering is the simplest and practical heuristic algorithm to solve the VCP (Christofides, 1975). In this paper, we use the vertex-coloring algorithm with a new vertex ordering criteria after an initialization, and we call our algorithm sequential vertex coloring (SVC). The proposed SVC is implemented as follows. A lower bound L_m of the frame length is computed and an initialization procedure pre-assigns a group of nodes to L_m time slots. The vertices are ordered using a traverse algorithm to be described below and the SVC assigns the first available time slot (color) to the first node (vertex). The number of time slots (colors) is increased by 1 when no colors are available for the node. The SVC stops when all nodes are assigned with a time slot.

It is more efficient to search for the frame length starting from a lower bound L_m compared to the value 1. L_m can be computed using graph theory (Jungnickel, 1999). Firstly the original graph $G=(V, E)$ is transformed into $G'=(V, E')$, where E in G stands for one-hop-away edges, and E' in G' stands for one-hop-away and two-hop-away edges. The lower bound L_m is:

$$L_m = \omega(G'). \tag{6}$$

where $\omega(G')$ is the maximal cardinality of a clique in G' (Salcedo-Sanz et al., 2003).

The initialization of the SVC is implemented as follows. We label the group of nodes to be initialized as $G=\{n_i, (i=1, 2, \ldots L_m\}$. The group G consists of the node with the maximal degree and the nodes which are one hop away from this node. Suppose node n_1 is the node with the maximal degree. The other nodes $\{n_i(i=2, 3, \ldots, L_m)\}$ represent nodes one hop away from node n_1. The initial assignments are done as follows: we assign time-slot i to node $n_i,(i=1, 2, \ldots, L_m)$. The initialization is based on the fact that all one-hop-away nodes must be assigned with different time slots. The initialization process can reduce the search time, because after the this process, only N-L_m nodes are left to be scheduled.

As stated in (Yeo et al., 2002), different vertex ordering criteria have different effects in searching. In this paper, in order to take advantage of the initial schedule process, we use the distance criteria, i.e. the topological distance from the pre-assigned nodes to the other unassigned nodes in the graph. It is obtained with a traverse algorithm as follows.

Algorithm traverse (stack S)

(1) Set $S=null$. Make a copy of G and name it as Q.
(2) Obtain a node x from the queue Q and delete node x from Q. If Q is *null*, then stop and return the node sequence stored in stack S. Else find the one-hop-away nodes of node x denoted as $N_{adj}=\{n_k,(k=1, 2, \ldots, p)\}$, where p is the number of one-hop-away nodes.
(3) For every node in N_{adj}, if node n_k is not in the queue Q, store the node n_k in Q. If n_k is not in G or queue S, store node n_k in queue S.
(4) Goto step 2.

Our SVC in the stage I can be described as follows.
Algorithm sequential vertex coloring

(1) Find the lower bound using Eq. (6). Set the current number of time slots as the lower bound, i.e. $M=L_m$.
(2) Assign time-slot i to node $n_i,(i=1,2,\ldots,L_m)$. Let the counter $p=M+1$.
(3) Order the uncolored vertices in the graph by the distance criteria.
(4) Find the number of non-conflicting colors c for the vertex i.
    (a) If $c>1$, assign the first available color to vertex i.
    (b) If $c=1$, assign the color to vertex i.
    (c) If $c<1$, goto Step 6.
(5) If p=N, stop; else p=p+1 and goto step 4.
(6) $M=M+1$ and goto step 4.

3.2. Stage II: Maximize the channel utilization

In this sub-section, we use the noisy chaotic neural network (NCNN) (Li & Wang, 2001; Wang et al., 2004; Wang et al., 2003; Wang & Tian, 2000) to maximize the total node transmissions based on the results obtained in the previous stage. The model of the NCNN, a stochastic variation of the transiently chaotic neural network (Chen & Aihara, 1995), can be described as follows.

$$x_{jk}(t)=\frac{1}{1+e^{-y_{jk}(t)/\varepsilon}}. \tag{7}$$

$$y_{jk}(t+1)=ky_{jk}(t)+\alpha\left(\sum_{\substack{i=1\\i\neq j}}^{N}\sum_{\substack{l=1\\l\neq k}}^{M}w_{jkil}x_{jk}(t)+I_{ij}\right), \quad -z(t)(x_{jk}(t)-I_0)+n(t). \tag{8}$$

$$z(t+1)=(1-\beta_1)z(t). \tag{9}$$

$$A[n(t+1)]=(1-\beta_2)A[n(t)]. \tag{10}$$

where

- x_{jk} output of neuron jk;
- y_{jk} input of neuron jk;
- w_{jkil} connection weight from neuron jk to neuron il, with $w_{jkil}=w_{iljk}$ and $w_{jkjk}=0$;

$$\sum_{\substack{i=1\\i\neq j}}^{N}\sum_{\substack{l=1\\l\neq k}}^{M}w_{jkil}x_{jk}+I_{ij}=-\partial E/\partial x_{jk}, \quad \text{input to neuron } jk. \tag{11}$$

- I_{jk} input bias of neuron jk;
- E energy function;
- k damping factor of nerve membrane ($0\leq k\leq 1$);
- α positive scaling parameter for inputs;
- β_1 damping factor for neuronal self-coupling ($0\leq\beta_1\leq 1$);
- β_2 damping factor for stochastic noise ($0\leq\beta_2\leq 1$);
- $z(t)$ self-feedback connection weight or refractory strength ($z(t)\geq 0$);
- I_0 positive parameter;
- ε steepness parameter of the output function ($\varepsilon>0$);
- $n(t)$ random noise injected into the neurons, in [-A, A] with a uniform distribution;
- $A[n]$ amplitude of noise n.

The energy function is (Funabiki & Kitamichi, 1999):

$$E=\frac{W_1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{\substack{k=1\\k\neq i}}^{N}d_{ik}x_{ij}x_{kj}+\frac{W_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}(1-x_{ij})^2. \tag{12}$$

where W_1 and W_2 are weighting coefficients. W_1 represents the relative strength of the constraint term that any pair of nodes which is one hop away or two hops away must not transmit simultaneously during each TDMA cycle. W_2 represents the strength of the optimization term which maximizes the total number of firing neurons or the total node transmissions.

From Eq. (8), Eqs. (11) and (12), we obtain the dynamics of the NCNN for neuron ij as follows:

$$y_{ij}(t+1)=ky_{ij}(t)+\alpha\left[-W_1\left(\sum_{\substack{k=1\\k\neq i}}^{N}d_{ik}x_{kj}\right)+W_2(1-x_{ij})\right] -z(t)[x_{ij}(t)-I_0]+n(t). \tag{13}$$

For an N-node BSP problem with M time slots, a noisy chaotic neural network with $N\times M$ neurons are adopted. We convert the continuous output x_{ij} of neuron ij to discrete neuron output t_{ij} as follows.

$$t_{ij}=\begin{cases}1 & x_{ij}>\sum_{k=1}^{N}\sum_{l=1}^{M}x_{kl}(t)/(N\times M),\\ 0 & \text{otherwise},\end{cases}$$

The $N\times M$ neurons are updated cyclically and asynchronously in the following sense. All neurons are cyclically updated in a fixed order. After all neurons are updated once, we consider that one iteration is finished. And the new state information is available for the other neurons in the next generation.

An issue related to efficiency of the NCNN in solving combinatorial optimization problems is to select appropriate parameters including system parameters in the neural network and weighting coefficients in the energy function. Although there are quite a few parameters to be selected in the NCNN, these parameters are similar to those used in other optimization problems (Li and Wang, 2001; Wang et al., 2004; Wang et al., 2003; Wang & Tian, 2000):

$$k = 0.9, \quad \alpha = 0.015, \quad \beta_1 = 0.001, \quad \beta_2 = 0.0001,$$
$$\varepsilon = 0.004, \quad I_0 = 0.65, \quad z_0 = 0.08, \quad A[n(0)] = 0.002. \tag{14}$$

The weighting coefficients are determined based on the rule that each term (W_1 and W_2) in energy function Eq. (12) should be comparable in magnitude, so that none of them dominates. Hence we choose $W_1 = 1.0$ and $W_2 = 0.6$ in the benchmark examples from BM (1 to BM (3. Besides, a comparison of different values of W_2 is carried out for Case 4 to Case 6 in Section 4.

The computational complexity of our hybrid method is examined as follows. In the first stage, the complexity for ordering vertex using the traverse algorithm is $O(N+L)$, where N is the number of node and L is the number of edges in the graph. The procedure to find non-conflicting colors in step (4) in the SVC algorithm has $O(N)$ complexity, which needs to run NM times. Thus the complexity in stage I is $O(N^2M)$, where N is the number of node and M is the number of time slot. In stage II, it is difficult to determine the exact number of iterations required by the NCNN for different problem instances with various problem sizes. We investigate the complexity in one iteration step instead: the worst time complexity in one iteration step for the NCNN is $O(NM)$.

4. Experimental results

4.1. Evaluation indices

We use three evaluation indices to compare with different algorithms. One is the TDMA frame length M. The second index is the channel utilization factor ρ defined in Eq. (3). The third is the average time delay η for each node to broadcast packets (Funabiki & Kitamichi, 1999):

$$\eta = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{M}{\sum_{j=1}^{M} t_{ij}}\right) = \frac{M}{N}\sum_{i=1}^{N}\left(\frac{1}{\sum_{j=1}^{M} t_{ij}}\right). \tag{15}$$

Another definition of average time delay can be found in (Wang & Ansari, 1997). To derive the definition of average time delay, the following assumptions were made (Wang & Ansari, 1997) (Yeo et al., 2002):

(1) Packets have a fixed length, and the length of a time slot is equal to the time required to transmit a packet.
(2) The inter-arrival 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 consist of its own traffic and the incoming traffic 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. Let the service rate of station i be μ_i (packets/slot).
(4) Packets can be transmitted only at the beginning of each time slot.

Under the above assumptions, the network can be modeled as $NM/D/1$ queues (a queue can be described by three parameters denoted as $A/S/m$, where A is the arrival process, S is the probability distribution of service times, and m is the number of servers. Here $M/D/1$ means a queue with Poisson distributed arrivals, constant service time distribution and 1 server.) (Bertsekas & Gallager, 1987). The average time delay for each node is defined as D_i, according to Pollaczek–Khinchin formula:

$$D_i = \frac{1}{\mu_i} + \frac{1}{\lambda_i}\frac{(\lambda_i/\mu_i)^2}{2(1-\lambda_i/\mu_i)}. \tag{16}$$

where $\mu_i = \sum_{m=1}^{M} x_{mi}/M$ (packet/slot).

The total time delay is given by (Wang & Ansari, 1997; Yeo et al., 2002):

$$D = \frac{\sum_{i=1}^{N}\lambda_i D_i}{\sum_{i=1}^{N}\lambda_i}. \tag{17}$$

In this paper, we will use both definitions of the average time delay in order to compare with other methods.

4.2. Simulation results

We first evaluate our hybrid method through three benchmark problems (BM #1 to BM #3 in Table 1), which were solved by the other methods compared in this paper. The three examples are 15-node-29-edge, 30-node-70-edge, and 40-node-66-edge, respectively. Besides the three benchmark problems, we apply our hybrid algorithm on several large instances (Case #4 to Case #6 in Table 1) with the number of node varying from 100 to 400. We run each instance 50 times with different, randomly selected initial states of the NCNN, but with the same set of system parameters as in Eq. (14).

The results of the three benchmark problems are summarized in Table 2 and compared with the best results obtained from other previous methods. The best resulting transmission scheduling for the BM #3 is shown in Fig. 2. A black square in this figure represents the transmission of

Table 1
Specifications of benchmark instances

Instance	Nodes N	Edges E	Maximum degree	Minimum degree	Lower bound
BM #1	15	29	7	2	8
BM #2	30	70	8	2	10
BM #3	40	66	7	1	8
Case #4	100	486	18	2	19
Case #5	200	1099	21	2	22
Case #6	400	805	8	2	9

Table 2
Comparisons of results obtained by our SVC-NCNN ($W_1=1.0$ and $W_2=0.6$) with different algorithms in the three benchmark problems

Case	SVC-NCNN		HNN-GA		SVC		GNN		MFA	
	η	M	η	M	η	M	η	M	η	M
BM #1	6.8	8	7.0	8	7.2	8	7.1	8	7.2	8
BM #2	9.2	10	9.3	10	10.0	10	9.5	10	10.5	12
BM #3	5.8	8	6.3	8	6.76	8	6.2	8	6.9	9

node i in time slot j. From this figure we can see that our proposed method can find feasible schedules for all nodes with the minimal frame length ($M=8$) in the first stage. Furthermore, the NCNN can find additional transmissions for each nodes, e.g., node 1 can transmit at additional time slots 4, 5, and 6 in one TDMA frame besides the time slot 1, i.e. node 1 can transmit 4 times in one frame cycle. From Table 2, we can see that our SVC-NCNN method can find shorter frame length than MFA does. In addition, our proposed hybrid method can find the smallest average time delay η among all methods in all three cases. In order to see clearly the difference between our method and other methods, we plot the average time delay for the three cases as shown in Fig. 3. Because no transmission schedules are reported in (Funabiki & Kitamichi, 1999), the GNN method cannot be compared and plotted in these figures. In

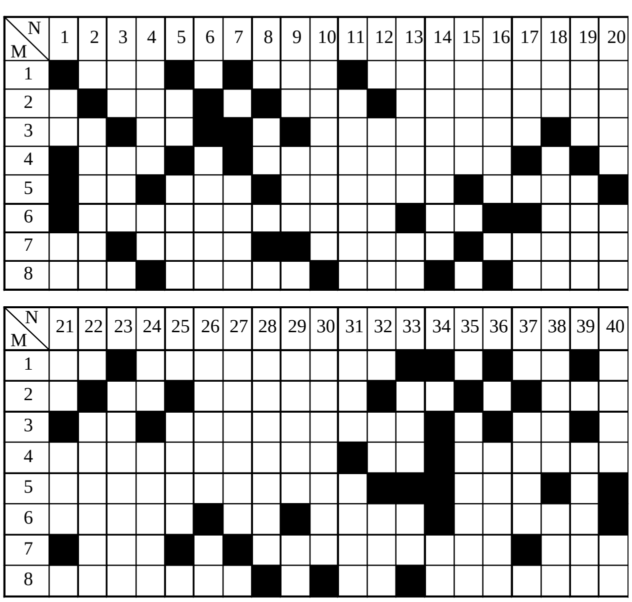

Fig. 2. The broadcast schedule for the 40-node-66-edge instance. N and M stand for the number of nodes and the number of time slots, respectively. The black square stands for the transmission of node i in slot j.

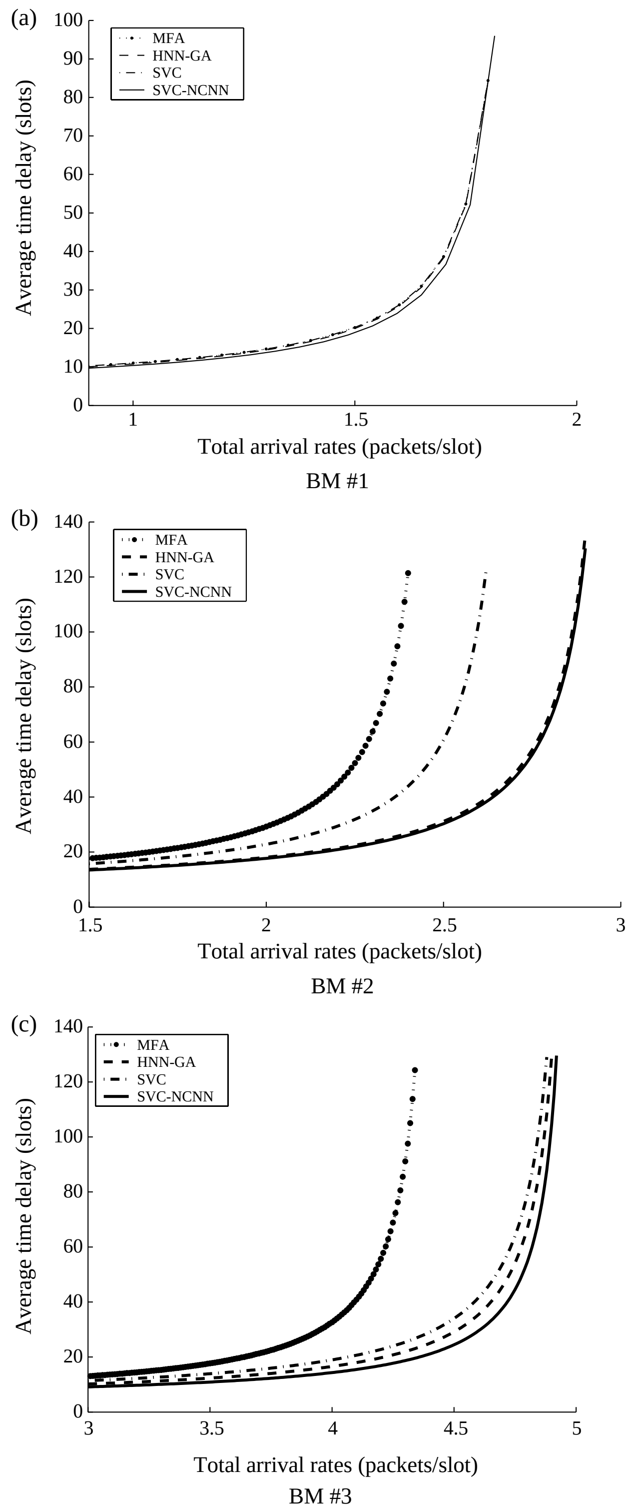

Fig. 3. Comparison of average time delays for the three benchmark problems BM #1 to BM #3 (according to Pollaczek–Khinchin formula Eq. (17)) with different approaches.

Table 3
Simulation results for Test Case #4 to #6 using our SVC-NCNN on a desktop personal computer with a 2.4 GHZ CPU and various choices of W_2 (W_1 is fixed at 1)

	W_2	M	η	Runtime (s)	Iteration steps	Converge rate (%)
Case #4	0.4	20	15.76	19.5	2697.5	100
	0.6	20	15.56	22.3	4335.2	100
	0.8	20	15.26	40.5	5894.5	100
	$\geq$0.9	19	–	107.0	15000	0
Case #5	0.4	22	19.22	87.2	2648.3	100
	0.6	22	18.28	102.3	4672.5	100
	0.8	22	17.45	325.4	6538.4	100
	$\geq$0.9	22	–	586.0	15000	0
Case #6	0.4	9	7.77	178.2	3915.3	100
	0.6	9	7.70	208.1	4634.9	100
	0.8	9	7.62	365.4	8321.4	100
	$\geq$0.9	9	–	623.0	15000	0

Fig. 3(a), it can be seen that all other methods obtain the same time delay. But our SVC-NCNN can find better transmission schedule which has the time delays less than other compared methods. The difference in time delay is greater as shown in Fig. 3(b) and (c) when the problem size becomes large. From Table 2 and Fig. 3, it can be concluded that SVC-NCNN can find the shortest conflict-free frame schedule while providing the maximum channel utilization among the existing algorithms compared in this paper.

In order to show the effect of weighting coefficients in the energy function, we investigate the effect of different coefficient W_2 in several large instances Case (4 to Case (6. Table 3 shows the results of test Case (6 using various values of W_2 while W_1 is fixed. We run the simulations 50 times and list the average values of time slot M, time delay η and runtime (in seconds). Better performance (smaller average time delays) can be obtained with larger W_2, but more iterations are required. When W_2 is greater than 0.9, the algorithm cannot find feasible solutions in the pre-defined 15000 iterations used in our simulations.

5. Conclusions

In this paper, we proposed a hybrid algorithm which combined sequential vertex coloring and the noisy chaotic neural network to solve the broadcast scheduling problem in wireless multihop networks. We used two stages for the two objectives of the BSP, i.e. we used sequential vertex coloring to find the minimal TDMA frame length with transmission scheduling in the first stage and the NCNN to maximize channel utilization and minimize time delays in the second stage. We evaluated our hybrid algorithm in three benchmark examples plus three larger instances. The results showed that our hybrid method finds better solutions than the other algorithms did in the examples.

References

Bertsekas, D., & Gallager, R. (1987). *Data networks*. Englewood Cliffs, NJ: Prentice-Hall.

Chakraborty, G. Hirano, Y. (1998). Genetic algorithm for broadcast scheduling in packet radio networks. In *IEEE world congress on computational intelligence*. (pp. 183–188).

Chen, L., & Aihara, K. (1995). Chaotic simulated annealing by a neural network model with transient chaos. *Neural Networks*, *8*, 915–930.

Christofides, N. (1975). *Graph theory—an algorithmic approach*. New York: Academic Press.

Ephremides, A., & Truong, T. V. (1990). Scheduling broadcast in multihop radio networks. *IEEE Transactions on Communications*, *38*, 456–460.

Funabiki, N., & Kitamichi, J. (1999). A gradual neural network algorithm for broadcast scheduling problems in packet radio networks. *IEICE Transactions on Fundamentals*, *E82-A*, 815–824.

Funabiki, N., & Takefuji, Y. (1993). A parallel algorithm for broadcast scheduling problems in packet radio networks. *IEEE Transactions on Communications*, *41*, 828–831.

Jungnickel, D. (1999). *Graphs, netwrks and algorithms*. Berlin, Germany: Springer.

Leiner, B. M., Nielson, D. L., & Tobagi, F. A. (1987). Issues in packet radio network design. *In Proceedings of IEEE*, *75*, 6–20.

Li, S., & Wang, L. P. (2001). *Channel assignment for mobile communications using stochastic chaotic simulated annealing In Proceedings of the 2001 International Work-Conference on Artificial Neural Networks*, Vol. 2084 pp. 757–776.

Lloyd, E. L. (2002). *Broadcast scheduling for TDMA in wireless multihop networks*. Wiley pp. 347–369.

Salcedo-Sanz, S., no Calzón, C. B., & Figueiras-Vidal, A. R. (2003). A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Transactions on Wireless Communications*, *2*, 277–283.

Shi, H., & Wang, L. P. (2005). A hybrid neural network for optimal tdma transmission scheduling in packet radio networks. In *Proceedings of the international joint conference on neural networks*. July 31–August 4, Montreal, Canada.

Wang, G., & Ansari, N. (1997). Optimal broadcast scheduling in packet radio networks using mean field annealing. *IEEE Journal on Selected Areas in Communications*, *15*, 250–260.

Wang, L. P., Li, S., Tian, F. Y., & Fu, X. J. (2004). A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing. *IEEE Transactions of System, Man, Cybern, Part B-Cybernetics*, *34*, 2119–2125.

Wang, L. P., Li, S., Wan, C., & Soong, B. H. (2003). *Minimizing interference in cellular mobile communications by optimal channel assignment using chaotic simulated annealing*. Berlin: Springer pp. 131–145.

Wang, L. P., & Tian, F. (2000). *Noisy chaotic neural networks for solving combinatorial optimization problems In Proceedings of International Joint Conference on Neural Networks*, Vol. 4 pp. 37–40.

Yeo, J., Lee, H., & Kim, S. (2002). An efficient broadcast scheduling algorithm for tdma ad-hoc networks. *Computer & Operations Research*, *29*, 1793–1806.