Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 9790629, 10 pages
http://dx.doi.org/10.1155/2016/9790629

Research Article

A New Distributed Approximation Algorithm for the Maximum Weight Independent Set Problem

Peng Du[1] and Yuan Zhang[2]

[1] *College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2] *National Mobile Communications Research Laboratory, Southeast University, Nanjing, China*

Correspondence should be addressed to Yuan Zhang; y.zhang@seu.edu.cn

Received 9 March 2016; Revised 29 July 2016; Accepted 9 August 2016

Academic Editor: Dan Simon

Maximum weight independent set (MWIS) is a combinatorial optimization problem that naturally arises in many applications especially wireless networking. This paper studies distributed approximation algorithms for finding MWIS in a general graph. In the proposed algorithm, each node keeps exchanging messages with neighbors in which each message contains partial solutions of the MWIS optimization program. A parameter H is introduced to achieve different tradeoff between approximation accuracy and space complexity. Theoretical analysis shows that the proposed algorithm is guaranteed to converge to an approximate solution after finite iterations; specifically, the proposed algorithm is guaranteed to converge to the optimal solution with $H = +\infty$. Simulation results confirm the effectiveness of the proposed distributed algorithm in terms of weight sum, message size, and convergence performance.

1. Introduction

Consider a graph $G = (V, E)$ with a set V of nodes and a set E of edges. For each node $i \in V$, there is a positive weight w_i. A subset of V can be represented by binary variable x_i, $1 \le i \le |V|$, where x_i is 1 if i is in the subset and 0 otherwise. A subset is called an independent set if no two nodes in the subset are connected by an edge. We are interested in finding the maximum weight independent set (MWIS) [1], which can be expressed as an integer program:

$$\begin{aligned} \max \quad & \sum_i w_i x_i \\ \text{s.t.} \quad & x_k + x_i \le 1, \quad (k, i) \in E \\ & x_i \in \{0, 1\}, \quad i \in V. \end{aligned} \tag{1}$$

The MWIS problem has been extensively studied in the literature. For example, it is known to be solvable in polynomial time for many cases including perfect graphs [2], interval graphs [2], disk graphs [3], claw-free graphs [4], fork-free graphs [5], trees [6], sparse random graphs [7, 8], circle graphs [9], and growth-bounded graphs [10]. Moreover, there has been extensive work on approximating the MWIS [11], and specialized algorithms have been developed for exactly computing the MWIS [12–15].

Further, the MWIS problem naturally arises in many applications, especially wireless networking, which require distributed solutions. For example [16–18], in scenarios involving resource scheduling in wireless networks that lack a centralized infrastructure and where nodes can only communicate with local neighbors, the MWIS problem needs to be solved in a distributed manner. Fundamentally, any two wireless nodes that transmit at the same resource will interfere with each other if they are located close-by. The scheduling problem is to decide which nodes should transmit at the given resource so that there is no interference and nodes with long queue length are given priority. If each node is given a weight equal to the queue length, it is optimal to schedule the set of nodes with the highest total weight. If a conflict graph is made, with an edge between each pair of interfering nodes, the scheduling problem is exactly the MWIS problem for the conflict graph. The lack of an infrastructure and the local nature of communication require a distributed algorithm for solving the MWIS problem.

There are three types of distributed algorithms for MWIS problem: greedy algorithms, carrier sense random multiple access (CSMA) algorithms, and message-passing algorithms.

For greedy algorithms, several simple distributed algorithms for MWIS have been proposed in the literature [19–22]. These algorithms are based on the greedy principle and require only knowledge of the local topology at each node. However, due to the inherent difficulty of the MWIS problem, greedy algorithms are not guaranteed to obtain the optimal solution for general graph.

For random multiple access algorithms, several CSMA-based algorithms have been proposed [23–26]. These algorithms are distributed, are of low complexity, and are easy to implement. They use a random access scheme according to which a link seizes the channel with a probability that increases exponentially with a certain link-dependent weight. As explained in [27], these schemes perform sampling over independent sets of the graph and converge to the optimal MWIS in graphs with large weights. However, according to [28], message-passing algorithms can achieve significantly smaller aggregate long-run average queue length than CSMA algorithms.

Several message-passing algorithms for producing a feasible solution to the MWIS were proposed in [28–34]. For example, in [30], Sanghavi et al. show that a simple modification of max product becomes gradient descent on the dual of the linear programming associated with the MWIS problem and converges to the dual optimum. They also develop a message-passing algorithm that recovers the primal MWIS solution from the output of the descent algorithm and show that the MWIS estimate obtained using these two algorithms in conjunction is correct when the graph is bipartite and the MWIS is unique. In [31], it has been shown that polynomial time methods exist for perfect graphs which include bipartite graphs and many others. Further, in [29], message-passing methods were developed for perfect graphs. Additionally, it is worth mentioning that message-passing methods have also been successfully applied to solve the maximum weight matching (MWM) problem [35–38].

This paper studies this problem and proposes a new distributed algorithm for the general graph. The proposed algorithm runs iteratively in which each node receives messages from its neighbors, updates the message of its own, and then broadcasts the message to the neighbors. During the message update procedure, each node shall first combine all received messages into a single one and then delete some elements from the message to control the size of the message. Theoretical analysis and computer simulation show that the proposed algorithm will converge after finite iterations and can achieve tradeoff between approximation accuracy and space complexity.

The rest of the paper is outlined as follows. In Section 2, some useful notations are introduced. In Section 3, a new distributed algorithm for MWIS is proposed. In Section 4, the theoretical analysis of the proposed distributed algorithm including convergence and optimality is presented. Section 5 reports simulation results of the proposed algorithm and its comparison to the existing distributed MWIS algorithms. Finally, concluding remarks are given in Section 6.

2. Notation

This section introduces some notations which will be used throughout this paper. The symbols used in this paper are summarized in the List of Symbols.

2.1. Partial Solution. Let $X = \{x_n,\ 1 \le n \le |V|\}$ denote the full variable set of the MWIS problem. Assuming Y is a subset of X, a partial solution of the MWIS problem is expressed as

$$p = \left\{x_k = v_k,\ x_k \in Y\right\}, \tag{2}$$

where v_k is 0 or 1. We call Y the partial variable set of the MWIS problem. The objective associated with p is expressed as

$$V_p = \sum_{x_k \in Y} w_k \cdot v_k. \tag{3}$$

Further, given a partial variable set Y, the partial solution set can be expressed as

$$\Omega = \left\{p_j,\ 1 \le j \le |\Omega|\right\}, \tag{4}$$

where

$$p_j = \left\{x_k = v_{j,k},\ x_k \in Y\right\}. \tag{5}$$

To help understand this, we use the graph shown in Figure 1 as an example. For this graph, there are four nodes and the weights are $w_1 = 3$, $w_2 = 4$, $w_3 = 5$, and $w_4 = 5$. The full variable set $X = \{x_1, x_2, x_3, x_4\}$. Assuming the partial variable set $Y = \{x_1, x_2\} \subseteq X$, then $p = \{x_1 = 1,\ x_2 = 0\}$ is a partial solution of the MWIS problem. The objective associated with this partial solution is $V_p = w_1 \cdot x_1 + w_2 \cdot x_2 = 3$.

2.2. Combination. For any two partial solutions $p_1 = \{x_k = v_k,\ x_k \in Y_1\}$ and $p_2 = \{x_l = u_l,\ x_l \in Y_2\}$, if there exists a common variable $x_i \in Y_1 \cap Y_2$ so that $v_i \ne u_i$, then we say these two partial solutions are contradictory; otherwise, if there does not exist such common variable, we say these two partial solutions are compatible. For more than two partial solutions, if there exist two partial solutions which are contradictory, we say these partial solutions are contradictory; otherwise, we say these partial solutions are compatible.

Then, for a number of partial solutions p_j, $1 \le j \le J$, the combination operation over them is defined as follows:

$$\bigotimes_{1 \le j \le J} p_j \triangleq \begin{cases} \emptyset, & \text{they are contradictory} \\ \bigcup\limits_{1 \le j \le J} p_j, & \text{they are compatible.} \end{cases} \tag{6}$$

Finally, for a number of partial solution sets $\Omega_i = \{p_{i,j},\ 1 \le j \le |\Omega_i|\}$, $1 \le i \le I$, the combination operation over them is defined as

$$\bigotimes_{1 \le i \le I} \Omega_i \triangleq \left\{ \bigotimes_{1 \le i \le I} p_{i,j},\ p_{i,j} \in \Omega_i \right\}. \tag{7}$$

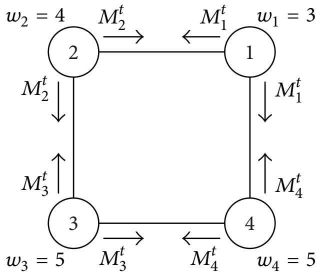


Figure 1: An example of the graph.

We continue to use the graph shown in Figure 1 as the example. For convenience, let $p_1 = \{x_1 = 0,\ x_2 = 0\}$, $p_2 = \{x_1 = 1,\ x_2 = 0\}$, $p_3 = \{x_1 = 0,\ x_4 = 0\}$, and $p_4 = \{x_1 = 1,\ x_4 = 0\}$. Firstly, for p_2 and p_3, since x_1 has different values in p_2 and p_3 (i.e., $x_1 = 1$ in p_2 and $x_1 = 0$ in p_3), we say p_2 and p_3 are contradictory and we have $p_2 \otimes p_3 = \emptyset$. Secondly, for p_2 and p_4, since there is no common variable which has different values in p_2 and p_4, we say p_2 and p_4 are compatible and we have $p_2 \otimes p_4 = \{x_1 = 1,\ x_2 = 0,\ x_4 = 0\}$. Thirdly, assuming $\Omega_1 = \{p_1, p_2\}$ and $\Omega_2 = \{p_3, p_4\}$, it can be verified that $\Omega_1 \otimes \Omega_2 = \{\{x_1 = 0,\ x_2 = 0,\ x_4 = 0\}, \{x_1 = 1,\ x_2 = 0,\ x_4 = 0\}\}$.

2.3. Truncation. For a partial solution set $\Omega = \{p_j,\ 1 \le j \le |\Omega|\}$, sort the elements in Ω by the values of the objective so that $V_{p_1} \ge V_{p_2} \ge \cdots \ge V_{p_{|\Omega|}}$. Define the truncation operation over Ω as

$$\operatorname{Trunc}(\Omega, H) \triangleq \{p_j, 1 \le j \le \min(|\Omega|, H)\}, \quad (8)$$

where H is a parameter.

We continue to use the graph shown in Figure 1 as the example. Assume $\Omega = \{p_1, p_2, p_3\}$, where $p_1 = \{x_1 = 0,\ x_2 = 1\}$, $p_2 = \{x_1 = 1,\ x_2 = 0\}$, and $p_3 = \{x_1 = 0,\ x_2 = 0\}$. It can be verified that $V_{p_1} = 4$, $V_{p_2} = 3$, and $V_{p_3} = 0$. Therefore, we have $V_{p_1} \ge V_{p_2} \ge V_{p_3}$. If we set $H = 2$, then we have $\operatorname{Trunc}(\Omega, H) = \{p_1, p_2\}$. That is, p_3 is deleted from Ω.

2.4. Feasible Partial Solution. Assuming Y is a subset of X, define the associated constraint set as

$$T(Y) = \{x_i + x_k \le 1,\ x_i \in Y,\ k \in N_i\}, \quad (9)$$

which contains all constraints involving variables in Y. Then, define the associated feasible partial solution set

$$F(Y, T(Y)) = \{\mathbf{v}_1 \colon \mathbf{v}_1 \succ T(Y)\}, \quad (10)$$

where $\mathbf{v}_1 = (v_k : x_k \in Y)$ and $\mathbf{v}_1 \succ T(Y)$ represents the notion that $\mathbf{v}_1$ satisfies the constraints in $T(Y)$. We have the following two lemmas.

Lemma 1. *For two subsets Y_1 and Y_2, one has*

$$\begin{aligned} &F(Y_1, T(Y_1)) \otimes F(Y_2, T(Y_2)) \\ &\quad = F(Y_1 \cup Y_2, T(Y_1 \cup Y_2)). \end{aligned} \quad (11)$$

Proof. See Appendix A. □

Lemma 2. *For K subsets Y_i, $1 \le i \le K$, one has*

$$\bigotimes_{1 \le i \le K} F(Y_i, T(Y_i)) = F\left(\bigcup_{1 \le i \le K} Y_i, T\left(\bigcup_{1 \le i \le K} Y_i\right)\right). \quad (12)$$

Proof. See Appendix B. □

We continue to use the graph shown in Figure 1 as the example. For two subsets $Y_1 = \{x_1\}$ and $Y_2 = \{x_4\}$, we have $T(Y_1) = \{x_1 + x_2 \le 1,\ x_1 + x_4 \le 1\}$ and $T(Y_2) = \{x_1 + x_4 \le 1,\ x_3 + x_4 \le 1\}$. On the one hand, there are five elements in $F(Y_1, T(Y_1))$, $(0,0,0)$, $(0,0,1)$, $(0,1,0)$, $(0,1,1)$, and $(1,0,0)$, and five elements in $F(Y_2, T(Y_2))$, $(0,0,0)$, $(0,0,1)$, $(0,1,0)$, $(1,0,0)$, and $(1,1,0)$. It can be verified that $F(Y_1, T(Y_1)) \otimes F(Y_2, T(Y_2))$ contains eight elements: $(0,0,0,0)$, $(0,0,1,0)$, $(0,0,0,1)$, $(0,1,0,0)$, $(0,1,1,0)$, $(0,1,0,1)$, $(1,0,0,0)$, and $(1,0,1,0)$. On the other hand, we have $Y_1 \cup Y_2 = \{x_1, x_4\}$ and $T(Y_1 \cup Y_2) = \{x_1 + x_2 \le 1,\ x_1 + x_4 \le 1,\ x_3 + x_4 \le 1\}$. It can be verified that $F(Y_1 \cup Y_2, T(Y_1 \cup Y_2))$ also contains the same eight elements. Therefore, Lemma 1 is correct for this example.

3. The Proposed Algorithm

The proposed algorithm is performed in an iterative manner. Let N_i denote the set of neighbors of node i. At each iteration t, node i sends a message M_i^t to each neighbor in N_i, where message M_i^t is actually a partial solution set. The details of the proposed distributed algorithm are presented as follows.

3.1. Algorithm. The operation of node i consists of two steps, which are presented in sequence as follows.

Step 1 (the combination operation). Node i receives the message M_j^{t-1} from each neighbor $j \in N_i$ and then generates a partial solution set as

$$Q_i^t = \bigotimes_{j \in N_i \cup \{i\}} M_j^{t-1}. \quad (13)$$

Specifically, for $t = 1$, we set

$$Q_i^1 = F(X_i^1, T(X_i^1)), \quad (14)$$

where

$$X_i^1 = \{x_i\} \cup \{x_k,\ k \in N_i\}. \quad (15)$$

For convenience, we express Q_i^t as

$$Q_i^t = \{p_j, 1 \le j \le |Q_i^t|\}, \quad (16)$$

where X_i^t is the partial variable set associated with node i in the tth iteration.

Step 2 (the truncation operation). The number of elements of Q_i^t can be very large. A parameter H is introduced to control

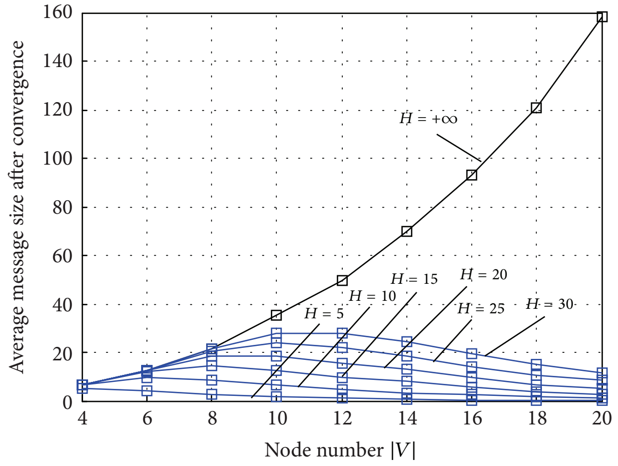


Figure 2: The average message size performance.

the space complexity. Node i only keeps H out of the total $|Q_i^t|$ elements and generates the message

$$M_i^t = \operatorname{Trunc}\left(Q_i^t, H\right), \tag{17}$$

where H is a parameter known by each node. Node i broadcasts the message M_i^t to its neighbors.

Each node shall repeat the above steps until convergence. As proved in Section 4 (Theorem 5), this process will converge after finite iterations. Upon convergence, assume the message node i sending to each neighbor is M_i. Let X_i denote the partial variable set associated with M_i. We can write $M_i = \{p_j, 1 \le j \le |M_i|\}$ and $p_j = \{x_k = v_{j,k}, x_k \in X_i\}$. Node i shall determine

$$j^* = \arg\max_j V_{p_j} \tag{18}$$

and estimate

$$x_i^* = v_{j^*,i}. \tag{19}$$

If we set $H = +\infty$, node i will not delete any partial solution in the truncation operation, and the algorithm will converge to the optimal solution as proved in Section 4 (Theorem 6). However, the message size will increase exponentially with the node number $|V|$ as shown in Section 5 (Figure 2). On the other hand, for general H, node i will delete a part of the partial solutions in the truncation operation to keep the message size small, and the algorithm will converge to an approximate solution with acceptable loss of optimality as shown in Section 5 (Figures 3 and 4). Therefore, the proposed algorithm can achieve different tradeoff between optimality and complexity.

3.2. Example 1: The Synchronous Case. To illustrate the above procedure, we use the graph shown in Figure 1 as an example. It can be verified that the maximum weight independent set is {node 2, node 4} (i.e., $x_1 = 0$, $x_2 = 1$, $x_3 = 0$, and $x_4 = 1$) and the associated objective is 9. For this example, we assume all nodes operate synchronously.

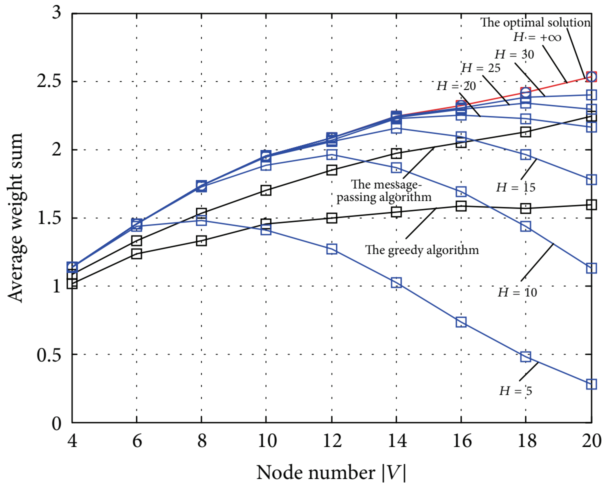


Figure 3: The average weight sum performance.

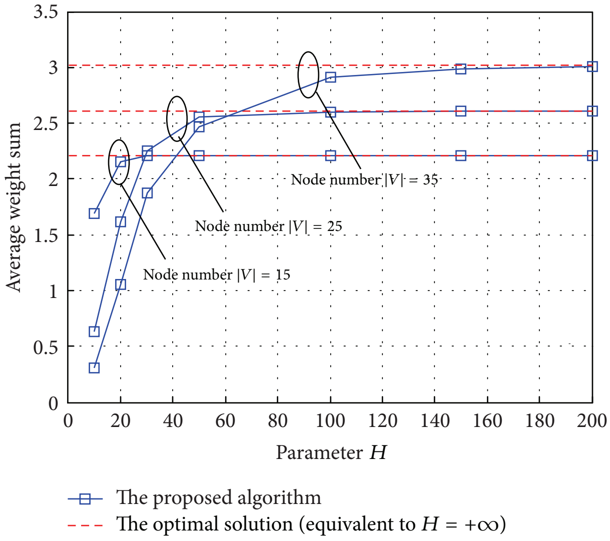


Figure 4: The average weight sum performance.

For each node i, it will send message M_i^t to its neighbors at the tth iteration. We will assume $H = +\infty$ and $H = 3$, respectively.

Case 1 ($H = +\infty$). We take node 1 as the example. Initially, $t = 1$, message Q_1^1 contains five elements: $(0,0,0)$, $(0,0,1)$, $(0,1,0)$, $(0,1,1)$, and $(1,0,0)$. Then, node 1 performs the truncation operation to generate M_1^1. Since $H = +\infty$, we have $M_1^1 = Q_1^1$, as shown in Table 1(a). For $t = 2$, since $H = +\infty$, node 1 has $M_1^2 = Q_1^2 = M_1^1 \otimes M_2^1 \otimes M_4^1$ in which there are seven elements, $(0,0,0,0)$, $(0,0,0,1)$, $(0,0,1,0)$, $(0,1,0,0)$, $(0,1,0,1)$, $(1,0,0,0)$, and $(1,0,1,0)$, as shown in Table 1(b). For $t \ge 3$, the set M_1^t does not change any more. We pick the solution with the highest objective, that is, the solution $x_1 = 0$, $x_2 = 1$, $x_3 = 0$, and $x_4 = 1$, which is exactly the optimal solution.

Table 1: An example with $H = +\infty$.

(a)

	x_1	x_2	x_3	x_4
		$t = 1$		
M_1^t	0	0		0
	0	0		1
	0	1		0
	0	1		1
	1	0		0
M_2^t	0	0	0	
	0	0	1	
	0	1	0	
	1	0	0	
	1	0	1	
M_3^t		0	0	0
		0	0	1
		0	1	0
		1	0	0
		1	0	1
M_4^t	0		0	0
	0		0	1
	0		1	0
	1		0	0
	1		1	0

(b)

	x_1	x_2	x_3	x_4
		$t = 2$		
M_1^t	0	0	0	0
	0	1	0	0
	0	0	1	0
	0	1	0	0
	0	1	0	1
	1	0	0	0
	1	0	1	0
M_2^t	0	0	0	0
	0	1	0	0
	0	0	1	0
	0	1	0	0
	0	1	0	1
	1	0	0	0
	1	0	1	0
M_3^t	0	0	0	0
	0	1	0	0
	0	0	1	0
	0	1	0	0
	0	1	0	1
	1	0	0	0
	1	0	1	0
M_4^t	0	0	0	0
	0	1	0	0
	0	0	1	0
	0	1	0	0
	0	1	0	1
	1	0	0	0
	1	0	1	0

Table 2: An example with $H = 3$.

(a)

	x_1	x_2	x_3	x_4
		$t = 1$		
M_1^t	0	0		1
	0	1		0
	0	1		1
M_2^t	0	0	1	
	0	1	0	
	1	0	1	
M_3^t		0	0	1
		0	1	0
		1	0	1
M_4^t	0		0	1
	0		1	0
	1		1	0

(b)

	x_1	x_2	x_3	x_4
		$t = 2$		
M_1^t	0	1	0	1
M_2^t	0	1	0	1
M_3^t	1	0	1	0
	0	0	1	0
	0	1	0	1
M_4^t	0	0	0	1
	0	1	0	1

Case 2 ($H = 3$). We still take node 1 as the example. Initially, $t = 1$, message Q_1^1 contains five elements: $(0, 0, 0)$, $(0, 0, 1)$, $(0, 1, 0)$, $(0, 1, 1)$, and $(1, 0, 0)$. Then, node 1 performs the truncation operation to generate M_1^1. Since $H = 3$, there are three elements in M_1^1, $(0, 0, 1)$, $(0, 1, 0)$, and $(0, 1, 1)$, as shown in Table 2(a). For $t = 2$, after similar operations, there is only one element in M_1^2, $(0, 1, 0, 1)$, as shown in Table 2(b). For $t \geq 3$, the set M_1^t does not change any more. We pick the solution with the highest objective, that is, the solution $x_1 = 0$, $x_2 = 1$, $x_3 = 0$, and $x_4 = 1$, which is exactly the optimal solution.

Comparing these two cases, we can observe that, for $H = 3$, the node will delete some partial solutions in the truncation operation to keep the message size small.

3.3. Example 2: The Asynchronous Case. The proposed algorithm can be run fully asynchronously. This is due to the fact that the combination operation, which is the core of the algorithm, does not require all nodes to act synchronously. We use the graph shown in Figure 1 again as the example. For this example, we assume node 1 is the one which is not synchronized with other nodes. Specifically, we assume that node 1 will not generate messages until $t = 2$. We use the proposed algorithm to solve this problem. To focus on the problem raised in this comment, we assume $H = +\infty$ so that the proposed algorithm can be expressed as the following iterative equation:

$$M_i^t = \bigotimes_{j\in N_i\cup\{i\}} M_j^{t-1}, \tag{20}$$

where M_i^t is the message broadcasted by node i in the tth iteration. The running process of the algorithm is described as follows.

Initially, for $t = 1$, according to (14) and (15), the message generated by each node can be expressed as $M_1^1 = \emptyset$, $M_2^1 = F(X_2^1, T(X_2^1))$, $M_3^1 = F(X_3^1, T(X_3^1))$, and $M_4^1 = F(X_4^1, T(X_4^1))$. Therefore, node 1 generates message M_1^1 which contains no elements, node 2 generates message M_2^1 which contains five elements, $(x_1, x_2, x_3) = (0, 0, 0)$, $(x_1, x_2, x_3) = (0, 0, 1)$, $(x_1, x_2, x_3) = (0, 1, 0)$, $(x_1, x_2, x_3) = (1, 0, 0)$, and $(x_1, x_2, x_3) = (1, 0, 1)$, and so on, as shown in Table 3(a).

Then, for $t = 2$, according to (20), the message generated by each node can be expressed as $M_1^2 = F(X_1^1, T(X_1^1))$, $M_2^2 = M_2^1 \otimes M_3^1$, $M_3^2 = M_3^1 \otimes M_2^1 \otimes M_4^1$, and $M_4^2 = M_4^1 \otimes M_3^1$. It can be verified that M_1^1 contains five elements: $(x_1, x_2, x_4) = (0, 0, 0)$, $(x_1, x_2, x_4) = (0, 0, 1)$, $(x_1, x_2, x_4) = (0, 1, 0)$, $(x_1, x_2, x_4) = (0, 1, 1)$, and $(x_1, x_2, x_4) = (1, 0, 0)$; M_2^2 contains eight elements; M_3^2 contains seven elements; and M_4^2 contains eight elements, as shown in Table 3(b).

For $t \geq 3$, according to (20), the message generated by each node can be expressed as $M_1^t = M_1^{t-1} \otimes M_2^{t-1} \otimes M_4^{t-1}$, $M_2^t = M_2^{t-1} \otimes M_3^{t-1} \otimes M_1^{t-1}$, $M_3^t = M_3^{t-1} \otimes M_2^{t-1} \otimes M_4^{t-1}$, and $M_4^t = M_4^{t-1} \otimes M_3^{t-1} \otimes M_1^{t-1}$. It can be verified that the message M_1^t contains seven elements, and so on, as shown in Table 3(c). We pick the solution with the highest objective, that is, the solution $(x_1, x_2, x_3, x_4) = (0, 1, 0, 1)$, which is exactly the optimal solution.

4. Theoretical Analysis

This section presents the theoretical analysis results of the proposed distributed algorithm.

Recall that X_i^t is the partial variable set associated with node i in the tth iteration. Firstly, we have the following two lemmas.

Lemma 3. *For general H, one has*

$$X_i^t = \bigcup_{k\in N_i\cup\{i\}} X_k^{t-1}, \tag{21}$$

$$M_i^t \subseteq F\left(X_i^t, T\left(X_i^t\right)\right). \tag{22}$$

Proof. We prove these two equations by induction on t. Initially, $t = 1$, according to the algorithm, we have $M_i^1 = \mathrm{Trunc}(Q_i^1, H) \subseteq Q_i^1 = F(X_i^1, T(X_i^1))$, where $X_i^1 = \{x_i\} \cup \{x_k,\ k \in N_i\}$. Therefore, this lemma holds for $t = 1$. Suppose this lemma is correct for $t \geq 2$. For $t + 1$, we have $M_i^{t+1} \subseteq Q_i^{t+1} = \otimes_{k\in N_i\cup\{i\}} M_k^t$. Assuming this lemma is correct for $t \geq 2$, we have $Q_i^{t+1} \subseteq \otimes_{k\in N_i\cup\{i\}} F(X_k^t, T(X_k^t))$. Applying Lemma 2, we then have $Q_i^{t+1} \subseteq F(\cup_{k\in N_i\cup\{i\}} X_k^t, T(\cup_{k\in N_i\cup\{i\}} X_k^t))$. Since X_i^{t+1} is the partial variable set associated with node i in the

Table 3: An asynchronous example.

(a)

	x_1	x_2	x_3	x_4
		$t = 1$		
M_1^t				
M_2^t	0	0	0	
	0	0	1	
	0	1	0	
	1	0	0	
	1	0	1	
M_3^t		0	0	0
		0	0	1
		0	1	0
		1	0	0
		1	0	1
M_4^t	0		0	0
	0		0	1
	0		1	0
	1		0	0
	1		1	0

(b)

	x_1	x_2	x_3	x_4
		$t = 2$		
M_1^t	0	0		0
	0	0		1
	0	1		0
	0	1		1
	1	0		0
M_2^t	0	0	0	0
	1	0	0	0
	0	0	0	1
	1	0	0	1
	0	0	1	0
	1	0	1	0
	0	1	0	0
	0	1	0	1
M_3^t	0	0	0	0
	0	1	0	0
	0	0	0	1
	0	1	0	1
	0	0	1	0
	1	0	0	0
	1	0	1	0
M_4^t	0	0	0	0
	1	0	0	0
	0	0	0	1
	0	0	1	0
	1	0	1	0
	0	1	0	0
	1	1	0	0
	0	1	0	1

(c)

	x_1	x_2	x_3	x_4
		$t = 3$		
M_1^t	0	0	0	0
	0	0	1	0
	0	0	0	1
	0	1	0	0
	0	1	0	1
	1	0	0	0
	1	0	1	0
M_2^t	0	0	0	0
	0	0	1	0
	0	0	0	1
	0	1	0	0
	0	1	0	1
	1	0	0	0
	1	0	1	0
M_3^t	0	0	0	0
	0	1	0	0
	0	0	0	1
	0	1	0	1
	0	0	1	0
	1	0	0	0
	1	0	1	0
M_4^t	0	0	0	0
	0	0	1	0
	0	0	0	1
	0	1	0	0
	0	1	0	1
	1	0	0	0
	1	0	1	0

$(t + 1)$th iteration, we know that $X_i^{t+1} = \cup_{k \in N_i \cup \{i\}} X_k^t$, which means that (21) is correct for $t + 1$. Further, we have $M_i^{t+1} \subseteq Q_i^{t+1} \subseteq F(X_i^{t+1}, T(X_i^{t+1}))$, which means that (22) is correct for $t + 1$. This completes the proof of this lemma. □

Lemma 4. *For $H = +\infty$, one has*

$$M_i^t = F\left(X_i^t, T\left(X_i^t\right)\right). \tag{23}$$

Proof. We prove this equation by induction on t. Initially, $t = 1$, according to the algorithm, since $H = +\infty$, we have $M_i^1 = Q_i^1 = F(X_i^1, T(X_i^1))$, where $X_i^1 = \{x_i\} \cup \{x_k, k \in N_i\}$. Therefore, this lemma holds for $t = 1$. Suppose this lemma is correct for $t \geq 2$. For $t + 1$, we firstly have $Q_i^{t+1} = \otimes_{k \in N_i \cup \{i\}} M_k^t$. Assuming this lemma is correct for $t \geq 2$, we have $Q_i^{t+1} = \otimes_{k \in N_i \cup \{i\}} F(X_k^t, T(X_k^t))$. Applying Lemma 2, we then have $Q_i^{t+1} = F(\cup_{k \in N_i \cup \{i\}} X_k^t, T(\cup_{k \in N_i \cup \{i\}} X_k^t))$. Since X_i^{t+1} is the partial variable set associated with node i in the $(t + 1)$th iteration, we have $X_i^{t+1} = \cup_{k \in N_i \cup \{i\}} X_k^t$. Since $H = +\infty$, we finally have $M_i^{t+1} = Q_i^{t+1} = F(X_i^{t+1}, T(X_i^{t+1}))$, which means that (23) is correct for $t + 1$. This completes the proof of this lemma. □

Based on these two lemmas, we have the following two theorems. Let $\text{diam}(G)$ denote the diameter of the graph G.

Theorem 5. *For general H, the proposed algorithm will converge to a feasible solution after* $2 \times \text{diam}(G) + 1$ *iterations.*

Proof. According to (21), X_i^t contains nodes to which the shortest distance from i is t. Since $\text{diam}(G)$ is the longest distance between any two nodes, we have $X_i^t = X$ for $t \geq \text{diam}(G) + 1$. Therefore, since $t = \text{diam}(G) + 2$, we have $M_i^t = \otimes_{k \in N_i \cup \{i\}} M_k^{t-1}$. For convenience, define $\widetilde{M}_i \triangleq M_i^{\text{diam}(G)+1}$. Therefore, since $t = 2 \times \text{diam}(G) + 1$, we have $M_i^t = \otimes_{k \in V} \widetilde{M}_k$ which will not change any more. Therefore, the proposed algorithm will converge after $2 \times \text{diam}(G) + 1$ iterations.

According to (22), due to $X_i^t = X$ for $t \geq \text{diam}(G) + 1$, we have $M_i^{\text{diam}(G)+1} \subseteq F(X, T(X))$ for $t \geq \text{diam}(G) + 1$. That is, each element in $\widetilde{M}_i$ is a feasible full solution. Therefore, for $t \geq 2 \times \text{diam}(G) + 1$, each element in $M_i^t = \otimes_{k \in V} \widetilde{M}_k$ is a feasible full solution. According to the algorithm, node i will estimate $\mathbf{x}^* = \arg\max_{\mathbf{x} \in M_i^t} \mathbf{w}^H \mathbf{x}$, where $\mathbf{x} = (x_1, \ldots, x_N)$ and $\mathbf{w} = (w_1, \ldots, w_N)$. Since each element contained in M_i^t is a feasible full solution of the MWIS problem, $\mathbf{x}^*$ must be feasible. Therefore, the proposed algorithm will converge to a feasible full solution after $2 \times \text{diam}(G) + 1$ iterations. □

Theorem 6. *For $H = +\infty$, the proposed algorithm will converge to the optimal solution after* $\text{diam}(G) + 1$ *iterations.*

Proof. For $t \geq \text{diam}(G) + 1$, since $\text{diam}(G)$ is the longest distance between any two nodes, we have $X_i^t = X$. Thus, according to (23), for $t \geq \text{diam}(G) + 1$, we have $M_i^t = F(X, T(X))$. Therefore, the algorithm is converged after $\text{diam}(G) + 1$ iterations. Since node i will estimate $\mathbf{x}^* = \arg\max_{\mathbf{x} \in F(X,T(X))} \mathbf{w}^H \mathbf{x}$, where $F(X, T(X))$ covers all feasible solutions of the MWIS problem, $\mathbf{x}^*$ must be the optimal one. Therefore, the proposed algorithm will converge to the optimal solution after $\text{diam}(G) + 1$ iterations. □

Theoretical analysis results show that the proposed algorithm is guaranteed to converge to an independent set and the tradeoff between approximation accuracy and algorithm complexity is controlled by parameter H, which are confirmed by the simulation results reported in Section 5.

5. Performance Evaluation

In this section, we evaluate the proposed algorithm via computer simulations. We first describe the simulation methodology and then present and analyze the simulation results.

We develop a simulator based on MATLAB. In the simulator, a total of $|V|$ nodes are randomly distributed over a 10-meter × 10-meter flat field. For each node, the weight is a random number ranging from 0 to 1. For any two nodes, if the distance is less than 6 meters, there will be an edge connecting them. Thus, we have generated a general graph. For this graph, we use five methods to find its MWIS. The first is to use the full version of the proposed algorithm, the second is to use the general version of the proposed algorithm with parameter H ranging from 5 to 30, the third is to use the greedy algorithm proposed in [19], the fourth is to use the message-passing algorithm proposed in [30], and the fifth is to use MATLAB function bintprog to solve the integer program in (1). For the first two methods, we collect four metrics that characterize the performance of the proposed algorithm: (1) the weight sum of the output set, (2) the iteration number needed to converge, (3) the diameter of the graph, and (4) the message size, that is, the number of ME contained in each message, upon convergence. For the other two methods, we only collect the weight sum of the output set. The procedure is repeated 1000 times for each value of $|V|$, each time with a new generated graph and weights, and results are averaged out.

We first report that the proposed algorithm is convergent for all simulation runs, and the output set is always independent. Then, we present the simulation results obtained from computer experiments, which are organized in four figures. Figure 2 shows the average message size upon convergence as the node number $|V|$ varied. Specifically, it is worth mentioning that since the message-passing algorithm does not always converge and the output set is not always independent, we only collect the results when the message-passing algorithm converges and the output set is independent. In this figure, for $H = +\infty$, the message size grows exponentially with node number, while for general H the message size is well controlled. Figures 3 and 4 show the average weight sum performance as the node number $|V|$ and parameter H varied. In these two figures, the proposed algorithm with general H will converge to an approximate solution with acceptable loss of optimality as parameter H is appropriately selected. Additionally, in Figure 3, the curve corresponding to $H = +\infty$ coincides with the curve of the optimal solution. Thus, the optimality part of Theorem 6 is verified. Finally, Figure 5 shows the average iteration number needed to converge as the node number $|V|$ varied. In this figure, the curve corresponding to $H = +\infty$ is always under the upper bound predicted by Theorem 6 (i.e., the diameter plus 1), while the curves corresponding to the general H are always under the upper bound predicted by Theorem 5 (i.e., two times the diameter plus 1). Thus, Theorem 5 and the convergence part of Theorem 6 are verified. Simulation results show that the proposed algorithm with $H = +\infty$ is a distributed optimal algorithm, while the proposed algorithm with general H is a distributed approximate algorithm which by changing parameter H can achieve different tradeoff between approximation accuracy and space complexity.

Finally, we empirically study how to set value for the parameter H. Actually, we need a way to estimate a good H for a given weighted graph. We propose to set H as the integer times of the number of nodes. The simulation results are summarized in Table 4. According to the simulation results, we suggest to set H as 2–4 times the number of nodes.

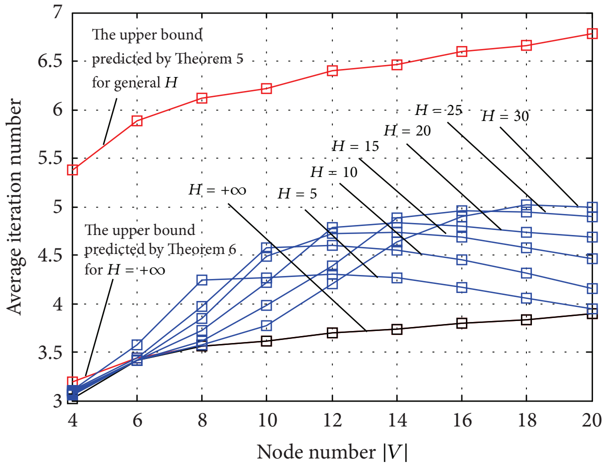


FIGURE 5: The average iteration number performance.

TABLE 4: Simulation results with different H.

$\|V\|$	H	The final solution	The optimal solution
$\|V\| = 15$	$H = 1 \cdot \|V\|$	1.79	2.21
	$H = 2 \cdot \|V\|$	2.21	
	$H = 4 \cdot \|V\|$	2.21	
$\|V\| = 25$	$H = 1 \cdot \|V\|$	1.62	2.61
	$H = 2 \cdot \|V\|$	2.55	
	$H = 4 \cdot \|V\|$	2.60	
$\|V\| = 35$	$H = 1 \cdot \|V\|$	1.87	3.02
	$H = 2 \cdot \|V\|$	2.56	
	$H = 4 \cdot \|V\|$	2.99	

6. Conclusions

A new distributed algorithm for finding the maximum weight independent set in a general graph is presented. The proposed algorithm runs iteratively in which each node receives a message form each neighbor, updates the message of its own, and sends it to each neighbor. It is shown that the proposed algorithm will converge to an independent set after finite iterations and can achieve different tradeoff between approximation accuracy and space complexity. Specifically, the proposed algorithm with $H = +\infty$ is guaranteed to converge to the optimal solution.

Appendix

A. Proof of Lemma 1

Proof. Assume $T(Y_1) = T_{11} \cup T_{12}$ and $T(Y_2) = T_{22} \cup T_{12}$, where $T_{12} = T(Y_1) \cap T(Y_2)$ and $T_{11} \cap T_{22} = \emptyset$. We have $F(Y_1, T(Y_1)) = \{\mathbf{v}_1 : \mathbf{v}_1 \succ T(Y_1)\} = \{\mathbf{v}_1 : \mathbf{v}_1 \succ T_{11}, \mathbf{v}_1 \succ T_{12}\}$ and $F(Y_2, T(Y_2)) = \{\mathbf{v}_2 : \mathbf{v}_2 \succ T(Y_2)\} = \{\mathbf{v}_2 : \mathbf{v}_2 \succ T_{22}, \mathbf{v}_2 \succ T_{12}\}$. According to the definition, we have $F(Y_1, T(Y_1)) \otimes F(Y_2, T(Y_2)) = \{\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2) : \mathbf{v}_1 \text{ and } \mathbf{v}_2 \text{ are compatible}\} = \{\mathbf{v} : \mathbf{v}_1 \succ T_{11}, \mathbf{v}_1 \succ T_{12}, \mathbf{v}_1 \succ T_{22}, \mathbf{v}_2 \succ T_{22}, \mathbf{v}_2 \succ T_{12}, \mathbf{v}_2 \succ T_{11}\} = \{\mathbf{v} : \mathbf{v} \succ T_{11} \cup T_{22} \cup T_{12}\} = F(Y_1 \cup Y_2, T(Y_1) \cup T(Y_2)) = F(Y_1 \cup Y_2, T(Y_1 \cup Y_2))$. This completes the proof of this lemma. □

B. Proof of Lemma 2

Proof. We prove this lemma by induction on K. According to Lemma 1, for $K = 2$, this lemma is correct. Suppose this lemma is correct for $K \geq 2$. By using Lemma 1 and hypothesis, we have $\otimes_{1\leq k\leq K} F(Y_k, T(Y_k)) \otimes F(Y_{K+1}, T(Y_{K+1})) = F(\cup_{1\leq k\leq K} Y_k, \cup_{1\leq k\leq K} T(Y_k)) \otimes F(Y_{K+1}, T(Y_{K+1})) = F(\cup_{1\leq k\leq K+1} Y_k, \cup_{1\leq k\leq K} T(Y_k) \cup \mathrm{T}(Y_{K+1})) = F(\cup_{1\leq k\leq K+1} Y_k, \cup_{1\leq k\leq K+1} T(Y_k))$. Thus, this lemma is also correct for $K + 1$. This completes the proof of this lemma. □

List of Symbols

$G = (V, E)$:	The graph
w_k:	The weight associated with each node
x_k:	The binary variable associated with each node
X:	The full variable set
p:	The partial solution
V_p:	The objective associated with p
Ω:	The partial solution set
$\otimes p_j$:	The combination operation over partial solutions $\{p_j\}$
$\otimes \Omega_i$:	The combination operation over partial solution set $\{\Omega_i\}$
$\mathrm{Trunc}(\Omega, H)$:	The truncation operation over Ω with parameter H
N_i:	The set of neighbors of node i
Y:	A subset of X
$T(Y)$:	The full constraint set associated with Y
$F(Y, T(Y))$:	The feasible solution set associated with Y
M_i^t:	The message broadcasted by node i in the tth iteration
Q_i^t:	The partial solution set generated by node i after the combination operation in the tth iteration
X_i^t:	The partial variable set associated with node i in the tth iteration
$\mathrm{diam}(G)$:	The diameter of graph G.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61571111).

References

[1] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982.

[2] M. Grotschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, Berlin, Germany, 1993.

[3] T. Matsui, "Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs," in *Proceedings of the Japan Conference on Discrete and Computational Geometry (JCDCG '98)*, pp. 194–200, Tokyo, Japan, December 1998.

[4] G. J. Minty, "On maximal independent sets of vertices in claw-free graphs," *Journal of Combinatorial Theory, Series B*, vol. 28, no. 3, pp. 284–304, 1980.

[5] V. E. Alekseev, "Polynomial algorithm for finding the largest independent sets in graphs without forks," *Discrete Applied Mathematics*, vol. 135, no. 1–3, pp. 3–16, 2004.

[6] G. H. Chen, M. T. Kuo, and J. P. Sheu, "An optimal time algorithm for finding a maximum weight independent set in a tree," *BIT Numerical Mathematics*, vol. 28, no. 2, pp. 353–356, 1988.

[7] R. Karp and M. Sipser, "Maximum matchings in sparse random graphs," in *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science (FOCS '81)*, pp. 364–375, Los Alamitos, Calif, USA, 1981.

[8] A. Czygrinow and M. Hanckowiak, "Distributed algorithms for weighted problems in sparse graphs," *Journal of Discrete Algorithms*, vol. 4, no. 4, pp. 588–607, 2006.

[9] G. Valiente, "A new simple algorithm for the maximum-weight independent set problem on circle graphs," in *Algorithms and Computation*, T. Ibaraki, N. Katoh, and H. Ono, Eds., vol. 2906 of *Lecture Notes in Computer Science*, pp. 129–137, Springer, New York, NY, USA, 2003.

[10] B. Gfeller and E. Vicari, "A randomized distributed algorithm for the maximal independent set problem in growth-bounded graphs," in *Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing (PODC '07)*, pp. 53–60, Portland, Ore, USA, August 2007.

[11] M. Halldorsson, "Approximations of independent sets in graphs," in *Approximation Algorithms for Combinatiorial Optimization*, pp. 24–45, Springer, Berlin, Germany, 2004.

[12] M. Furer and S. Kasiviswanathan, "Algorithms for counting 2-SAT solutions and colorings with applications," in *Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management (AAIM '07)*, pp. 47–57, 2007.

[13] F. V. Fomin, S. Gaspers, and S. Saurabh, "Branching and treewidth based exact algorithms," in *Algorithms and Computation*, T. Asano, Ed., vol. 4288 of *Lecture Notes in Computer Science*, pp. 16–25, Springer, New York, NY, USA, 2006.

[14] L. Babel, "A fast algorithm for the maximum weight clique problem," *Computing*, vol. 52, no. 1, pp. 31–38, 1994.

[15] P. R. Ostergard, "A fast algorithm for the maximum clique problem," *Discrete Applied Mathematics*, vol. 120, no. 1–3, pp. 197–207, 2002.

[16] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[17] S. Sarkar and S. Ray, "Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2307–2323, 2008.

[18] P. Wang and S. Bohacek, "Practical computation of optimal schedules in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 305–318, 2011.

[19] S. Basagni, "Finding a maximal weighted independent set in wireless networks," *Telecommunication Systems*, vol. 18, no. 1–3, pp. 155–168, 2001.

[20] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem,"

Discrete Applied Mathematics, vol. 126, no. 2-3, pp. 313–322, 2003.

[21] A. Kako, T. Ono, T. Hirata, and M. Halldsson, "Approximation algorithms for the weighted independent set problem," in *Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 341–350, 2005.

[22] C. Joo, X. Lin, J. Ryu, and N. B. Shroff, "Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1476–1488, 2016.

[23] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1846–1859, 2009.

[24] L. Jiang, D. Shah, J. Shin, and J. Walrand, "Distributed random access algorithm: scheduling and congestion control," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6182–6207, 2010.

[25] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2010.

[26] L. Jiang and J. Walrand, "Approaching throughput-optimality in distributed CSMA scheduling algorithms with collisions," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 816–829, 2011.

[27] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6541–6555, 2012.

[28] I. C. Paschalidis, F. Huang, and W. Lai, "A message-passing algorithm for wireless network scheduling," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1528–1541, 2015.

[29] J. Foulds, N. Navaroli, P. Smyth, and A. Ihler, "Revisiting MAP estimation, message passing and perfect graphs," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS '11)*, Fort Lauderdale, Fla, USA, April 2011.

[30] S. Sanghavi, D. Shah, and A. S. Willsky, "Message passing for maximum weight independent set," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 4822–4834, 2009.

[31] A. Weller, "Revisiting the limits of MAP inference by MWSS on perfect graphs," in *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS '15)*, San Diego, Calif, USA, May 2015.

[32] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[33] X.-Y. Wang and Y. Zhang, "An improved distributed algorithm for maximum weight independent set (MWIS) problem," *Journal of Electronics and Information Technology*, vol. 34, no. 3, pp. 689–693, 2012.

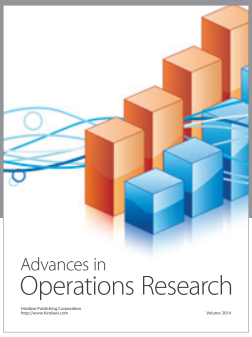
[34] P. Du, "A heuristic distributed algorithm for the maximum weight independent set problem," *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*, vol. 33, no. 6, pp. 18–28, 2013.

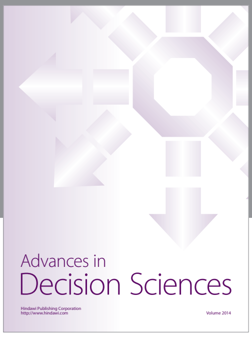
[35] S. Sanghavi, D. Malioutov, and A. Willsky, "Belief propagation and LP relaxation for weighted matching in general graphs," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2203–2212, 2011.

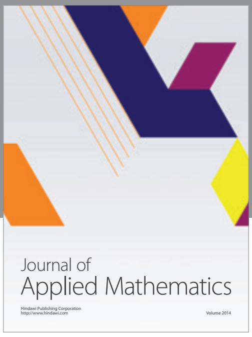
[36] Y. Zhang, "Distributed maximum weight matching algorithm based on belief propagation," *Journal on Communications*, vol. 33, no. 10, pp. 132–137, 2012.

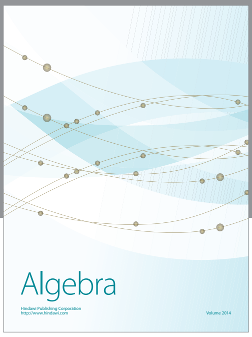
[37] I. C. Paschalidis, W. Lai, and X. Song, "Optimized scheduled multiple access control for wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2573–2585, 2009.

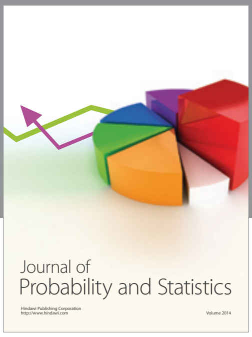
[38] I. C. Paschalidis, W. Lai, and D. Starobinski, "Asymptotically optimal transmission policies for large-scale low-power wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 105–118, 2007.


Advances in
Operations Research

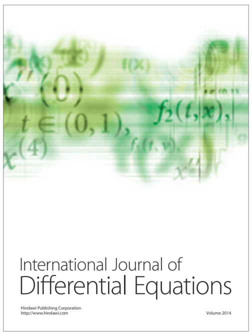

Advances in
Decision Sciences


Journal of
Applied Mathematics

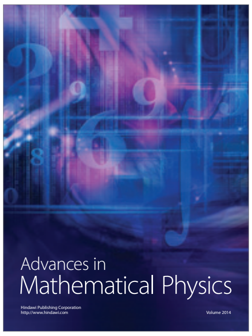

Algebra

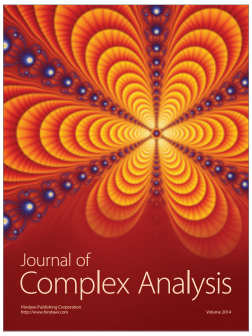

Journal of
Probability and Statistics

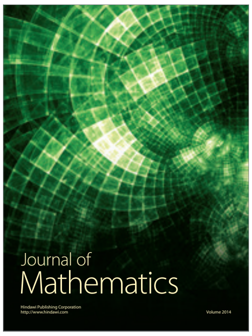
The Scientific
World Journal

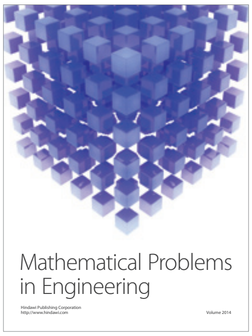

International Journal of
Differential Equations

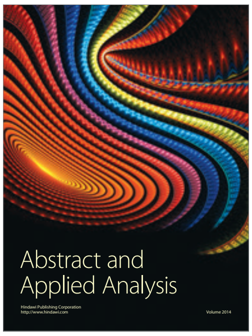
International Journal of
Combinatorics

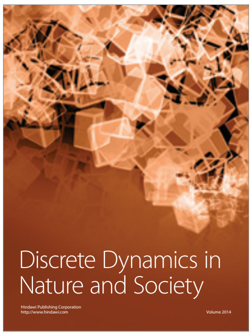

Advances in
Mathematical Physics

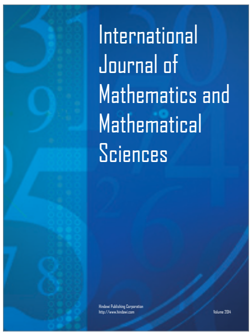

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

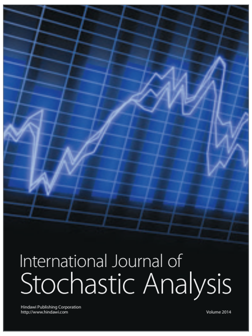

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization