Discrete Applied Mathematics 204 (2016) 173–184

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Binary linear programming models for robust broadcasting in communication networks

Ronald G. McGarvey [a,*], Brian Q. Rieksts [b], José A. Ventura [c,1], Namsu Ahn [d]

[a] Department of Industrial and Manufacturing Systems Engineering, Harry S. Truman School of Public Affairs, University of Missouri, E3437D Lafferre Hall, Columbia, MO 65211, United States

[b] Institute for Defense Analyses, Cost Analysis and Research Division, 4850 Mark Center Drive, Alexandria, VA 22311, United States

[c] Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, 310 Leonhard Building, University Park, PA 16802, United States

[d] Department of Industrial Management, Ulsan College, Ulsan, South Korea

ARTICLE INFO

Article history:
Received 5 April 2015
Received in revised form 4 November 2015
Accepted 8 November 2015
Available online 8 December 2015

Keywords:
Integer programming
Communication networks
Broadcasting
Fault-tolerant networks

ABSTRACT

Broadcasting is an information dissemination process in communication networks whereby a message, originated at any node of a network, is transmitted to all other nodes of the network. In c-broadcasting, each node having the message completes up to c transmissions to its neighbors over the communication lines in one time unit. In a k-fault tolerant c-broadcast network, the broadcasting process can be accomplished even if k communication lines fail. This paper presents innovative binary linear programming formulations to construct c-broadcast graphs, k-fault-tolerant c-broadcast graphs, and their time-relaxed versions. The proposed mathematical models are used to generate eight previously unknown minimum c-broadcast graphs, new upper bounds for eleven other instances of the c-broadcast problem, and over 30 minimum k-fault-tolerant c-broadcast graphs. The paper also provides a construction method to produce an upper bound for an infinite family of k-fault-tolerant c-broadcast graphs.

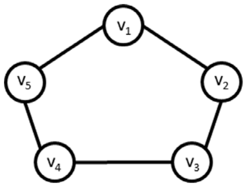
1. Introduction

Communication is a propagation process of information between a set of information producers and a set of information consumers through logical and physical media. Logical media refers to the protocol that is used, such as language, symbols, and numbers, while physical media represents the communication links, including cables, radio, and light. When there are more than two members involved in the propagation process, it is necessary to establish the scheme to link them satisfying certain goals. The linking scheme is the communication network. Thus, the topology of the network has a significant influence on its performance and operational cost. Examples of communication networks that need to be designed efficiently and economically are abundant in many application areas, such as computer networks, telecommunication networks, and interconnection networks [6].

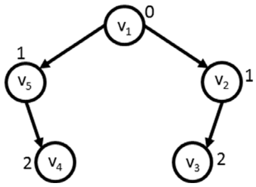
Broadcasting can be considered as the most basic information dissemination process. It is the process of disseminating a message from a given node (originator) to all other nodes in a network. After receiving the message, a node may transmit

* Corresponding author. Tel.: +1 573 882 9564.
E-mail addresses: mcgarveyr@missouri.edu (R.G. McGarvey), brieksts@ida.org (B.Q. Rieksts), jav1@psu.edu (J.A. Ventura), namsu.ahn@gmail.com (N. Ahn).
[1] Tel.: +1 814 865 3841; fax: +1 814 863 4745.

http://dx.doi.org/10.1016/j.dam.2015.11.008



(a) 2-mbg with five nodes.



(b) Broadcast protocol.

Fig. 1. 2-mbg with five nodes and its broadcast protocol.

the message to its adjacent nodes. In regular broadcasting [14], a node can transmit the message to only one adjacent node per time unit, where a time unit is defined as the transmission time of the message, but different pairs of nodes may communicate simultaneously. In c-broadcasting [21], which is a generalization of regular broadcasting, up to c adjacent nodes are permitted to receive the message from a single node per time unit.

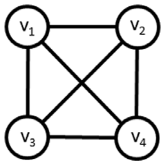
A communication network can be modeled as a connected undirected graph $G(V, E)$, where V is the set of nodes and E the set of communication links (edges). For a given graph G and originator $v \in V$, the minimum time to complete c-broadcasting from v is defined as $b_c(v)$. Thus, the c-broadcast time of graph G across all potential originator nodes is $b_c(G) = \max\{b_c(v)|v \in V\}$. Note that, if p nodes have the message at the beginning of a time unit, at most $(c+1)p$ nodes can have the message at the end of the time unit. Thus, for an arbitrary graph G of order n, i.e., $n = |V|$, the minimum possible value for $b_c(G)$ is $\lceil \log_{c+1} n \rceil$. A graph with this property is called a c-broadcast graph (c-bg). The minimum number of edges of a c-bg of order n is given by the c-broadcast function $B_c(n)$. A c-bg of order n with $B_c(n)$ edges is said to be a minimum c-broadcast graph (c-mbg). The message transmission process originated at node v in c-broadcasting determines a spanning tree. A *c-broadcast protocol* (or *c-broadcast tree*) is a rooted spanning tree with n nodes in which the originator v is the root and all nodes are labeled by their receiving times, which are equal to or less than $b_c(G)$.

Fig. 1 shows a 2-mbg with five nodes and its broadcast protocol. Since all nodes are topologically equivalent, the 2-broadcast protocol in Fig. 1(b) is valid for any originator. The number next to each node indicates the time that the node receives the message.

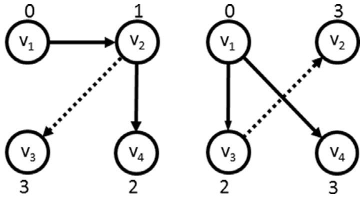
In addition to the c-broadcast function, some related problems will be discussed over the course of this paper. One such problem is time-relaxed c-broadcasting. For this case, additional time units are allowed by restricting $b_{t,c}(G) = \lceil \log_{c+1} n \rceil + t$ instead of $b_c(G) = \lceil \log_{c+1} n \rceil$. The function $B_{t,c}(n)$ is the minimum number of edges of a time-relaxed c-broadcast graph ((t, c)-rbg) of order n with $b_c(G) \le \lceil \log_{c+1} n \rceil + t$.

Another problem of interest is the k-fault tolerant (k-ft) broadcast problem [38]. The literature in this area has focused on single-transmission broadcasting, i.e., $c = 1$. These graphs are structured such that the message reaches its destination even in the event of permanent failures in no more than k communication links. The objective is to construct sparse graphs with reliable transmission schemes. The protocols of a k-ft broadcast graph are predefined in such a way that if any k edges in the protocol fail, the message will still reach all of the non-originators in the graph. If there is a common edge in all paths to a given node in some protocol, this edge could fail, preventing the node from receiving the message. Thus, a k-ft protocol must contain $k+1$ independent (edge disjoint) paths to each non-originator in the network. The constraints of the regular broadcast problem also apply to the k-ft broadcast problem. During a particular time unit, a node may not send and receive messages from two different nodes simultaneously according to Liestman [38]. In a message transfer between adjacent nodes, it is possible for a node to both send and receive the message from a single node during the same time unit. In this case, the message becomes scrambled if both nodes have the message, but this is irrelevant since they already have the message. If there is a failure and one of the nodes does not have the message, a transmission is sent from the node with the message to the node without the message. Liestman [38] proves $\lceil \log_2 n \rceil + 1$ is the minimum number of time units required for completing a 1-ft broadcast protocol in a network of three or more nodes. For the general k-ft problem, there is not a known closed form expression for the minimum broadcast time for every instance, but several results provide the minimum broadcast time for instances satisfying certain properties. For example, Liestman [38] proves $\lceil \log_2 n \rceil + 1$ is the minimum number of time units required for completing a 1-ft broadcast protocol in a network of three or more nodes. A summary of these closed form results for the k-ft problem is presented in [1].

A broadcast network that permits k-ft protocols from each originator in minimum time is called a k-fault tolerant broadcast graph (k-ftbg). Although the literature focuses on $c = 1$, in this paper we extend these concepts to address general k-ft c-broadcast graphs ((c, k)-bg's). The function $B_{c,k}(n)$ is defined as the minimum number of edges of all (c, k)-bg's with n nodes. We note that $B_{c,k=0}(n) = B_c(n)$. A graph with n nodes and $B_{c,k}(n)$ edges is called a k-fault-tolerant minimum c-broadcast graphs ((c, k)-mbg's). Fig. 2(a) provides an example of a $(1, 1)$-mbg with four nodes. In Fig. 2(b), a 1-ft protocol is displayed with node v_1 being the originator. This protocol is separated into two parts to show two independent paths to each node. During the first two time units, each node receives the message if failures do not occur, but it is necessary to continue to broadcast to provide two independent paths to each non-originator. During the last time unit (time unit three), a bi-directional exchange takes place between nodes v_2 and v_3 (denoted by the dashed lines), and a message is sent from node v_1 to node v_4. Since the protocol contains two independent paths to each non-originator, is completed in $\lceil \log_2 4 \rceil + 1 = 3$



(a) (1, 1)-mbg with four nodes.



(b) 1-ft protocol for originator v_1.

Fig. 2. (1, 1)-mbg with four nodes and its 1-ft protocol from originator v_1.

time units, and satisfies the broadcast criteria, it follows that the protocol in Fig. 2(b) is a 1-ft protocol. The graph shown in Fig. 2(a) is also symmetric allowing the 1-ft protocol from Fig. 2(b) to be valid from every possible originator.

The research in this paper focuses on constructing broadcast networks. Although these networks have been primarily considered as theoretical models, military command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) networks provide an application for these models. Dekker [9] discusses metrics for evaluating the effectiveness of military C4ISR networks. In these networks, intelligence gathering nodes typically send information to command and control nodes. The command and control nodes make decisions and issue orders to be carried out by force nodes. Delays of information flow are highlighted as a significant metric by Dekker [9]. Dekker [10] discusses an example where time delay in communicating instructions led to disaster for US forces in Mogadishu on October 3, 1993. Dekker [10] also points out that improved technology has increased the possible combinations of communication links between nodes. As Lee and Ghosh [35] indicate, raw information transmitted to every node would result in information overload. The goal for information flow in a C4ISR network is to disseminate information to the appropriate nodes quickly. Our research constructs networks and protocols to disseminate information quickly between any nodes potentially deemed appropriate for a particular mission.

In the remainder of this paper, we present new binary linear programming formulations to construct c-bg's, (c, k)-bg's, and their time-relaxed versions. Section 2 provides a literature review of techniques used to generate various types of broadcast graphs. Sections 3 and 5 include mathematical models to generate c-mbg's and (c, k)-mbg's, respectively. Some properties of (c, k)-bg's are derived in Section 4. Section 6 presents the results of a computational study that identifies new minimum broadcast graphs not previously identified in the literature. In Section 7, the paper concludes by suggesting possible extensions for future work.

2. Literature review

Hedetniemi and Hedetniemi [30] present a survey of several problems in broadcasting and the related area of gossiping. Harutyunyan, et al. [29] provide another survey of broadcasting and gossiping. Hromkovic, et al. [31] also published a book on these problems. Farley et al. [14] introduce the regular broadcast problem with results for some small values of $B_{c=1}(n)$. Harutyunyan and Liestman [24] present some of the best-known broadcast graphs for many values of n. In Farley [13], a recursive algorithm is presented for constructing 1-bg's with any number of nodes. The problem of recognizing whether an arbitrary graph is a 1-bg is *NP*-complete [17] according to Slater et al. [45]. Currently, exact values of $B_{c=1}(n)$ are known for $n = 2^p$, $n = 2^p - 2$, $n = 127$, $n = 1023$, $n = 4095$, and for several values of $n \leq 63$. Shao [43] finds values for $B_{c=1}(1023)$ and $B_{c=1}(4095)$. Averbuch et al. [4] describe the current state of research for exact values of $B_{c=1}(n)$, but note that much of the research has focused on constructing broadcast graphs. Improvements over many of the previously known upper bounds on $B_{c=1}(n)$ are presented by Gargano and Vaccaro [19]. The construction methods of their paper are based on hypercubes. Bermond et al. [8] provide four methods to construct sparse 1-bg's including node addition and node deletion. Other research to improve upper bounds on $B_{c=1}(n)$ has involved compounding algorithms by [7,11]. Researchers have continued to improve lower bounds in [5,22] while other researchers have found new upper bounds in [23,28,4].

Shastri [44] considers the time-relaxed problem. Shastri's research includes a general construction method for t-bg's in addition to the sparsest possible t-bg's with n nodes for $n \leq 14$. Yao et al. [48] show that $B_{t=1,c=1}(15) = 17$ instead of the conjectured value of $B_{t=1,c=1}(15) = 18$ by Shastri [44]. Dinneen et al. [12] extend their compounding algorithm for the regular broadcast problem to the time-relaxed case and later some of their results were independently found by Yao et al. [47]. Yao, et al. [47] also proves $B_{t=1,c=1}(22) = 24$ and provides some additional upper bounds for the time-relaxed problem. Averbuch et al. [3] also addresses the construction of time relaxed broadcast graphs.

Grigni and Peleg [21] introduce c-broadcasting by using the concept of conference-call, through which a node is allowed to send a message simultaneously to up to c neighbors at a time, for some constant $c \geq 1$. Konig and Lazard [33] provide c-mbg's for $c + 3 \leq n \leq 2c + 3$. Lee and Ventura [36] extend the compounding algorithm of Weng and Ventura [46] to c-broadcasting. Harutyunyan and Liestman [25] also present upper and lower bounds on $B_c(n)$. Shao [43] develops lower bounds on $B_c(n)$ as well. Upper bounds for the time-relaxed c-broadcast function are found in [27,37,2]. In [26], Harutyunyan and Liestman investigate c-broadcasting in trees. Liestman and Peters [39] introduce broadcasting in bounded degree graphs. Lazard [34] studies c-broadcasting in bounded-degree graphs and provides some c-mbg's for small values of n and c.

Another version of conference-call broadcasting is investigated by Richards and Liestman [42]. They study c-broadcasting in complete $(c+1)$-uniform hypergraphs, in which a conference call involves the nodes of some hyperedges. Farley and Proskurowski [16] study bounded-call broadcasting where the maximum number of calls from a node is a predetermined constant. They also investigate whispering ($c=1$) and shouting ($c=\infty$) [15].

Pelc [40] presents a survey of the research for various reliability problems in broadcasting and gossiping. Liestman [38] introduces the k-ft broadcast problem and discusses the tradeoff between the broadcast time and the number of edges required for a k-ftbg. In order to prove the minimum broadcast time for a 1-ftmbg is $\lceil \log_2 n \rceil + 1$, he develops a general construction algorithm for 1-ftmbg's. Although there is not a closed form expression for the minimum broadcast time for a general k-ftbg, bounds on this time have been developed by [18,20,38,41].

Gargano [18] provides a result that yields an infinite family of k-ftmbg's defined by $B_k(2^m) = (m+k)2^{m-1}$ for each $k \le 2^m - m,\ m \ge 3$. Khachatrian and Haroutunian [32] present another infinite class that states $B_{k=1}(2^m - 2) = m(2^{m-1} - 1)$. Ahlswede et al. [1] contribute several results to the 1-ft broadcast problem. Their research provides the remaining values of $B_{k=1}(n)$ for $n \le 16$. An infinite class is given as $B_{k=1}(2^m - 6) = [(m-1)(2^m - 6)]/2$ for each even m. A method based on broadcast trees is provided for constructing k-ftbg's with an arbitrary number of nodes for $k < \lfloor \log_2 n \rfloor$. A graph product construction method is also presented for compounding 1-bg's with hypercubes.

3. A binary linear programming model for c-broadcasting

In this section, we present a binary linear programming formulation to minimize the edge set cardinality of a c-bg G of order n with broadcast time $b_c(G) \le T$. If the desired result is a c-mbg, T is set to $\lceil \log_{c+1} n \rceil$. If the desired result is a (t, c)-rbg, T is set to $\lceil \log_{c+1} n \rceil + t$.

The binary variables of the model are defined as follows:

$$z_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ exists in the network,} \quad \text{for } (i,j) \in S_1, \\ 0 & \text{otherwise,} \end{cases}$$

where $S_1 = \{(i,j) : 1 \le i \le j-1,\ i+1 \le j \le n\}$.

$$x_{ijtu} = \begin{cases} 1 & \text{if node } i \text{ calls node } j \text{ at time } t \text{ for a given originator } u \text{ of the message,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{for } i \in S_2(j,u,t),\ j \in S_4(u),\ t \in S_5, \text{ and } u \in S_6,$$

where $S_2(j,u,t) = \{i : i = u \text{ and } i \ne j \text{ if } t = 1;\ 1 \le i \le n \text{ and } i \ne j \text{ if } t \ne 1\}$, $S_4(u) = \{j : 1 \le j \le n,\ j \ne u\}$, $S_5 = \{t : 1 \le t \le T\}$, and $S_6 = \{u : 1 \le u \le n\}$.

In addition define set,

$$S_3(j,u) = \{i : 1 \le i \le n, i \ne j, i \ne u\}.$$

The proposed binary linear programming formulation is as follows:

$$\text{minimize} \sum_{(i,j) \in S_1} z_{ij},$$

subject to

$$\sum_{j \in S_4(u)} x_{ujtu} \le c, \quad \text{for } t \in S_5 \text{ and } u \in S_6, \tag{1}$$

$$\sum_{t=1}^{r-1} \sum_{i \in S_2(j,u,t)} c\, x_{ijtu} - \sum_{i \in S_3(j,u)} x_{jiru} \ge 0, \quad \text{for } j \in S_4(u),\ r \in S_5 \backslash \{1\}, \text{ and } u \in S_6, \tag{2}$$

$$\sum_{t \in S_5} \sum_{i \in S_2(j,u,t)} x_{ijtu} = 1, \quad \text{for } j \in S_4(u), \text{ and } u \in S_6, \tag{3}$$

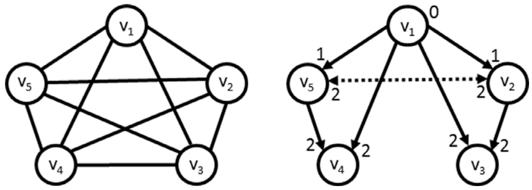
$$z_{ij} - \sum_{t \in S_5} x_{ijtu} \ge 0, \quad \text{for } j \in S_4(u),\ (i,j) \in S_1, \text{ and } u \in S_6, \tag{4}$$

$$z_{ij} - \sum_{t \in S_5} x_{jitu} \ge 0, \quad \text{for } i \in S_4(u),\ (i,j) \in S_1, \text{ and } u \in S_6, \tag{5}$$

$$z_{ij} \in \{0,1\}, \quad \text{for } (i,j) \in S_1 \tag{6}$$

$$x_{ijtu} \in \{0,1\}, \quad \text{for } i \in S_2(j,u,t),\ j \in S_4(u),\ t \in S_5, \text{ and } u \in S_6. \tag{7}$$

The objective is to minimize the cardinality of the edge set that defines the c-bg. Since only c messages may be sent by an originator u in a given time period, constraint set (1) is necessary to prevent each originator from sending more



(a) (2, 1)-mbg with five nodes. (b) Protocol for originator v_1.

Fig. 3. (2, 1)-mbg with five nodes and its protocol from originator v_1.

that c messages per time unit. Similarly, constraint set (2) guarantees that a non-originator will not transmit until it has received the message in a prior time unit. Constraint 3 guarantees each non-originator j receives exactly one message for each potential originator u. Furthermore, constraint sets (2) and (3) together insure that no more than c messages are sent from a non-originator during a particular time unit. Since exactly one message is received by a non-originator according to constraint set (3), constraint set (2) restricts the messages to be transmitted from a non-originator to be at most c messages during a single time unit. Constraint sets (4) and (5) guarantee that an edge exists if a message is sent between two nodes. The variables also have binary restrictions.

The binary linear programming formulation provides a framework to construct graphs that are similar to a c-mbg or (t, c)-rbg, but have different assumptions. For instance, the formulation could be modified so that the objective function could have a cost for each edge instead of minimizing the edge set cardinality. In a military example, a message may flow from an intelligence gathering node up the chain of command to a command and control node. A raw transmission from the intelligence gathering node to every node would lead to information overload [35], so a message may only be permitted to be transmitted to a subset of non-originators. The subset of non-originators to receive the message can be defined by only including the appropriate subset of constraints from constraint set (3). Similarly, if communication is not possible between two nodes, variables corresponding to this edge could be removed from the formulation.

4. Properties of k-fault tolerant c-broadcast graphs

Although researchers have considered k-ftbg and c-bg, these problems have not been generalized to k-fault tolerant c-broadcast graphs ((c, k)-bg). A (c, k)-bg is defined to be a k-ftbg that is permitted to transmit and receive a total of c messages during a given time unit. In such c-bg's, each non-originator receives the message even in the presence of k failures in communication lines. The function $B_{c,k}(n)$ denotes the minimum number of edges of a (c, k)-bg of order n. A (c, k)-bg with $B_{c,k}(n)$ edges is called a k-fault tolerant c-minimum broadcast graph ((c, k)-mbg). As mentioned in Section 2, there is not a known closed form expression for the minimum broadcast time for every instance, but several results provide the minimum broadcast time for instances satisfying certain properties. For general (c, k)-bg's, the minimum broadcast time required to construct a fault tolerant protocol is not necessarily greater than that for c-mbg's.

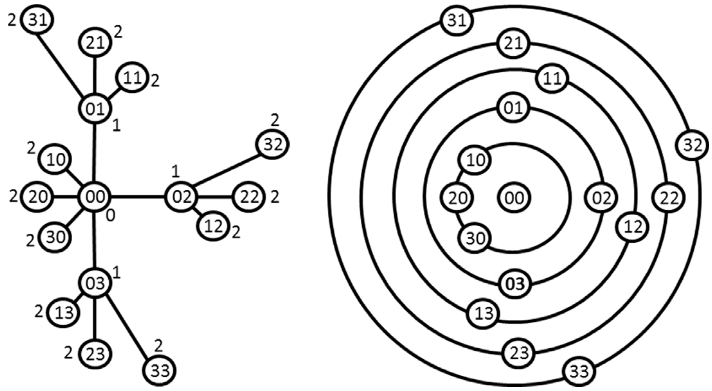
In Fig. 3(a), an example of a five node $(2, 1)$-bg with a minimum broadcast time of 2 is shown. This broadcast time is the same as for a five node c-mbg. Fig. 3(b) displays the protocol for this $(2, 1)$-bg. The numbers adjacent to each node and edge represent the time units that the node receives the message over that edge. Since each node is topologically equivalent, this protocol may be performed from each originator. Also, consider that the four non-originators must receive two messages each. Thus, a total of eight messages must be received. For eight messages to be received in a 2-broadcast protocol in 2 time units, the degree of the originator must be at least four. This implies that the graph in Fig. 3(a) is a $(2, 1)$-mbg.

The formulation in the next section may be used to determine the minimum broadcast time of an arbitrary (c, k)-mbg. Lemmas 1 and 2 present a closed form result for the minimum broadcast time for some specific (c, k)-mbg's. Theorem 1 provides an upper bound on $B_{c,k}(n)$ for $n = (c + 1)^m$ nodes where m is a positive integer. In the following results, the nodes indexed 0 through $(c + 1)^m - 1$ are denoted by a vector containing their base $c + 1$ representation with m digits. Many authors such as Ahlswede [1] have used this notation to derive results for $c = 1$. For notational purposes, we define $\alpha^{t,j} = (\alpha_1^{t,j}, \ldots, \alpha_m^{t,j})$, where $\alpha_i^{t,j} = 0$ if $i \neq m - t + 1$, and $\alpha_i^{t,j} = j$ if $i = m - t + 1$ for $i = 1, \ldots, m,\ t = 1, \ldots, m$, and $j = 1, \ldots, c$. Also, let $\beta^p = (\beta_1, \ldots, \beta_m)$, where $\beta_i = c + 1$ if $i \neq p$, and $\beta_i = c$ if $i = p$, for $i = 1, \ldots, m$. We define vector $a = (a_1, \ldots, a_m)$ modulo vector $b = (b_1, \ldots, b_m)$ to be evaluated as $(a_1 \bmod b_1, \ldots, a_m \bmod b_m)$.

Lemma 1. *For k even and $2 \leq k < c$, $m + 1$ is the minimum broadcast time for a (c, k)-bg with $(c + 1)^m$ nodes, for any positive integer m.*

Proof. To prove the broadcast time is bounded above by $m + 1$, a protocol must be constructed that is completed within this time. Without lost of generality, suppose the originator is node 0. At time t, $1 \leq t \leq m$, a node u with the message transmits the message to nodes $u \oplus \alpha^{t,j}$ for $j = 1, \ldots, c$, where $\oplus$ represents the vector sum modulo $(c + 1)$. Since this part of the protocol is identical to the protocol of a c-mbg with $(c + 1)^m$ nodes, only c messages are sent or received by a single node during a given time unit. Also, each non-originator has the message after the first m time units if failures do not occur.

At time $m + 1$, consider a non-originator u. Suppose the pth digit from the right is the rightmost non-zero digit of the string representing u. At time $m + 1$, node u receives the message from nodes $((u + \alpha^{p,j-1}) \bmod \beta^p) + \alpha^{p,1}$ and



(a) Protocol during first two time units.

(b) Protocol during time unit 3.

Fig. 4. Protocol for (c, k)-bg with $c = 3$, $k = 2$, and 16 nodes.

$((u+\alpha^{p,c-j-1}) \bmod \beta^p)+\alpha^{p,1}$ for $j = 1, \ldots, k/2$. These transmissions occur for each non-originator and are bi-directional. Since k is even and each node only transmits or receives from nodes that are identical except for the rightmost non-zero digit, each non-originator exchanges messages with at most $c - 1$ neighbors. Thus, the protocol is a c-broadcast protocol.

For the protocol to be k-ft, there must be $k + 1$ edge disjoint paths to each non-originator contained within the protocol. Suppose u is an arbitrary non-originator. A path exists from the originator to u during the first m time units. At time $m + 1$, u receives the message from nodes $((u + \alpha^{p,j-1}) \bmod \beta^p) + \alpha^{p,1}$ and $((u + \alpha^{p,c-j-1}) \bmod \beta^p) + \alpha^{p,1}$ for $j = 1, \ldots, k/2$. In the construction of the protocol, node u and nodes transmitting a message to node u at time $m + 1$ are identical except for their rightmost non-zero digit. Since these rightmost digits are different, these nodes have different predecessors that are direct children of the originator. The paths from the originator through each of these predecessors to u are disjoint, so the protocol contains $k + 1$ disjoint paths to u. Therefore, the minimum broadcast time has an upper bound of $m + 1$.

To prove by contradiction that this is the minimum broadcast time, suppose the minimum broadcast time is less than $m+1$. Since the graph is a (c, k)-bg, each node must receive $k+1$ messages. This implies that a total of $(k+1)((c+1)^m-1)$ messages must be received during time units $1 \le t \le m$. But this is a contradiction since at most $(c + 1)^m - 1$ messages are sent during this time in a c-broadcast protocol. Thus, the minimum broadcast time is $m + 1$. ■

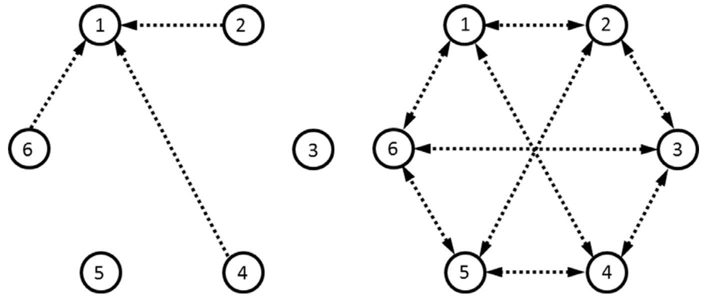
To illustrate this protocol, consider the example with $c = 3$, $k = 2$, and $m = 2$ shown in Fig. 4. During the first two time units, a c-broadcast protocol for $(c + 1)^m$ nodes is performed as shown for originator $(0, 0)$ in Fig. 4(a). For example at $t = 2$, node $(0, 2)$ transmits to node $(u \oplus \alpha^{2,1}) = (0, 2) + (1, 0) \bmod 4 = (1, 2)$. At time $t = 3$, each non-originator transmits to nodes that have indices identical to it except for the rightmost non-zero digit as Fig. 4(b) displays. For example, consider node $(0, 2)$. The rightmost non-zero digit is the second digit, so node $(0, 2)$ receives a message from $[(u+\alpha^{p,j-1}) \bmod \beta^p]+\alpha^{p,1} = (((0, 2)+(0, 0)) \bmod (4, 3))+(0, 1) = (0, 3)$. Also, node $(0, 2)$ receives the message from node $((u + \alpha^{p,c-j-1}) \bmod \beta^p) + \alpha^{p,1} = (((0, 2) + (0, 1)) \bmod (4, 3)) + (0, 1) = (0, 1)$. Each edge in Fig. 4(b) corresponds to a bi-directional message exchange during time unit 3.

Lemma 2. *For k odd and $1 \le k < c$, $m + 1$ is the minimum broadcast time for a (c, k)-bg with $(c + 1)^m$ nodes, for any positive integer m.*

Proof. For this proof, the result is separated into the cases of c even and c odd. First, suppose c is odd. Since k is also odd, we have $k + 1 < c$ to satisfy the criteria in Lemma 1 for a $(c, k + 1)$-bg. Thus, there is a c-bg with $k + 2$ edge disjoint paths that can be completed within $m + 1$ time units. Also, it is not possible to transmit $(k + 1)((c + 1)^m - 1)$ messages with a c-broadcast protocol in m time units. Thus, the minimum broadcast time is $m + 1$ if c is odd.

For the second case, suppose that c is even. At time unit t, $1 \le t \le m$, the protocol is identical to that discussed in Lemma 1. Thus, each node receives the message from the c-broadcast protocol during the first m time units. For time unit $m + 1$, let the pth digit in the string representing a non-originator u be the rightmost non-zero digit. At time $m + 1$, each non-originator u receives k messages that are sent by nodes $((u+\alpha^{p,c/2-1}) \bmod \beta^p)+\alpha^{p,1}$, $((u+\alpha^{p,j-1}) \bmod \beta^p)+\alpha^{p,1}$, and $((u + \alpha^{p,c-j-1}) \bmod \beta^p) + \alpha^{p,1}$ for $j = 1, \ldots, (k - 1)/2$. Similar to Lemma 1, these transmissions are bi-directional exchanges, a single non-originator does not transmit to more than c – 1 neighbors, and k paths additional disjoint paths are provided to each non-originator. Since there are k+1 disjoint paths to each non-originator within a c-broadcast protocol, there exists a (c, k)-bg with protocols that may be completed within $m+1$ time units. Also, since it is not possible to transmit $(k + 1)((c + 1)^m - 1)$ messages in m time units, this upper bound on minimum broadcast time is tight. Therefore, in either case, the minimum broadcast time is $m + 1$. ■

We now consider an example of the protocol for k odd and c even. In this example, $c = 6$, $k = 3$, and $m = 1$. The originator node 0 transmits the message to nodes 1, 2, 3, 4, 5, and 6 in time unit 1. Fig. 5 shows the transmissions in time



(a) Messages sent to node 1 at time 2. (b) All transmissions at time 2.

Fig. 5. Protocol during the last time unit for $(c,\ k)$-bg with $c = 6, k = 3$, and 7 nodes.

unit 2, but does not include the originator (node 0) since it does not transmit any messages. Fig. 5(a) displays the messages received by node 1 in time unit 2. All transmissions during time unit 2 are shown in Fig. 5(b). Since the protocol is symmetric with respect to each non-originator, no node is involved in more than three transmissions during the last time unit.

Theorem 1. *For* $1 \le k < c,\ B_{c,k}((c+1)^m) \le \frac{cm(c+1)^m}{2}$ *for any positive integer m.*

Proof. With a base $c+1$ representation of the nodes, consider a graph G that has $(c+1)^m$ nodes and an edge between two nodes if and only if these nodes differ by exactly one digit. Note that each transmission in the protocols of Lemmas 1 and 2 only occurs between nodes that do not differ by more than one digit. This implies that G is a (c, k)-bg. Since G has $\frac{cm(c+1)^m}{2}$ edges and is a (c, k)-bg, $B_{c,k}((c+1)^m)$ is bounded above by $\frac{cm(c+1)^m}{2}$. ■

5. A binary linear programming model for *k*-fault tolerant *c*-broadcasting

We now present a binary linear programming formulation that can be utilized to generate a sparse k-fault tolerant c-broadcast graph G of order n with broadcast time $b(G) \le T$. If the desired result is a $(1, 1)$-mbg, T is set to the minimum broadcast of $\lceil \log_2 n \rceil + 1$ that was derived by Liestman [38]. If the desired result is a (c, k)-mbg, a closed form solution for the minimum broadcast time does not exist. The formulation presented below can be used to determine the minimum broadcast time. T is initially set to $\lceil \log_{c+1} n \rceil$. If this formulation is infeasible, T is incremented by one until a feasible solution is obtained with T corresponding to the minimum broadcast time for this (c, k)-mbg. To compute an optimal $(t,\ c,\ k)$-rbg, T is chosen to be the minimum broadcast time plus t additional time units. The binary variables of the model are defined as follows:

$$Z_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ exists in the network,} \quad \text{for}(i,j) \in S_1, \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ijtup} = \begin{cases} 1 & \text{if node } i \text{ calls node } j \text{ at time } t \text{ for path } p \text{ from a given originator } u, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{for } i \in S_2(j,u,t),\ j \in S_4(u),\ t \in S_5,\ u \in S_6 \text{ and } p \in S_7,$$

where $S_7 = \{p : 1 \le p \le k+1\}$, $S_8(j) = \{i : 1 \le i \le j-1\}$, and $S_9(i) = \{j : i+1 \le j \le n\}$.

$$w_{ijtu} = \begin{cases} 1 & \text{if a message is sent across edge } (i,j) \text{ at time } t \text{ for a given originator } u, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{for } (i,j) \in S_1,\ t \in S_5 \backslash \{1\},\ \text{and } u \in S_6.$$

The proposed binary linear programming formulation is as follows:

$$\text{minimize} \sum_{(i,j)\in S_1} z_{ij},$$

subject to

$$\sum_{p\in S_7} \sum_{j\in S_4(u)} x_{ujtup} \le c, \quad \text{for } t \in S_5 \text{ and } u \in S_6, \tag{8}$$

$$\sum_{t=1}^{r-1} \sum_{i\in S_2(j,u,t)} c\, x_{ijtup} - \sum_{i\in S_3(j,u)} x_{jirup} \ge 0, \quad \text{for } j \in S_4(u),\ r \in S_5 \backslash \{1\},\ u \in S_6,\ \text{and } p \in S_7, \tag{9}$$

$$\sum_{t\in S_5} \sum_{i\in S_2(j,u,t)} x_{ijtup} = 1, \quad \text{for } j \in S_4(u),\ u \in S_6,\ \text{and } p \in S_7, \tag{10}$$

$$\sum_{p\in S_7}\sum_{t\in S_5} x_{ijtup} - z_{ij} \le 0, \quad \text{for } (i,j)\in S_1,\ j\in S_4(u),\ \text{and } u\in S_6, \tag{11}$$

$$\sum_{p\in S_7}\sum_{t\in S_5} x_{jitup} - z_{ij} \le 0, \quad \text{for } (i,j)\in S_1,\ i\in S_4(u),\ \text{and } u\in S_6, \tag{12}$$

$$\sum_{p\in S_7} x_{ijtup} - w_{ijtu} \le 0, \quad \text{for } (i,j)\in S_1,\ t\in S_5\backslash\{1\},\ u\in S_6,\ \text{and } p\in S_7, \tag{13}$$

$$\sum_{p\in S_7} x_{jitup} - w_{ijtu} \le 0, \quad \text{for } (i,j)\in S_1,\ t\in S_5\backslash\{1\},\ u\in S_6,\ \text{and } p\in S_7, \tag{14}$$

$$\sum_{i\in S_8(q)} w_{iqtu} + \sum_{j\in S_9(q)} w_{qjtu} \le c, \quad \text{for } q\in S_4(u),\ t\in S_5\backslash\{1\},\ \text{and } u\in S_6, \tag{15}$$

$$z_{ij}\in\{0,1\}, \quad \text{for } (i,j)\in S_1, \tag{16}$$

$$x_{ijtup}\in\{0,1\}, \quad \text{for } i\in S_2(j,u,t),\ j\in S_4(u),\ t\in S_5,\ u\in S_6,\ p\in S_7, \tag{17}$$

$$w_{ijtu}\in\{0,1\}, \quad \text{for } (i,j)\in S_1,\ t\in S_5\backslash\{1\},\ \text{and } u\in S_6. \tag{18}$$

In this formulation, the objective is to minimize the cardinality of the edge set of the (c,k)-bg. For each originator, the constraint sets guarantee that there are protocols for c-bg's that contain $k+1$ disjoint paths to each non-originator. To prevent the originator from transmitting more than c messages during a given time unit, constraint set (8) is necessary. Constraint set (9) prevents a message corresponding to a particular path from being sent by a node before the message is received. Constraint set (10) provides each non-originator with a message for each of the $k+1$ paths in a particular protocol for an originator. In constraint sets (11) and (12), the existence of an edge is established if a transmission occurs across that edge in some protocol. Constraints sets (8) through (12) are analogous to constraint sets (1) through (5) for the formulation for c-bg's. Constraint sets (11) and (12) also specify that the $k+1$ paths for a particular originator are disjoint. Although paths are disjoint in terms of the message being sent across the same edge in the same direction, bi-directional message exchanges may occur over an edge. That is, paths are disjoint with respect to directed edges, but are not necessarily disjoint with respect to undirected edges. The binary variables w_{ijtu} are introduced to allow these exchanges. The values w_{ijtu} are determined by constraint sets (13) and (14). These binary variables are one if a message is transmitted across an edge $(i,\ j)$ in either direction or both at time t for originator u, but zero if no transmission occurs. Constraint set (15) prevents a non-originator from sending, receiving, or participating in a bi-directional message exchange with more than c neighboring nodes during a given time unit for each protocol. The remaining constraint sets correspond to the binary restrictions.

Similar to the formulation for c-bg's, the formulation for (c,k)-bg's provides the flexibility to easily modify some of the assumptions of a (c,k)-bg. Costs for edges could be added to the objective function, the message can be restricted so that it is only sent to a subset of non-originators, and variables can be removed from the formulation if a transmission is impossible across an edge.

6. Computational results

In this section, computational results from the binary linear programming formulations of Sections 3 and 5 provide many heretofore undetermined optimal graphs. Results using the formulation in Section 3 provides eight previously unknown values of $B_c(n)$ and eleven improved upper bounds on other instances of $B_c(n)$. The formulation in Section 5 was used to obtain over 30 values of $B_{0,c,k,u}(n)$ as well as over 100 values for time relaxed and bounded degree (c,k)-bgs.

Since the number of variables in the formulations increases rapidly, even relatively small graphs may require extensive computational effort. In the computations, some modifications to the formulation are implemented to facilitate a more rapid convergence. These strategies included adding constraints to reduce the size of the search space, fixing certain variables to a value of one, and relaxing certain binary constraints. These results are based on execution of these models using GAMS Release 24.3.3 with CPLEX 12.6.0.1 as the optimization software package. These software packages were run on a PC with a MS Windows 64 bit operating system, Intel core i7 with 4 cores (8 processors) and 2.4 GHz, and 15.7 GB of memory.

To illustrate these strategies, consider the formulation corresponding to $B_{c=4}(12)$. Although the formulation in Section 3 does not converge to an optimal solution in 3600 s, the computational time may be drastically reduced by modifying the formulation. To reduce the search space, a constraint on the minimum degree of each node is added. Note that the minimum broadcast time for $B_{c=4}(12)$ is 2 time units. Thus, if the originator has degree one, only six nodes may be reached in a 4-broadcast protocol in 2 time units. If the originator has degree two, a 4-broadcast protocol may reach at most 11 nodes in 2 time units. Therefore, the minimum degree on each node is $d_{\min}=3$ and the following constraint set (19) is added to reduce the search space:

$$\sum_{i\in S_8(q)} z_{iq} + \sum_{j\in S_9(q)} z_{qj} \ge d_{\min}, \quad \text{for } q\in S_6. \tag{19}$$

Another strategy involves assigning some variables to a value of one. That is, specify some edges that must exist in a minimum c-broadcast graph. First, an upper bound is used to determine the maximum degree of some node in the network.

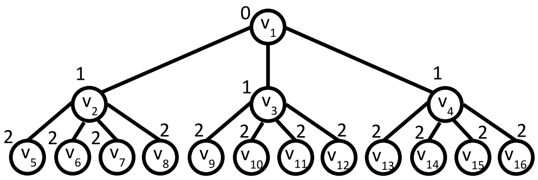


Fig. 6. Maximum number of nodes in a 4-broadcast protocol for an originator of degree 3.

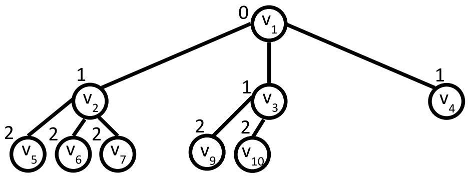


Fig. 7. Edges that must be in a minimum 4-broadcast graph with 12 nodes.

When proving optimality, the upper bound can be assumed to be one less than sparsest known graph since this is as an upper bound on an improved solution. This upper bound may be determined from the best current feasible solution from running the formulation or from another construction method such as a compounding algorithm. To illustrate this, suppose that an upper bound, U, of 21 on $B_{c=4}(12)$ is given. This implies that there is at least one node with degree $\left\lfloor \frac{2U}{n} \right\rfloor = \left\lfloor \frac{2*21}{12} \right\rfloor = 3$ or less. Since the minimum degree is 3, at least one node exists with degree 3. Without loss of generality, let v_1 be an originator with degree 3. In Fig. 6, the 4-broadcast protocol is displayed with the maximum number of nodes that may be reached in two time units for an originator with degree 3. To reach twelve nodes within 2 time units in a 4-broadcast protocol, v_1 must transmit the message to v_2, v_3, and v_4 at time 1. The edges $(v_1,\ v_2)$, $(v_1,\ v_3)$, and $(v_1,\ v_4)$ must be part of the minimum c-broadcast graph.

If v_2, v_3, and v_4 only transmit the message to two nodes each, then the 4-broadcast protocol contains at most 10 nodes. Therefore, at least one of these nodes transmits the message to at least three nodes. Suppose without loss of generality that node v_2 transmits the message to the three children v_5, v_5, and v_6 in time unit 2 in the 4-broadcast protocol for v_1. This implies that the edges $(v_2,\ v_5)$, $(v_2,\ v_6)$, and $(v_2,\ v_7)$ must be part of the minimum c-broadcast graph. Even if v_2 transmits to four nodes in time unit 2, v_3 or v_4 must transmit the message to at least two nodes for the protocol to contain 12 nodes. Without loss of generality, suppose v_3 sends two messages and the edges $(v_3,\ v_9)$, and $(v_3,\ v_{10})$ must be part of a minimum c-broadcast graph. The eight edges shown in Fig. 7 may be assigned to be in the graph without any loss of optimality. The CPU time is reduced to 13.1 s by assigning the eight z_{ij} variables corresponding to these edges to a value of one and using the minimum degree constraints. All of these implementation strategies do not change the optimality of the graph resulting from the formulation.

In Table 1, computational results for $B_c(n)$ are displayed for $c = 2$, 3, and 4. In each case, a column provides the best previously-identified solution as an upper bound on $B_c(n)$ with boldface entries in the $B_c(n)$ column corresponding to optimal graphs. The number appearing in brackets in the "Prior $B_c(n)$" provides the reference to each best previously-identified solution. The objective function values from computational results for our new formulation are shown in the next column with boldface numbers denoting optimal graphs. Our formulation has identified eight new optimal values of $B_c(n)$. In some cases, the optimization software did not converge to an optimal solution within 3600 s, so the incumbent solution is reported as an upper bound on $B_c(n)$. Entries appear in italicized font indicate a non-optimal, but new best-known solution; our formulation has identified an additional 11 such graphs. If an implementation strategy was used, the column $d_{\min}$, n_f, *UB* is included to describe the strategy. The first component represents the minimum degree for constraint set (19), the second component n_f is the number of edges that are fixed, and the upper bound that is assumed to determine the number of fixed edges is *UB*. For every case in Table 1, the binary constraints on x_{ijtu} can be relaxed during the final time unit with no loss of optimality. The lower bounds on $B_c(n)$ are reported for cases where the optimization run of the formulation was terminated before converging to an optimal solution. The CPU time is also reported in seconds for each instance.

For the remaining computational results, the broadcast function is defined as $B_{t,c,k,d}(n)$ where t is the additional number of time units allowed over the minimum time, c is the number of messages that may be sent or received during a time unit, k is the number of possible faults, and d is the bound on the degree. The case where the degree is unbounded is denoted by $d = u$. Similarly, $T_{t,c,k,d}(n)$ denotes the minimum broadcast time for a graph corresponding to $B_{t,c,k,d}(n)$. The values of $T_{t,c,k,d}(n)$ in Tables 2 and 4 were determined by iteratively solving the formulations for broadcast times starting with $\lceil \log_{c+1} n \rceil$ as described in Section 5. The CPU time is reported in seconds.

Table 2 presents optimal results for (c, k)-mbgs for $k = 1, 2$ and $c = 2, 3, 4$ for values of n between 3 and 8. Optimal results for their time relaxed versions are shown in Table 3.

Table 1
$B_c(n)$ values for $8 \le n \le 15$ and $2 \le c \le 4$.

n	Prior $B_2(n)$	$B_2(n)$	$d_{\min}$, n_f, UB	Lower bound	CPU time	Prior $B_3(n)$	$B_3(n)$	$d_{\min}$, n_f, UB	Lower bound	CPU time	Prior $B_4(n)$	$B_4(n)$	$d_{\min}$, n_f, UB	Lower bound	CPU time
8	**12** [34]	**12**	–	–	0.1	**11** [34]	**11**	–	–	0.5	**11** [34]	**11**	–	–	1.1
9	**18** [21]	**18**	–	–	0.1	**13** [33]	**13**	–	–	1.6	**13** [33]	**13**	–	–	2.4
10	**12** [34]	**12**	–	–	196.5	**15** [36]	**15**	–	–	4.5	**15** [33]	**15**	–	–	21.4
11	**13** [34]	**13**	2, 4, 13	–	19.2	**18** [25]	**18**	–	–	144.9	**17** [33]	**17**	–	–	147.2
12	**15** [34]	**15**	2, 6, 14	–	1012.1	30 [36]	**22**	3, 11, 22	–	8.0	30 [36]	**21**	3, 8, 21	–	13.1
13	19 [36]	*17*	2, 6, 17	15	3600.0	33 [36]	**26**	3, 11, 26	–	3429.2	33 [36]	**24**	3, 10, 24	–	174.1
14	21 [36]	*20*	2, 5, 21	16	3600.0	37 [36]	**31**	4, 13, 31	–	16,139.5	37 [36]	**27**	3, 11, 27	–	1604.3
15	24 [36]	*22*	2, 6, 24	17	3600.0	42 [36]	**38**	–	–	356.0	42 [36]	**30**	3, 9, 29	–	3813.7
16	28 [36]	28	2, 6, 28	19	3600.0	**48** [21]	**48**	–	–	187.1	48 [36]	*33*	–	27	3600.0
17	29 [36]	30	2, 7, 30	20	3600.0	29 [36]	*23*	1, 4, 23	14	3600.0	52 [36]	*38*	4, 12, 38	35	3600.0
18	30 [36]	34	–	20	3600.0	30 [36]	30	–	15	3600.0	57 [36]	*43*	4, 15, 43	38	3600.0
19	40 [36]	*38*	2, 7, 38	24	3600.0	34 [36]	*32*	2, 4, 32	21	3600.0	63 [36]	*50*	4, 15, 50	40	3600.0
20	42 [36]	42	3, 8, 42	31	3600.0	35 [36]	35	2, 5, 35	22	3600.0	70 [36]	*57*	4, 17, 57	43	3600.0

Table 2
$B_{0,c,k,u}(n)$ values for $3 \le n \le 8$, $2 \le c \le 4$, and $k = 1, 2$.

n	k	$B_{0,2,k,u}(n)$	$T_{0,2,k,u}(n)$	CPU time	$B_{0,3,k,u}(n)$	$T_{0,3,k,u}(n)$	CPU time	$B_{0,4,k,u}(n)$	$T_{0,4,k,u}(n)$	CPU time
3	1	**3**	2	<0.1	**3**	2	<0.1	**3**	2	<0.1
4	1	**5**	2	<0.1	**5**	2	<0.1	**5**	2	<0.1
5	1	**10**	2	<0.1	**7**	2	<0.1	**7**	2	0.1
6	1	**8**	3	1.4	**12**	2	1.5	**9**	2	0.3
7	1	**10**	3	38.0	**18**	2	2633.4	**14**	2	6.8
8	1	**13**	3	3909.4	24[b]	2	36,000.0	**16**	2	20.3
4	2	**6**	3	<0.1	**6**	2	<0.1	**6**	2	<0.1
5	2	**10**	3	9.4	**9**	2	<0.1	**9**	2	<0.1
6	2	**12**	3	212.9	**9**	3	0.5	**12**	2	0.4
7	2	**14**	3	662.0	**12**	3	71.8	**21**	2	5353.1
8	2	21[a]	3	36,000.0	**15**	3	32,212.2	**14**	3	1973.6

[a] The lower bound on $B_{0,2,2,u}(8)$ from the terminated optimization run is 16.
[b] The lower bound on $B_{0,3,1,u}(8)$ from the terminated optimization run is 22.

Table 3
Time relaxed $B_{t,c,k,u}(n)$ values for $3 \le n \le 8$, $1 \le t \le 3$, $2 \le c \le 4$, and $k = 1$.

n	c	$B_{1,c,1,u}(n)$	$T_{1,c,1,u}(n)$	CPU time	$B_{2,c,1,u}(n)$	$T_{2,c,1,u}(n)$	CPU time	$B_{3,c,1,u}(n)$	$T_{3,c,1,u}(n)$	CPU time
3	2	**3**	3	<0.1	**3**	4	<0.1	**3**	5	0.0
4	2	**4**	3	0.1	**4**	4	<0.1	**4**	5	0.0
5	2	**7**	3	0.5	**5**	4	0.2	**5**	5	0.2
6	2	**7**	4	2.0	**6**	5	3.8	**6**	6	1.4
7	2	**8**	4	28.7	**8**	5	28.3	**7**	6	29.3
8	2	**10**	4	518.4	**10**	5	3526.2	**9**	6	338.4
3	3	**3**	3	<0.1	**3**	4	<0.1	**3**	5	0.0
4	3	**4**	3	<0.1	**4**	4	<0.1	**4**	5	0.0
5	3	**6**	3	0.2	**5**	4	0.1	**5**	5	0.3
6	3	**8**	3	1.4	**7**	4	0.9	**6**	5	8.7
7	3	**10**	3	25.8	**8**	4	26.8	**8**	5	26.7
8	3	**12**	3	3532.5	**10**	4	612.9	**9**	5	499.7
3	4	**3**	3	<0.1	**3**	4	<0.1	**3**	5	0.0
4	4	**4**	3	<0.1	**4**	4	<0.1	**4**	5	0.0
5	4	**6**	3	0.1	**5**	4	<0.1	**5**	5	0.2
6	4	**8**	3	2.5	**7**	4	0.7	**6**	5	0.9
7	4	**10**	3	30.3	**8**	4	23.8	**8**	5	24.6
8	4	**11**	3	229.6	**10**	4	187.9	**9**	5	123.6

Table 4
$B_{0,c,k,d}(n)$ values for some combinations of $3 \le n \le 8$, $2 \le c \le 4$, $k = 1, 2$, and $d = 3, 4$.

n	k	d	$B_{0,2,k,d}(n)$	$T_{0,2,k,d}(n)$	CPU time	$B_{0,3,k,d}(n)$	$T_{0,3,k,d}(n)$	CPU time	$B_{0,4,k,d}(n)$	$T_{0,4,k,d}(n)$	CPU time
3	1	3	**3**	2	<0.1	**3**	2	<0.1	**3**	2	<0.1
4	1	3	**5**	2	<0.1	**5**	2	<0.1	**5**	2	<0.1
5	1	3	**7**	3	0.4	**6**	3	0.1	**6**	3	0.1
6	1	3	**8**	3	1.7	**8**	3	1.1	**8**	3	1.2
7	1	3	**8**	4	46.2	**8**	4	27.0	**8**	4	18.4
8	1	3	**10**	4	68.6	**12**	3	35.0	**12**	3	30.0
3	1	4	**3**	2	<0.1	**3**	2	<0.1	**3**	2	<0.1
4	1	4	**5**	2	<0.1	**5**	2	<0.1	**5**	2	<0.1
5	1	4	**10**	2	<0.1	**7**	2	<0.1	**7**	2	<0.1
6	1	4	**8**	3	1.6	**12**	2	0.7	**12**	2	0.6
7	1	4	**10**	3	15.5	**10**	3	21.2	**14**	2	1.9
8	1	4	**13**	3	2384.4	**12**	3	8308.7	**16**	2	11.7
4	2	4	**6**	3	<0.1	**6**	2	<0.1	**6**	2	<0.1
5	2	4	**10**	3	8.4	**9**	2	<0.1	**9**	2	<0.1
6	2	4	**12**	3	159.9	**9**	3	0.2	**9**	3	0.6
7	2	4	**14**	3	92.5	**12**	3	166.0	**12**	3	48.5

The formulations are easily modified for bounded degree graphs by adding the constraint set

$$\sum_{i \in S_8(q)} z_{iq} + \sum_{j \in S_9(q)} z_{qj} \le d_{\max}, \quad \text{for } q \in S_6, \tag{20}$$

where $d_{\max}$ is the maximum degree. These constraints also reduce the search space. Table 4 provides new optimal values for bounded degree (c, k)-mbg's.

7. Future work

The main component of future exploration on integer programming formulations for broadcasting and gossiping would be to implement these formulations more efficiently and obtain new mbg and gossip graphs. Alterations of the formulations could likely be developed for other problems in these categories. For example, c-gossiping could be considered. Such extension could allow for a new formulation for the linear cost gossiping problem as well.

Appendix. Supplementary data

Output files (in GAMS Data Exchange *.*GDX* format) detailing all solutions presented in Tables 1–4 have been posted to the following publicly-accessible URL: https://hdl.handle.net/10355/46234.

References

[1] R. Ahlswede, L. Gargano, H.S. Haroutunian, L.H. Khachatrian, Fault-tolerant minimum broadcast networks, Networks 27 (1996) 293–307.
[2] A. Averbuch, I. Peeri, Y. Rodity, Improved upper bound on m-time-relaxed k-broadcasting communication networks. Tel Aviv University. http://www.cs.tau.ac.il/~amir1/PS/IRIT-P.pdf (accessed 28.02.15).
[3] A. Averbuch, Y. Rodity, B. Shoham, Construction of minimum time-relaxed broadcasting communication networks, J. Combin. Math. Combin. Comput. 43 (2002) 123–134.
[4] A. Averbuch, R. Shabtai, Y. Roditty, Efficient construction of broadcast graphs, Discrete Appl. Math. 171 (2014) 9–14.
[5] G. Barsky, H. Grigoryan, H. Harutyunyan, Tight lower bounds on broadcast function for $n = 24$ and 25, Discrete Appl. Math. 175 (2014) 109–114.
[6] J.C. Bermond, C. Delorme, J.J. Quisquater, Strategies for interconnection networks: Some methods from graph theory, J. Parallel Distrib. Comput. 3 (1986) 433–449.
[7] J.C. Bermond, P. Fraigniaud, J.G. Peters, Antepenultimate broadcasting, Networks 26 (1995) 125–137.
[8] J.C. Bermond, P. Hell, A. Liestman, J.G. Peters, Sparse broadcast graphs, Discrete Appl. Math. 36 (1992) 97–130.
[9] A. Dekker, Applying social network concepts to military C4ISR Architecture, Connections 24 (3) (2002) 93–103.
[10] A. Dekker, Centralisation and decentralisation in network centric warfare, J. Battlefield Technol. 6 (2) (2003) 23–28.
[11] M.J. Dinneen, J.A. Ventura, M.C. Wilson, G. Zakeri, Compound constructions of minimal broadcast networks, Discrete Appl. Math. 93 (1999) 205–232.
[12] M.J. Dinneen, J.A. Ventura, M.C. Wilson, G. Zakeri, Construction of time-relaxed minimal broadcast networks, Paral. Proc. Lett. 9 (1) (1999) 53–68.
[13] A. Farley, Minimum broadcast networks, Networks 9 (1979) 313–332.
[14] A.M. Farley, S.T. Hedetniemi, S. Mitchell, A. Proskurowski, Minimal broadcast graphs, Discrete Math. 25 (1979) 189–193.
[15] A.M. Farley, A. Proskurowski, Between whispering and shouting, Congr. Numer. 72 (1990) 209–212.
[16] A.M. Farley, A. Proskurowski, Bounded-call broadcasting, Discrete Appl. Math. 53 (1994) 37–53.
[17] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[18] L. Gargano, Tighter bounds on fault-tolerant broadcasting and gossiping, Networks 22 (1992) 469–486.
[19] L. Gargano, U. Vaccaro, On the construction of minimal broadcast networks, Networks 19 (1989) 637–689.
[20] L. Gargano, U. Vaccaro, Minimum time broadcast networks tolerating a logarithmic number of faults, SIAM J. Discrete Math. 5 (1992) 178–198.
[21] M. Grigni, D. Peleg, Tight bounds on minimum broadcast networks, SIAM J. Discrete Math. 4 (2) (1991) 207–222.
[22] H. Grigoryan, H. Harutyunyan, New lower bounds on broadcast function, Lecture Notes in Comput. Sci. 8546 (2014) 174–184.
[23] H. Harutyunyan, An efficient vertex addition method for broadcast networks, Internet Math. 5 (3) (2009) 211–225.
[24] H. Harutyunyan, A. Liestman, More broadcast graphs, Discrete Appl. Math. 98 (1–2) (1999) 81–102.
[25] H. Harutyunyan, A. Liestman, Improved upper and lower bounds for k-broadcasting, Networks 37 (2) (2001) 94–101.
[26] H. Harutyunyan, A. Liestman, k-broadcasting in trees, Networks 38 (3) (2001) 163–168.
[27] H. Harutyunyan, A. Liestman, On the monotonicity of the broadcast function, Discrete Math. 262 (2003) 149–157.
[28] H. Harutyunyan, A. Liestman, Upper bounds on the broadcast function using minimum dominating sets, Discrete Math. 312 (20) (2012) 2992–2996.
[29] H. Harutyunyan, A. Liestman, J. Peters, D. Richards, Broadcasting and gossiping, in: J. Gross, J. Yellen, P. Zhang (Eds.), The Handbook of Graph Theory, Chapman and Hall/CRC, 2013, pp. 1477–1494.
[30] S.M. Hedetniemi, S.T. Hedetniemi, A. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (1988) 319–349.
[31] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, W. Unger, Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance, in: Texts Theoretical Computer Science. EATCS Series, Springer, Berlin, 2005.
[32] L.H. Khachatrian, O.S. Haroutunian, Construction of new classes of minimal broadcast networks, in: Conference on Coding Theory, Armenia, 1990.
[33] J.C. Konig, E. Lazard, Minimum k-broadcast graphs, Discrete Appl. Math. 53 (1994) 199–209.
[34] E. Lazard, Broadcasting in DMA-bound bounded degree graphs, Discrete Appl. Math. 37–38 (1992) 387–400.
[35] T. Lee, S. Ghosh, A novel approach to asynchronous, decentralized, decision-making in military command and control, in: Proceedings of the ISADS 95 Second International Symposium on Autonomous Decentralised Systems, 1995, pp. 25–31.
[36] S. Lee, J.A. Ventura, An algorithm for constructing minimal c-broadcast networks, Networks 38 (2001) 6–21.
[37] S. Lee, J.A. Ventura, Construction of time-relaxed c-broadcast networks, Telecommun. Syst. 24 (1) (2003) 47–59.
[38] A. Liestman, Fault-tolerant broadcast graphs, Networks 15 (1985) 159–171.
[39] A. Liestman, J. Peters, Broadcast networks of bounded degree, SIAM J. Discrete Math. 1 (1988) 531–540.
[40] A. Pelc, Fault-tolerant broadcasting and gossiping in communication networks, Networks 28 (1996) 143–156.
[41] D. Peleg, A. Schäffer, Time bounds on fault-tolerant broadcasting, Networks 19 (1989) 803–822.
[42] D. Richards, A. Liestman, Generalizations of broadcasting and gossiping, Networks 18 (1988) 125–138.
[43] B. Shao, On k-broadcasting in graphs (Ph.D. thesis), Concordia University, Montreal, PQ, Canada, 2006.
[44] A. Shastri, Time-relaxed broadcasting in communication networks, Discrete Appl. Math. 83 (1998) 263–278.
[45] P.J. Slater, E.J. Cockayne, S.T. Hedetniemi, Information dissemination in trees, SIAM J. Comput. 10 (1981) 692–701.
[46] M.X. Weng, J.A. Ventura, A doubling procedure for constructing minimal broadcast networks, Telecommun. Syst. 3 (1995) 259–293.
[47] T. Yao, P. Che Bor Lam, W. Lin, G. Zhou, On time-relaxed broadcasting networks, Discrete Appl. Math. 158 (2010) 1029–1034.
[48] T. Yao, G. Zhou, J. Zhou, A 1-relaxed minimum broadcast graph on 15 vertices, Appl. Math. Lett. 17 (2004) 249–252.