Noname manuscript No.
(will be inserted by the editor)

Insert your title here

Do you have a subtitle? If so, write it here

First Author · Second Author

Received: date / Accepted: date

Abstract **Keywords**

1 Introduction

The minimum broadcast time consists of a set of communication nodes with a subset of source nodes. The task is to disseminate a signal to every node in a shortest possible time while abiding by communication rules. An *informed* node is a node that has received the signal. Otherwise, a node is *uninformed*. At the beginning, the set of informed nodes is exactly the set of sources. An informed node u can send the signal to an uninformed node v if u and v are located within a communication vicinity of each other.

The continuous time is divided into discrete time steps. At each time step, every informed node can forward the signal to at most one uninformed neighbor. The number of informed nodes at some time step can therefore be up to double the number of informed nodes at the previous time step.

2 Network Model and Notation

The communication network is represented by a connected undirected unweighted graph $G = (V, E)$ and a subset of nodes $S \subseteq V$, where $|V| = n$ and $|S| = s$. Broadcasting is defined as a sequence of sets $S = V_0 \subseteq \cdots \subseteq V_k = V$ where each V_i represents the nodes informed after time step i, $0 \leq i \leq k$.

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

For each node $v \in V_i \setminus V_{i-1}$, there exists a single node $p(v) \in V_{i-1}$ adjacent to v, which forwarded the signal to v. Also, for every two distinct nodes $u, v \in V_i \setminus V_{i-1}$ we have $p(u) \neq p(v)$. The value k is referred to as *delay*. The optimization problem in question is defined as follows [3,4]:

Problem 1 Given $G = (V, E)$ and $S \subseteq V$, find a sequence $S = V_0 \subseteq \cdots \subseteq V_k = V$ of minimum length k and a mapping $p : V \setminus S \to V$, such that for each $v \in V \setminus S : \{v, p(v)\} \in E$, and for each $u, v \in V_i \setminus V_{i-1} : p(u) \neq p(v) \Leftrightarrow u \neq v$.

We also define the set $A = \{(i, j), (j, i) : \{i, j\} \in E\}$ that consists of all arcs that can be derived by directing edges in E. A degree, in-degree and out-degree of node v in G are denoted by $\deg_G(v)$, $\deg^-_G(v)$ and $\deg^+_G(v)$, respectively. Whenever there is no danger of confusion, the subscript G is omitted. The set of neighbors of $v \in V$ in G is denoted by $N(v)$.

For convenience, we consider the following definition of broadcast trees [2]: For $s \in S$, a broadcast tree T_s with node set V_{T_s} is a time-labeled directed subgraph of G describing a broadcast originated by s by the following rules:

1. T_s is rooted at s with arcs directed towards the leaves.
2. Each node v is labeled with an integer $t(v)$, where $t(s) = 0$.
3. Whenever v is a parent of u in T_s $t(v) < t(u)$.
4. Whenever v and u are siblings in T_s, $t(v) \neq t(u)$.

A set of trees $T = \{T_s : \cup_{s \in S} V_{T_s} = V\}$ forms a partition of G into trees referred to as *broadcast forest* (or *broadcast protocol*?). T can be derived from a given sequence of sets of nodes defining broadcasting and the mapping p. Given a broadcast forest T, the delay is determined as $k = \max_{v \in V}\{t(v)\}$. Further, let T^i_s be a subtree of T_s obtained by pruning all nodes $v \in V_{T_s}$ with $t(v) > i$. Analogously, we define $T^i = \{T^i_s : s \in S\}$.

3 Exact methods

Problem 1 can be formulated as an integer linear program and solved to optimality. In this section, we present two different modeling approaches.

3.1 Broadcast time model

The first studied model is a straightforward formulation of the problem. Consider variables

$$x^t_{ij} = \begin{cases} 1, & \text{if } j \in V_t \text{ and } p(j) = i, \\ 0, & \text{otherwise,} \end{cases}$$

and a variable c representing the number of necessary time steps. The worst case scenario is when G is a path $v_1, \ldots, v_n$ with $S = \{v_1\}$. In such an instance, the necessary number of time steps is $n - 1$, which gives a trivial upper bound on c. Problem 1 is then formulated as follows:

$$\min c \tag{1a}$$
$$\sum_{i \in N(s)} x_{si}^1 \leq 1 \qquad s \in S, \tag{1b}$$
$$\sum_{t=1}^{n-1} \sum_{j \in N(i)} x_{ji}^t = 1 \qquad i \in V \setminus S, \tag{1c}$$
$$\sum_{j \in N(i)} x_{ij}^t \leq 1 \qquad t = 1, \ldots, n-1, i \in V, \tag{1d}$$
$$\sum_{j \in N(i)} x_{ij}^t \leq \sum_{u=1}^{t-1} \sum_{k \in N(i) \setminus \{j\}} x_{ki}^u \qquad i \in V \setminus S, t = 2, \ldots, n-1, \tag{1e}$$
$$\sum_{t=1}^{n-1} t \cdot x_{ij}^t \leq c \qquad (i,j) \in A, \tag{1f}$$
$$x_{ij}^1 = 0 \qquad (i,j) \in A, i \notin S, \tag{1g}$$
$$x \in \{0,1\}^{A \times V}, c \in \{1, \ldots, n-1\}. \tag{1h}$$

Constraints (1b) indicate that for each source node s, there is at most one adjacent node $i \in V_1$ such that $p(i) = s$. By (1c), for every non-source node, there is exactly one node j such that $p(i) = j$. Constraints (1d) enforce that for each node $i \in V$ and each subset V_t, there is at most one adjacent node $j \in V_t$ with $p(j) = i$. The requirement that a non-source node has $j \in V_t : p(j) = i$ only if there exists $k \in V_{t-1} : p(i) = k$ is modeled by (1e). The length of the sequence of subsets is captured by (1f), and finally, (1g) state that if $p(j) \notin S$ for some $j \in V$, then $j \notin V_1$.

3.2 Binomial tree model

The following method solves Problem 1 by solving a sequence of decision problems:

Problem 2 Given $G = (V, E)$, $S \subseteq V$ and $k \in \mathbb{N}$, is there a sequence $S = V_0 \subseteq \cdots \subseteq V_k = V$ and a mapping $p : V \setminus S \to V$, such that for each $v \in V \setminus S : \{v, p(v)\} \in E$ and for each $u, v \in V_i \setminus V_{i-1} : p(u) \neq p(v) \Leftrightarrow u \neq v$?

For a delay k, at most $s \cdot 2^k$ nodes can be informed within k steps. This can be achieved when the broadcast forest T consists of binomial trees B^k of order k rooted at sources $s \in S$. Hence, if there is a partition of G into s pruned binomial trees of order at most k rooted at sources, then (G, S, k) is a YES instance of Problem 2.

Let $I = \{1, \ldots, 2^k\}$. For a directed rooted binomial tree $B^k = (V^k, A^k)$, we define a systematic numbering of nodes in V^k, so that a node number determines a unique position in B^k. I.e., we need a bijective mapping $\beta : V^k \to I$. A suitable mapping β assigns values increasingly with decreasing outgoing degree. If there is an ambiguity, a node whose parent has a lower number is assigned a lower number. This mapping is defined recursively as

$$\beta(v) = \begin{cases} 1, \text{ if } v \in S \text{ is a root of } B^k, \\ \beta(p(v)) + 2^{k-deg^+(v)-1}, \text{ otherwise.} \end{cases} \tag{2}$$

Observation 1 *For each $i \in \{1,\dots,k\}$, the set $\{v \in V^k : 1 \le \beta(v) \le 2^i\} \setminus \{v \in V^k : 1 \le \beta(v) \le 2^{i-1}\}$ contains nodes with out-degree $k-i$.*

Observation 2 *Children of $v \in V^k$ with $deg^+(v) = \ell$ have out-degree $0,\dots,\ell-1$.*

Proposition 1 *An instance (G,S,k) of Problem 2 is a YES-instance iff there exists a partition of G into s directed pruned binomial trees $B_1^k,\dots,B_s^k$ of order k rooted at sources in S, such that for each $B_i^k = (V_i^k, A_i^k)$ we have that $A_i^k \cap \{(u,v) : u \in V_\alpha \wedge v \in V_i^k\} = \emptyset$.*

Proof Assume there is a node partition of G into directed rooted binomial trees $B_1^k, \dots B_s^k$. Nodes in V are divided into a sequence of subsets $S = V_0 \subseteq \dots \subseteq V_k = V$ such that $V_i = \{v \in V : 1 \le \beta(v) \le 2^i\}$, and $p(v)$ is a parent of v in a corresponding binomial tree. For $u,v \in V_i \setminus V_{i-1}$ we have from Observation 1 that $deg^+(u) = deg^+(v) = k-i$. Furthermore, $p(u) \neq p(v)$ must hold, because due to 2 no node has two children with the same out-degree. These observations can be applied because the mapping p corresponds to arcs in binomial trees $B_1^k,\dots,B_s^k$.

Conversely, suppose there is a sequence of subsets and a mapping p in G with the desired properties. For $1 \le i \le k$, $|V_i| \ge 2|V_{i-1}|$ must hold, because otherwise $\forall u,v \in V_i : p(u) \neq p(v) \Leftrightarrow u \neq v$ could not be satisfied. For some node $v \in V_i$, there is at most $k-i$ nodes u such that $p(u) = v$. Pruned binomial trees covering G can then be constructed simply by following the mapping p. □

Consider a graph $G' = (V', E')$ constructed by adding a universal node v_0 to G. The set of nodes and edges is then $V' = V \cup \{v_0\}$ and $E' = E \cup \{\{v_0, v\} : v \in V\}$. The ILP model based on partition into binomial trees uses only one type of variables

$$y_{is}^v = \begin{cases} 1, \text{ if node } v \text{ is the } i\text{-th node of the binomial tree rooted at } s \in S, \\ 0, \text{ otherwise,} \end{cases}$$

where $v \in V'$ and $i \in I$. With the definition of G' above, it is straightforward to specify constraints that enforce desired values for y-variables. Whenever $y_{is}^{v_0} = 1$, it indicates that the binomial tree B_s^k was pruned in node i.

Let us define the set $C(i)$ of β-values of children of node v with $\beta(v) = i$ in B^k:

$$C(i) = \{2^j + i : j = \lceil \log_2 i \rceil, \dots, k-1\}. \tag{3}$$

The values of the variables must fulfill constraints

$$
\begin{aligned}
&\sum_{i\in I}\sum_{s\in S} y^v_{is} = 1 && v \in V, && \text{(4a)}\\
&\sum_{v\in V'} y^v_{is} = 1 && i \in I, s \in S, && \text{(4b)}\\
&y^s_{1s} = 1 && s \in S, && \text{(4c)}\\
&y^u_{is} + y^v_{\ell s} \le 1 && i \in I, \ell \in C(i), s \in S, u, v \in V, \{u, v\} \notin E, && \text{(4d)}\\
&y^{v_0}_{is} + y^v_{\ell s} \le 1 && i \in I, \ell \in C(i), s \in S, v \in V && \text{(4e)}\\
& && y \in \{0,1\}^{I\times S\times V'}. && \text{(4f)}
\end{aligned}
$$

As the model solves the decision problem, it suffices to find any feasible solution, and no objective function is needed. The interpretation of constraints (4a) is that every node in the original graph G belongs to exactly one binomial tree. Note that these constraints are quantified only over V and not over V'. In this way it is achieved that v_0 can be regarded as a part of several binomial trees. By (4b) is ensured that there is always exactly one i-th node of each binomial tree. By the summation over V' is ensured, that pruned nodes are collectively represented by v_0. Next, (4c) enforce that source nodes are always the first nodes in corresponding binomial trees, in accordance with definition (2) of the mapping β. The remaining two sets of constraints guarantee that the arcs of binomial trees follow edges in E'. In particular, it is enforced by (4d) that if u and v are not adjacent in G, then v must not act as a child of u in any binomial tree. Finally, (4e) forbids any node from V to be a child of v_0 in any binomial tree. This reflects the obvious fact that if a tree is pruned at some node, all its descendants must also be excluded from the tree. Without (4d) and (4e), it could be possible to find a feasible solution, even when no partition of G into pruned binomial trees exists.

Constraints (4d) and (4e) can be strengthened by

$$
\begin{aligned}
&y^{v_0}_{is} + y^u_{is} + \sum_{v\in V\setminus N(u)} y^v_{\ell s} \le 1 && u \in V, && \text{(5a)}\\
&y^{v_0}_{is} + y^u_{\ell s} + \sum_{v\in V\setminus N(u)} y^v_{is} \le 1 && u \in V. && \text{(5b)}
\end{aligned}
$$

4 Lower bounds

In this section, we study lower bounds on delay k for several restrictions of input graphs. An optimal solution is obtained by solving a sequence of decision problems with varying k. It is therefore desirable to determine tight lower and upper bounds in order to arrive in the optimum after solving as few decision problems as possible. Obvious bounds for a general graph instance are given by

Observation 3 *For an instance (G,S) of Problem 1,*

$$\left\lceil \log \frac{n}{s} \right\rceil \leq k \leq n - s.$$

Consider a d-regular graph with one source s. The broadcast forest T consists of a single tree T_s. We investigate the number of leaves in T_s and derive a lower bound on the dealy for this graph class. For $k = 1, 2$, $L(T_s^1) = L(T_s^2) = 2$. For $k \geq 3$, $L(T_s^k)$ corresponds to the number of nodes with degree $1, \ldots, d-1$ in T_s^{k-1}. It can also be interpreted as the sum of number of leaves in $T_s^{k-d+1}, \ldots, T_s^{k-1}$, which leads to the following formula

$$L(T_s^k) = \sum_{i=k-d+1}^{d-1} L(T_s^i). \tag{6}$$

Eq. (6) exactly corresponds to the recursive definition of Fibonacci sequence of order $d-1$. As each of the two base cases $L(T_s^l$ and $L(T_s^2)$ equal double the base cases of the Fibonacci sequence, the number of leaves in time step k is calculated as

$$L(T_s^k) = 2F_k^{d-1}. \tag{7}$$

Since each node in H_s^k had been a leaf in exactly one time step, the number of nodes in T_s^k can be expressed as

$$|V_k| = 2\sum_{i=1}^{k} F_i^{d-1}. \tag{8}$$

Proposition 2 *For a d-regular graph on n and s source, the lower bound on delay is*

$$\left\lceil \min\{k : 2\sum_{i=1}^{k} F_i^{d-1} \geq n\}/s \right\rceil. \tag{9}$$

Proof □

An additional knowledge of a degree sequence of G can be exploited. Alg. 1 calculates a lower bound on the delay when given number of nodes, sources and a degree sequence in G as an input. The algorithm iteratively updates possible node degrees in F_i in each time step i, and records the maximum potential number of nodes in V_i. For the purpose of finding lower bounds, it is assumed that each node $v \in V_i$ with $\deg_{F_i}(v) < \deg_G(v)$ informs a new uninformed node. The number of iterations is then the lowest possible delay for given input. Once a node v reaches its maximum degree, i.e., when $\deg_{F_i} = \deg_G(v)$ for some i, v does not inform any other node in the next time steps.

The input is assumed to be correct in the sense that a graph with given degree sequence exists, and by definition, the degree sequence is ordered non-increasingly. For each iteration k, Alg. 1 stores degrees of nodes in F_k in variables $a_1, \ldots, a_n$. Note that the forest F_k is not actually constructed. The algorithm operates merely with potential degrees of nodes in F_k. Next, variable c keeps the value $|V_k|$, i.e., the number nodes informed within k steps. Finally, c_n stores $|V_k \setminus V_{k-1}|$, thus the number of nodes newly informed in time step k.

```
   Data: n, s, d_1, ..., d_n ∈ ℕ, s ≤ n,
         1 ≤ d_n ≤ ··· ≤ d_2 ≤ d_1
1  a_1, ..., a_{2s} ← 1;
2  for i = 2s + 1, ..., n do
3  |  a_i ← 0;
4  end
5  c ← 2s;// every source informes a new node
6  k ← 1;
7  while c < n do
8  |  k ← k + 1;
9  |  c_n ← 0;
10 |  for i = 1, ..., c do
11 |  |  if a_i < d_i then
12 |  |  |  a_i ← a_i + 1;
13 |  |  |  c_n ← c_n + 1;
14 |  |  |  if c + c_n < n then
15 |  |  |  |  a_{c+c_n} ← 1;// Newly informed node
16 |  |  |  end
17 |  |  end
18 |  end
19 |  c ← c + c_n;
20 end
21 return k;
```

Algorithm 1: Lower bound exploiting distribution of degrees

Proposition 3 *If G is an arbitrary graph with n nodes, s sources and node degrees d_i, $1 \leq i \leq n$, Alg. 1 calculates a lower bound for the delay k from Problem 1 in G.*

Proof (Idea:) The most optimistic scenario is when F_k consists of full binomial trees. In such a case, for $u \in V_i$ and $v \in V_j$ we have $\deg_{F_k}(v) \leq \deg_{F_k}(u) \Leftrightarrow i \leq j$. We can therefore assert that in an optimal solution, $\deg_G(u) \leq \deg_G(v) \Leftrightarrow i \leq j$. Whenever $\deg_{F_k}(v) < \deg_G(v)$, v informs a new node in $V \setminus V_k$ with maximum degree in G, which is ensured by the decreasing order of input node degrees. For every $v \in V$, $\deg_{F_k}(v) \leq \deg_G(v)$ always holds. Further, when the condition $c < n$ is tested on line 7, c contains the correct value of number of nodes in V_k. The returned value is therefore a lower bound on the delay. □

References

1. Chu, X., Chen, Y., Time division intersatellite link topology generation problem: Modeling and solution, International Journal of Satellite Communications and Networking, 194 – 206, 36 (2017)
2. Grigni, M., Peleg, D., Tight bounds on minimum broadcast networks Networks, 207-222, 4 (1991)
3. Jansen, K., Müller, H., The minimum broadcast time problem for several processor networks, Theoretical Computer Science, 69 – 85, 147 (1995)
4. Middendorf, M., Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2, Information Processing Letters, 281 – 287, 46 (1993)