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 an 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, $0 \leq i \leq k$. For each

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

S. Author
second address

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 optimization problem in question is defined as follows [2,3]:

Problem 1 Given $G = (V, E)$ and $S \subseteq V$, find a sequence $S = V_0 \subseteq \cdots \subseteq V_k = V$ of minimum lenght 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 $deg_G(v)$, $dev_G^-(v)$ and $deg_G^+(v)$, respectively. Whenever there is no danger of confusion, the subscrip G is omitted. The set of neighbors of $v \in V$ in G is denoted by $N(v)$.

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

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

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:

$$
\begin{aligned}
& \min c && && \text{(1a)}\\
& \sum_{i\in N(s)} x^1_{si} \le 1 && s\in S, && \text{(1b)}\\
& \sum_{t=1}^{n-1}\sum_{j\in N(i)} x^t_{ji} = 1 && i\in V\setminus S, && \text{(1c)}\\
& \sum_{j\in N(i)} x^t_{ij} \le 1 && t=1,\ldots,n-1, i\in V, && \text{(1d)}\\
& \sum_{j\in N(i)} x^t_{ij} \le \sum_{u=1}^{t-1}\sum_{k\in N(i)\setminus\{j\}} x^u_{ki} && i\in V\setminus S, t=2,\ldots,n-1, && \text{(1e)}\\
& \sum_{t=1}^{n-1} t\cdot x^t_{ij} \le c && (i,j)\in A, && \text{(1f)}\\
& x^1_{ij} = 0 && (i,j)\in A, i\notin S, && \text{(1g)}\\
& && x\in\{0,1\}^{A\times V}, c\in\{1,\ldots,n-1\}. && \text{(1h)}
\end{aligned}
$$

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 arcs along which signals initiated at sources $s \in S$ are transmitted form binomial trees B^k of order k rooted at s. Hence, if there is a partition of G into s truncated binomial trees of order at most k rooted at sources, then (G, S, k) is a YES instance of Problem 2. Finding a partition of G into s truncated binomial trees can be equivalently formulated as finding a partition of G'' into s (complete) binomial trees, where G'' is constructed from

G as follows: Let $\alpha := s \cdot 2^k - |V|$, and let $K_\alpha = (V_\alpha, E_\alpha)$ be a complete graph on α nodes. Each node in K_α is connected to every node $v \in V$ in the original graph G. Thus, $G'' = (V'', E'')$ with $V'' = V \cup V_\alpha$ and $E'' = E \cup E_\alpha \cup \{\{u, v\} : u \in V \wedge v \in V_\alpha\}$. The set of arcs A'' is constructed by creating two arcs of opposite orientation for each edge, but arcs with orientation from V_α to V are excluded. Formally, $A'' = A \cup \{(u, v), (v, u) : \{u, v\} \in E_\alpha\} \cup \{(u, v) : u \in V \wedge v \in V_\alpha\}$.

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

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

Proposition 1 *An instance (G, S, k) of Problem 2 is a YES-instance iff there exists a partition of G'' into s directed binomial trees $B_1^k, \ldots, B_s^k$ 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 $B_1^k, \ldots B_s^k$. Nodes in V are divided into a sequence of subsets $S = V_0 \subseteq \cdots \subseteq V_k = V$ such that $V_i = \{v \in V : 1 \leq \beta(v) \leq 2^i\}$, 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 $B_1^k, \ldots, B_s^k$.

Conversely, suppose there is a sequence of subsets and a mapping p in G with the desired properties. For $1 \leq i \leq k$, $|V_i| \geq 2|V_{i-1}|$ 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 $p(u) = v$. Truncated binomial trees covering G can then be constructed simply by following the mapping p. The remaining arcs that complement binomial trees are distributed along $A'' \setminus A$. □

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. An obvious weakness of this approach is that the number of nodes increases to $|V''| = \mathcal{O}(ns)$, and the dimension of variables is thus $\mathcal{O}(n^2 s^2)$. However, once a suitable partition is found, the arcs of binomial trees contained in K_α can be diversely shuffled while preserving the layour of binomial trees in G. Instead of adding the entire complete graph K_α, a single node v_0 with a loop (v_0, v_0) is connected as an apex to the original G. Let us denote this multigraph as $G' = (V', E')$, where $V' = V \cup \{v_0\}$, $E' = E \cup \{\{u, v_0\} : u \in V\} \cup \{\{v_0\}\}$. The arc set is then analogously defined as $A' = A \cup \{(u, v_0) : u \in V\} \cup \{(v_0, v_0)\}$. The requirement for partition into binomial trees has to be adjusted accordingly. The subtrees contained in G remain unchanged, every arc $(u, v) \in A_i^k, i = 1, \ldots, s$ in G'' with $u \in V$ and $v \in V_\alpha$ becomes (u, v_0) in G', and every $(u, v) \in A_\alpha$ becomes (v_0, v_0). So, v_0 acts as a universal node that can substitute several nodes in each binomial tree.

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

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

The values of the variables must fulfill constraints

$$\begin{aligned}
\sum_{i \in I} \sum_{s \in S} y_{is}^v &= 1 & v \in V, \quad & \text{(4a)}\\
\sum_{v \in V'} y_{is}^v &= 1 & i \in I, s \in S, \quad & \text{(4b)}\\
y_{1s}^s &= 1 & s \in S, \quad & \text{(4c)}\\
y_{is}^u + y_{\ell s}^v &\leq 1 & i \in I, \ell \in C(i), s \in S, u, v \in V', (u, v) \notin A', \quad & \text{(4d)}\\
& & y \in \{0,1\}^{I \times S \times V'}. \quad & \text{(4e)}
\end{aligned}$$

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 a part of several binomial trees. By (4b) is ensured that there is always exactly one i-th node of each binomial tree. Next, (4c) enforce that source nodes are the first nodes in corresponding binomial trees, in accordance with definition of the mapping β. The remaining constraints (4d) guarantee that the arcs of binomial trees follow arcs in A'. The definition of A' also prevents arcs of the binomial trees to be oriented from V_α to V. In other words, once the signal leaves the original graph G and enters v_0, it cannot return back to G. Without this requirement, it could be possible to find a partition of G' into binomial trees, even though no partition of G into truncated binomial trees exists.

4 Lower bounds

In this section, we investigate 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,*

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

```
Data: n, s, d
for i = 1, ..., d − 1 do
|   a_i ← 2^{d−i};
end
a ← 2;
k ← d;
c ← a + Σ_{i=1}^{d−1} a_i;
while c < n do
|   a ← a + a_{d−1};
|   tmp ← Σ_{i=1}^{d−1} a_i;
|   for i = 1, ..., d − 2 do
|   |   a_{d−i} ← a_{d−(i+1)};
|   end
|   a_1 ← tmp;
|   c ← a + Σ_{i=1}^{d−1} a_i;
|   k ← k + 1;
end
return ⌈k/s⌉;
```

Algorithm 1: Lower bound for d-regular graphs

Proposition 2 *If G is an arbitrary graph with n nodes, s sources and each node has a degree at most d, Alg. 1 calculates a lower bound for k from Problem 1 in G.*

Proof □

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. Jansen, K., Müller, H., The minimum broadcast time problem for several processor networks, Theoretical Computer Science, 69 – 85, 147 (1995)
3. Middendorf, M., Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2, Information Processing Letters, 281 – 287, 46 (1993)