Deep Heuristic for Broadcasting in Arbitrary Networks

Authors: Hovhannes A. Harutyunyan, Narek Hovhannisyan, Rakshit Magithiya
Presenter: Hovhannes A. Harutyunyan

Department of Computer Science and Software Engineering, Concordia University, Canada
haruty@cs.concordia.ca

July 11th, 2022

Outline

Outline

Introduction

- Growth of using computer networks,
- Great attention to all major problems in this area,
- Information dissemination,
- Broadcasting:
    - ◇ Process of distributing a message starting from a single node (*originator*) to all other nodes of the network using the network's links.

Introduction

- The process of broadcasting is split into discrete time units.

Introduction

- The process of broadcasting is split into discrete time units.
- Initially, only one vertex (*originator*) has the message.

Introduction

- The process of broadcasting is split into discrete time units.
- Initially, only one vertex (*originator*) has the message.
- In each time unit, a vertex with the message (*sender*) can *call* at most one uninformed neighbor (*receiver*).

Introduction

- The process of broadcasting is split into discrete time units.
- Initially, only one vertex (*originator*) has the message.
- In each time unit, a vertex with the message (*sender*) can *call* at most one uninformed neighbor (*receiver*).
- All the calls are in parallel during the same time unit.

Introduction

- The process of broadcasting is split into discrete time units.
- Initially, only one vertex (*originator*) has the message.
- In each time unit, a vertex with the message (*sender*) can *call* at most one uninformed neighbor (*receiver*).
- All the calls are in parallel during the same time unit.
- If all the vertices in the graph have the message, the process halts.

Introduction

- The network: $G = (V, E)$, originator $u \in V$.

Introduction

- The network: $G = (V, E)$, originator $u \in V$.
- $b(u, G)$: minimum time required to finish the broadcasting originating from u.

Introduction

- The network: $G = (V, E)$, originator $u \in V$.
- $b(u, G)$: minimum time required to finish the broadcasting originating from u.
- $b(G) = \max\{b(u, G) | u \in V(G)\}$
  - ◇ For any graph: $b(G) \geq \lceil \log n \rceil$

Introduction

- The network: $G = (V, E)$, originator $u \in V$.
- $b(u, G)$: minimum time required to finish the broadcasting originating from u.
- $b(G) = \max\{b(u, G) | u \in V(G)\}$
    - ◇ For any graph: $b(G) \geq \lceil \log n \rceil$
- **Broadcast scheme** is a sequence $(C_1, C_2, \ldots, C_t)$, where C_i is the set of calls performed in time unit i.
- **Optimal broadcast scheme**, denoted by $\mathcal{S}(G, v)$, is a broadcast scheme for an originator u that uses $b(G, u)$ time units.

Introduction

- The network: $G = (V, E)$, originator $u \in V$.
- $b(u, G)$: minimum time required to finish the broadcasting originating from u.
- $b(G) = \max\{b(u, G) | u \in V(G)\}$
  - ◇ For any graph: $b(G) \geq \lceil \log n \rceil$
- **Broadcast scheme** is a sequence $(C_1, C_2, ..., C_t)$, where C_i is the set of calls performed in time unit i.
- **Optimal broadcast scheme**, denoted by $\mathcal{S}(G, v)$, is a broadcast scheme for an originator u that uses $b(G, u)$ time units.
- Two major lines or research:
  - ◇ Construct graphs (networks) with given broadcast times
  - ◇ **Given a graph and message originator, find the broadcast time**

Introduction - Broadcast time problem

- The *broadcast time problem (BTP)* is defined as

Problem 1 (BTP)

Instance: *(G, v, t), where $G = (V, E)$ is a graph, $v \in V$ is the originator, and t is a natural number.*
Output: *"Yes" if $b(G, v) \leq t$; "No" otherwise.*

Introduction - Broadcast time problem

- The *broadcast time problem (BTP)* is defined as

Problem 1 (*BTP*)

Instance: *(G, v, t), where $G = (V, E)$ is a graph, $v \in V$ is the originator, and t is a natural number.*
Output: *"Yes" if $b(G, v) \leq t$; "No" otherwise.*

- NP-Complete in arbitrary graphs [17].
  - Remains NP-Complete even in more restricted families of graphs such as 3-regular planar graphs [14].

Introduction - Previous work

- Exact algorithms:
  - Trees [17],
  - Unicyclic graph [11],
  - Necklace graph [6],
  - Tree of cycles, Tree of cliques [13].

Introduction - Previous work

- Exact algorithms:
  - ◇ Trees [17],
  - ◇ Unicyclic graph [11],
  - ◇ Necklace graph [6],
  - ◇ Tree of cycles, Tree of cliques [13].
- Approximation algorithms:
  - ◇ Current best approximation algorithm: $\mathcal{O}(\frac{\log n}{\log \log n})$-approximation ratio [3],
  - ◇ NP-hard to approximate with ratio $3 - \epsilon$, where $\epsilon > 0$ [2].

Introduction - Previous work

- Heuristics:
  - ◇ Exponential Backtracking Algorithm [15]
  - ◇ Several optimized heuristics based on the backtracking algorithm [16]
  - ◇ Genetic algorithm: $O(|E||V|^3)$ [4]

Introduction - Previous work

- Heuristics:
  - Exponential Backtracking Algorithm [15]
  - Several optimized heuristics based on the backtracking algorithm [16]
  - Genetic algorithm: $O(|E||V|^3)$ [4]
- Comparable heuristics:
  - Round Heuristic: $O(R|V||E|\log|V|)$ [1]
  - Tree Based Algorithm: $O(R|E|)$ [7]

Outline

Deep Heuristic - Definitions

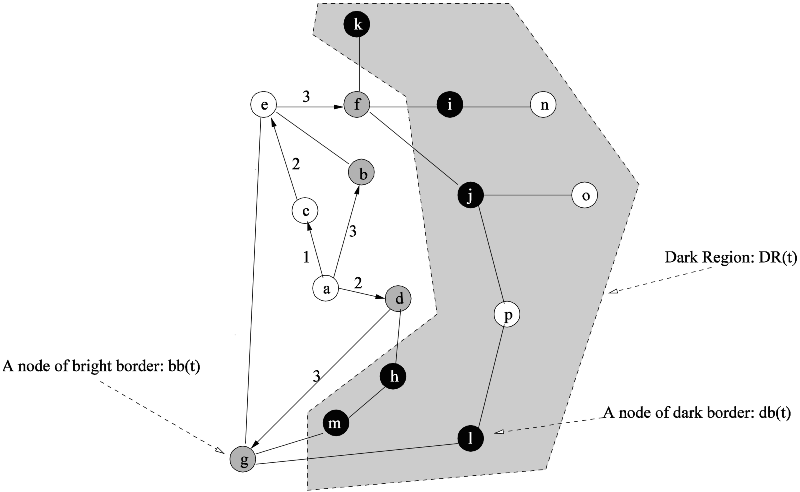


Figure: Definitions of Graph Parts

Deep Heuristic - Definitions

Definition 2

For a given graph G at round t, there are two kinds of regions according to the situation of the message distribution, the dark region and the bright region.

- The **dark region**, denoted by $DR(t)$, is a subset of nodes in G that is composed of all uninformed nodes at the beginning of round t.
- The **bright region**, denoted by $BR(t)$, is a subset of nodes in G that is composed of all informed nodes at the beginning of round t.
- Those nodes in $DR(t)$ that have informed neighbors, compose the **dark border**, denoted by $db(t)$.
- The **bright border** $bb(t)$ is composed of those informed nodes that have uninformed neighbors.
- The edges that cross between the dark region and the bright region are called **cross board edges**, which are denoted by $cbe(t)$.

Deep Heuristic - Definitions

Definition 3

For a graph and an uninformed node v at round t, there is a **shortest distance** from node v to a node in $bb(t)$. The shortest distance is denoted as $D(v, t)$.

Definition 4

Child, parent and descendants: Given an uninformed vertex u and its uninformed neighbor v, if $D(u, t) = D(v, t) + 1$, one can say u is a **child** of v, and v is the **parent** of u. The vertex u, its children and its children's children are all called v's **descendants**.

Definition 5

For a graph $G = (V, E)$ and an uninformed node v at round t, the **descendant graph** of v consists of the node v and all its descendants and is denoted by $DG(V, E, v)$, or rather $DG(v)$.

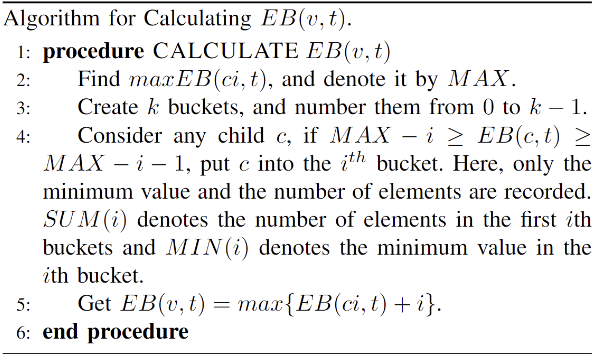
Deep Heuristic - Definitions

Definition 6

Estimated time: in order to estimate the broadcast time of $DG(v)$ in round t, we use $EB(v, t)$. $EB(v, t)$ is defined recursively as follows:

- $EB(v, t) = 0$, if node v has no children.
- If v has k children, $c_1, c_2, \dots, c_k$, and all these k children are listed in order of $EB(c_i, t) \geq EB(c_{i+1}, t)$, then $EB(v, t) = max\{EB(c_i, t) + i\}$, for $1 \leq i \leq k$.

Deep Heuristic - Definitions



Algorithm for Calculating $EB(v,t)$.

```
procedure CALCULATE $EB(v,t)$
    Find $maxEB(ci,t)$, and denote it by $MAX$.
    Create $k$ buckets, and number them from $0$ to $k-1$.
    Consider any child $c$, if $MAX - i \geq EB(c,t) \geq MAX - i - 1$, put $c$ into the $i^{th}$ bucket. Here, only the minimum value and the number of elements are recorded. $SUM(i)$ denotes the number of elements in the first $i$th buckets and $MIN(i)$ denotes the minimum value in the $i$th bucket.
    Get $EB(v,t) = max\{EB(ci,t)+i\}$.
end procedure
```

Lemma 7

$EB(v,t) = max\{SUM(i) + MIN(i)\}$, *for* $0 \leq i < k$.

Deep Heuristic - Improvements

- Perform broadcasting using TBA,
- At round t, the subgraphs originated from vertex d and the subgraph of s hold different density properties,
- The subgraph of d is dense and the subgraph of s is sparse,
- $EB(d) \geq EB(e)$,
- Prevent sending the information to a dense subgraph at time unit t.

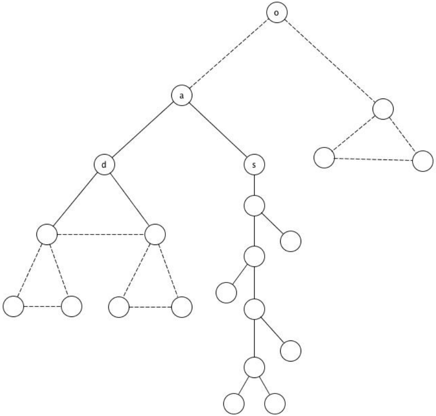


Figure: Example Graph G with two subgraphs from vertex a.

Deep Heuristic

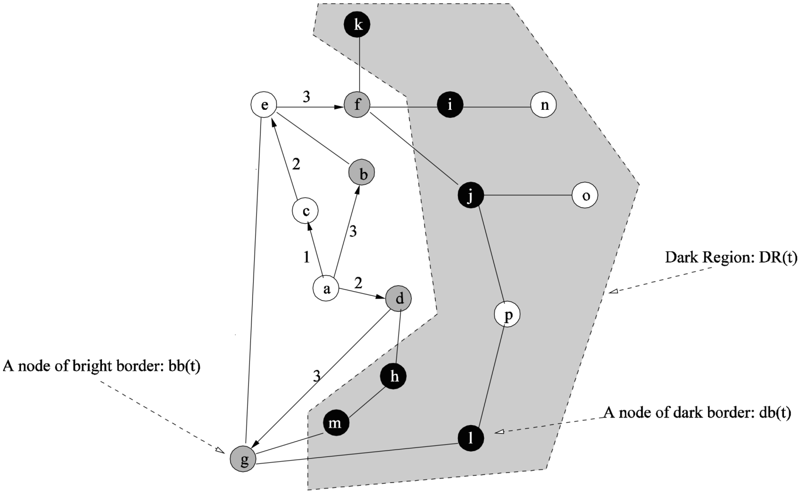
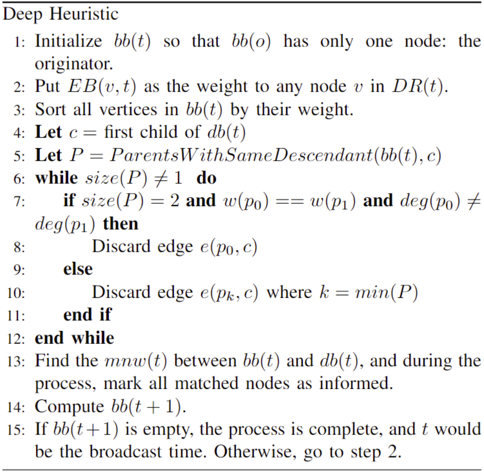


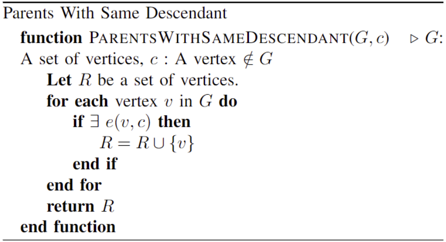
Figure: An example of possible states during the broadcasting process

Deep Heuristic



Deep Heuristic

```
1: Initialize $bb(t)$ so that $bb(o)$ has only one node: the originator.
2: Put $EB(v,t)$ as the weight to any node $v$ in $DR(t)$.
3: Sort all vertices in $bb(t)$ by their weight.
4: **Let** $c =$ first child of $db(t)$
5: **Let** $P = ParentsWithSameDescendant(bb(t), c)$
6: **while** $size(P) \neq 1$ **do**
7:     **if** $size(P) = 2$ **and** $w(p_0) == w(p_1)$ **and** $deg(p_0) \neq deg(p_1)$ **then**
8:         Discard edge $e(p_0, c)$
9:     **else**
10:        Discard edge $e(p_k, c)$ where $k = min(P)$
11:    **end if**
12: **end while**
13: Find the $mnw(t)$ between $bb(t)$ and $db(t)$, and during the process, mark all matched nodes as informed.
14: Compute $bb(t+1)$.
15: If $bb(t+1)$ is empty, the process is complete, and $t$ would be the broadcast time. Otherwise, go to step 2.
```



Parents With Same Descendant

```
function PARENTSWITHSAMEDESCENDANT($G, c$)    ▷ $G$: A set of vertices, $c$ : A vertex $\notin G$
    Let $R$ be a set of vertices.
    for each vertex $v$ in $G$ do
        if $\exists\ e(v, c)$ then
            $R = R \cup \{v\}$
        end if
    end for
    return $R$
end function
```

Deep Heuristic - Time Complexity

- Weight assignment to each node (step 2): $O(|E|)$
- Sorting the bright border (step 3): $O(|V|)$
- Finding nodes with common descendants (step 5): $O(|V|)$
- Find a matching between bright and dark borders (step 13): $O(|E|)$

Deep Heuristic - Time Complexity

- Weight assignment to each node (step 2): $O(|E|)$
- Sorting the bright border (step 3): $O(|V|)$
- Finding nodes with common descendants (step 5): $O(|V|)$
- Find a matching between bright and dark borders (step 13): $O(|E|)$
- Each broadcasting round: $O(|E|)$

Deep Heuristic - Time Complexity

- Weight assignment to each node (step 2): $O(|E|)$
- Sorting the bright border (step 3): $O(|V|)$
- Finding nodes with common descendants (step 5): $O(|V|)$
- Find a matching between bright and dark borders (step 13): $O(|E|)$
- Each broadcasting round: $O(|E|)$
- Overall: $O(|E| \cdot b)$, where b is the broadcast time returned by the algorithm.

Outline

Deep Heuristic - Practical Results

- The results we obtained are compared with the results of all the heuristics presented in the previous chapters.
    - ◇ The result of Round Heuristic from [1] (RH)
    - ◇ The Tree Based Algorithm obtained from [7] (TBA)
    - ◇ The Random algorithm from [8] (P-R)
    - ◇ The Semi-Random algorithm from [8] (S-R)
    - ◇ The Minimum-Weight Cover heuristic from [8] (MWC)
    - ◇ The Minimum-Weight Cover Modified heuristic [8] (MWC-M)

Deep Heuristic - Practical Results

- Some abbreviations used:
    - ◇ OPT: The optimal broadcast time in the respective topology
    - ◇ LOW: The best known theoretical lower bound on the broadcast time in the respective topology
    - ◇ UP: The best known theoretical upper bound on the broadcast time in the respective topology
    - ◇ D: The dimension of the topology

Deep Heuristic - Practical Results

P	Edges	RH	TBA	MWC	MWC-M	P-R	S-R	DH
0.015	316	10	10	11	11	10	10	11
0.016	346	10	10	11	11	10	10	11
0.017	373	10	10	11	11	10	10	11
0.018	388	9	9	11	11	10	10	10
0.019	391	11	11	10	10	10	10	10
0.02	411	9	9	10	10	10	10	9
0.022	423	9	9	10	10	10	10	9
0.024	475	8	8	10	11	10	10	7
0.025	494	9	8	11	11	10	10	8
0.026	507	8	8	11	10	10	10	8

Table: Practical results for GT-ITM Random model with 200 vertices

Deep Heuristic - Practical Results

Edges	RH	TBA	MWC	MWC-M	P-R	S-R	DH
1169	14	13	14	13	13	13	14
1190	14	14	14	14	13	13	13
1200	16	15	14	14	13	13	15
1206	14	14	14	14	14	14	15
1219	15	14	14	14	13	13	14
1222	15	14	15	15	14	14	14
1231	14	13	14	14	13	13	14
1232	14	13	14	14	13	13	13
1247	13	14	14	14	14	14	13
1280	14	13	14	14	13	14	13

Table: Practical results for GT-ITM Transit-Stub model with 600 vertices

Deep Heuristic - Practical Results

Edges	RH	TBA	MWC	MWC-M	P-R	S-R	DH
2115	17	16	16	17	16	16	16
2121	17	17	16	15	15	15	16
2142	16	15	16	15	15	15	16
2151	15	15	16	15	15	15	17
2169	17	17	16	16	15	15	15
2177	18	17	16	16	16	16	16
2185	16	16	15	15	15	15	16
2219	17	16	15	16	15	15	16
2220	15	15	15	15	14	14	15
2230	16	15	16	16	15	15	15

Table: Practical results for GT-ITM Transit-Stub model with 1056 vertices

Deep Heuristic - Practical Results

Edges	RH	TBA	MWC	MWC-M	P-R	S-R	DH
354	17	17	16	16	16	16	17
414	15	14	14	14	14	14	14
474	14	13	14	14	14	14	14
357	17	17	16	16	16	16	16
477	15	14	14	14	14	14	14
535	16	15	13	13	13	13	15
422	15	14	14	14	14	14	13
482	14	13	14	14	14	14	13
541	14	14	14	13	13	13	13

Table: Practical results for Tiers model with 355 vertices

Deep Heuristic - Practical Results

Edges	RH	TBA	MWC	MWC-M	P-R	S-R	DH
1214	22	21	21	21	21	21	21
1324	23	21	21	20	20	20	21
1447	22	21	22	22	22	22	21
1106	24	24	21	21	21	21	22
1216	22	21	21	21	21	21	22
1326	23	21	20	21	20	20	21
1110	24	23	21	21	21	21	21
1220	22	21	21	21	21	21	20
1331	20	20	20	20	20	20	20
1449	21	20	22	22	22	22	20

Table: Practical results for Tiers model with 1105 vertices

Deep Heuristic - Practical Results

Edges	MWC	MWC-M	P-R	S-R	DH
420	22	22	22	22	28
840	15	15	15	14	14
1260	13	13	13	12	14
1680	14	14	13	13	12
2092	15	14	13	13	12
2440	16	16	14	14	12
2671	17	17	16	16	14
2733	18	18	16	15	13
2755	19	18	18	18	14

Table: Practical results for BRITE Top-down Waxman model with 400 vertices

Deep Heuristic - Practical Results

Edges	MWC	MWC-M	P-R	S-R	DH
399	22	22	22	22	28
777	17	17	17	16	16
1134	15	14	14	13	13
1470	14	13	13	13	12
1785	14	14	13	13	12
2079	14	14	13	13	12
2352	14	14	14	14	12
2604	16	16	14	14	12
2835	16	16	16	15	11

Table: Practical results for BRITE Top-down BA model with 400 vertices

Deep Heuristic - Practical Results

Edges	MWC	MWC-M	P-R	S-R	DH
1020	29	29	29	29	30
2040	19	19	18	18	17
3060	19	19	18	17	17
4080	17	18	17	16	16
5100	18	18	16	16	16
6108	18	18	17	16	14
7116	19	18	17	17	14
8117	19	19	17	18	15
9122	19	19	17	19	14

Table: Practical results for BRITE Top-down Waxman model with 1000 vertices

Deep Heuristic - Practical Results

Edges	MWC	MWC-M	P-R	S-R	DH
999	35	35	35	35	36
1977	23	23	22	22	20
2934	24	25	23	21	17
3870	22	22	21	18	17
4785	20	20	19	17	16
5679	19	19	18	17	15
6552	19	18	18	17	16
7404	20	19	17	17	16
8235	19	19	17	18	15

Table: Practical results for BRITE Top-down BA model with 1000 vertices

Outline

Conclusion and Future Works

- Designed an efficient heuristic, which improves behavior of some existing heuristics in certain key situations.

Conclusion and Future Works

- Designed an efficient heuristic, which improves behavior of some existing heuristics in certain key situations.
- Very well suitable for graphs where most of the vertices have high degree and higher density.

Conclusion and Future Works

- Designed an efficient heuristic, which improves behavior of some existing heuristics in certain key situations.
- Very well suitable for graphs where most of the vertices have high degree and higher density.
- Based on our extensive simulations, the Deep Heuristics perform exceptionally well in some of the models representing real-world networks.

Conclusion and Future Works

- Designed an efficient heuristic, which improves behavior of some existing heuristics in certain key situations.
- Very well suitable for graphs where most of the vertices have high degree and higher density.
- Based on our extensive simulations, the Deep Heuristics perform exceptionally well in some of the models representing real-world networks.
- Time complexity lower than that of many other heuristics mentioned in this paper.

Thank you!

Important References I

[1] R. Beier and J.F. Sibeyn.
A powerful heuristic for telephone gossiping.
In *Proceedings of the 7th Colloquium on Structural Information and Communication Complexity*, (SIROCCO), pages 17–35. Carleton Scientific, 2000.

[2] M. Elkin and G. Kortsarz.
Sublogarithmic approximation for telephone multicast: Path out of jungle (extended abstract).
In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, (SODA), pages 76–85. Society for Industrial and Applied Mathematics, 2003.

[3] M. Elkin and G. Kortsarz.
A combinatorial logarithmic approximation algorithm for the directed telephone broadcast problem.
SIAM journal on Computing, 35(3):672–689, 2005.

Important References II

[4] P. Fraigniaud and S/ Vial.
Comparison of heuristics for one-to-all and all-to-all communications in partial meshes.
Parallel Processing Letters, 9(01):9–20, 1999.

[5] H. Grigoryan and H.A. Harutyunyan.
New lower bounds on broadcast function.
In *International Conference on Algorithmic Applications in Management*, (AAIM), pages 174–184. 2014.

[6] H. A. Harutyunyan, G. Laza, and E. Maraachlian.
Broadcasting in necklace graphs.
In *Canadian Conference on Computer Science & Software Engineering (C3S2E '09)*, pages 253–256. ACM, 2009.

[7] H. A. Harutyunyan and B. Shao.
An efficient heuristic for broadcasting in networks.
Journal of Parallel and Distributed Computing, 66(1):68–76, 2006.

Important References III

[8] H. A. Harutyunyan and W. Wang.
Broadcasting algorithm via shortest paths.
In *2010 IEEE 16th International Conference on Parallel and Distributed Systems*, pages 299–305. IEEE, 2010.

[9] H.A. Harutyunyan, A.L. Liestman, and B. Shao.
A linear algorithm for finding the k-broadcast center of a tree.
Networks, 53(3):287–292, 2009.

[10] H.A. Harutyunyan and E. Maraachlian.
Linear algorithm for broadcasting in unicyclic graphs.
In *13th International Conference on Computing and Combinatorics*, (COCOON), pages 372–382. Springer, 2007.

[11] H.A. Harutyunyan and E. Maraachlian.
On broadcasting in unicyclic graphs.
Journal of Combinatorial Optimization, 16(3):307–322, 2008.

Important References IV

[12] M. Čevnik and J. Žerovnik.
Broadcasting on cactus graphs.
Journal of Combinatorial Optimization, 33(1):292–316, 2017.

[13] E. Maraachlian.
Optimal broadcasting in treelike graphs.
PhD thesis, Concordia University, 2010.

[14] M. Middendorf.
Minimum broadcast time is np-complete for 3-regular planar graphs and deadline 2.
Information Processing Letters, 46(6):281–287, 1993.

[15] P. Scheuermann and M. Edelberg.
Optimal broadcasting in point-to-point computer networks.
Dep. Elec. Eng. Comput. Sci., Northwestern Univ., Evanston, IL, Tech. Rep, 1981.

Important References V

[16] P. Scheuermann and G. Wu.
Heuristic algorithms for broadcasting in point-to-point computer networks.
IEEE Transactions on Computers, 33(09):804–811, 1984.

[17] P.J. Slater, E.J. Cockayne, and S.T. Hedetniemi.
Information dissemination in trees.
SIAM Journal on Computing, 10(4):692–701, 1981.

[18] Y.-H. Su, C.-C. Lin, and DT Lee.
Broadcasting in heterogeneous tree networks.
In *16th International Computing and Combinatorics Conference,*, (COCOON), pages 368–377. Springer, 2010.