Introduction
Preliminaries and Literature Review
Contribution
Hypercube of Trees
Fully Connected Trees
Future Works
Development for our heuristic
Messy broadcasting
Broadcasting using universal lists
Conclusion and Timeline

Broadcasting Problem in a Specific Class of Graphs

By: Saber Gholami
Supervisor: Professor Hovhannes Harutyunyan

Concordia University,
Department of Computer Science and Software Engineering

April 28th, 2021

Outline

1. Introduction
2. Preliminaries and Literature Review
3. Contribution
    - Hypercube of Trees
    - Fully Connected Trees
4. Future Works
    - Development for our heuristic
    - Messy broadcasting
    - Broadcasting using universal lists
5. Conclusion and Timeline

Outline

1 Introduction
2 Preliminaries and Literature Review
3 Contribution
    Hypercube of Trees
    Fully Connected Trees
4 Future Works
    Development for our heuristic
    Messy broadcasting
    Broadcasting using universal lists
5 Conclusion and Timeline

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.

Outline

1 Introduction
2 Preliminaries and Literature Review
3 Contribution
  Hypercube of Trees
  Fully Connected Trees
4 Future Works
  Development for our heuristic
  Messy broadcasting
  Broadcasting using universal lists
5 Conclusion and Timeline

Preliminaries

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

Preliminaries

- 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: $b(G) \geq \lceil \log n \rceil$

Preliminaries

- 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: $b(G) \geq \lceil \log n \rceil$
- Two major problems in this area:
    - **Broadcast time problem,**
    - Network design.

Literature Review - Broadcast time problem

- NP-Complete in arbitrary graphs [22],
    - Remains NP-Complete even in more restricted families of graphs such as planar and decomposable graphs [17].

Literature Review - Broadcast time problem

- NP-Complete in arbitrary graphs [22],
  - ◇ Remains NP-Complete even in more restricted families of graphs such as planar and decomposable graphs [17].
- Exact solutions:
  - ◇ Trees [22],
  - ◇ Unicyclic graph [11],
  - ◇ Necklace graph [9]
  - ◇ Tree of cycles, Tree of cliques [19]

Literature Review - Broadcast time problem - cont.

- Approximation algorithms:
  - $(\sqrt{|V|})$-additive approximation [18],
  - $(\frac{\log^2 |V|}{\log \log |V|})$-approximation algorithm [20],
  - $(\frac{\log k}{\log \log k})$-approximation algorithm for multicasting [5]
    - Or a $(\frac{\log |V|}{\log \log |V|})$-approximation solution for broadcasting [5]

Literature Review - Broadcast time problem - cont.

- Approximation algorithms:
  - ($\sqrt{|V|}$)-additive approximation [18],
  - ($\frac{\log^2|V|}{\log\log|V|}$)-approximation algorithm [20],
  - ($\frac{\log k}{\log\log k}$)-approximation algorithm for multicasting [5]
    - Or a ($\frac{\log|V|}{\log\log|V|}$)-approximation solution for broadcasting [5]
- Inapproximability of the problem:
  - broadcast time could not be approximated within a factor of $\frac{57}{56} - \varepsilon$ for an arbitrary graph [21]
  - NP-Hard to approximate the broadcast problem within a ratio of $(3 - \varepsilon)$, for any $\varepsilon > 0$ [4].

Literature Review - Broadcast time problem - cont.

- Heuristics:
  - ◇ Round Heuristic: $O(R|V||E|\log|V|)$ [1]
  - ◇ Tree Based Algorithm: $O(R|E|)$ [12]
  - ◇ Random heuristic: $O(|E|)$ [14]
    - · Semi-random version: $O(|E|)$ [13]
  - ◇ Based on Genetic Algorithm [16]

Outline

1 Introduction

2 Preliminaries and Literature Review

3 Contribution
  Hypercube of Trees
  Fully Connected Trees

4 Future Works
  Development for our heuristic
  Messy broadcasting
  Broadcasting using universal lists

5 Conclusion and Timeline

Hypercube of Trees

- A Hypercube of Trees HT_k:
  - ◇ A hypercube of dimension k +
  - ◇ 2^k arbitrary trees.
- Very useful structure:
  - ◇ Distributing data [15],
  - ◇ Simultaneous exchange of packets between processors [2],
  - ◇ ...
- Current upperbound: $(2 - \varepsilon)$-approximation [3]

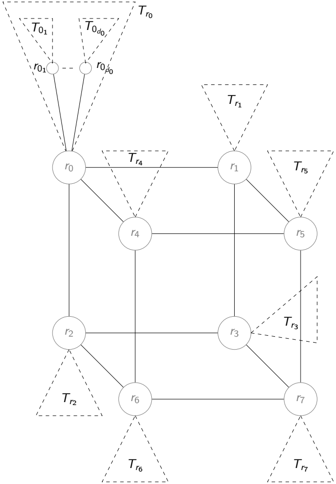


Figure: HT_3, A hypercube of trees with dimension 3

HT_k - Proposed Heuristic

- Instead of finding $b(u, HT_k)$, solve this:
    - $b(u, HT_k) \leq \tau$?

HT_k - Proposed Heuristic

- Instead of finding $b(u, HT_k)$, solve this:
  - ◇ $b(u, HT_k) \leq \tau$?
- Already know the upper bound and lower bound:
  - ◇ $\underbrace{\max\left\{k, \max_{0\leq i\leq 2^k-1}\{b(r_i, T_i)\}\right\}}_{lb} \leq b(u, HT_k) \leq \underbrace{k + \max_{0\leq i\leq 2^k-1}\{b(r_i, T_i)\}}_{ub}$

HT_k - Proposed Heuristic

- Instead of finding $b(u, HT_k)$, solve this:
  - ◇ $b(u, HT_k) \leq \tau$?
- Already know the upper bound and lower bound:
  - ◇ $\underbrace{\max\left\{k, \max_{0\leq i\leq 2^k-1}\{b(r_i, T_i)\}\right\}}_{lb} \leq b(u, HT_k) \leq \underbrace{k + \max_{0\leq i\leq 2^k-1}\{b(r_i, T_i)\}}_{ub}$
- Do a binary search on this range (MBS).
  - ◇ Invoke the main heuristic (br) within this function.

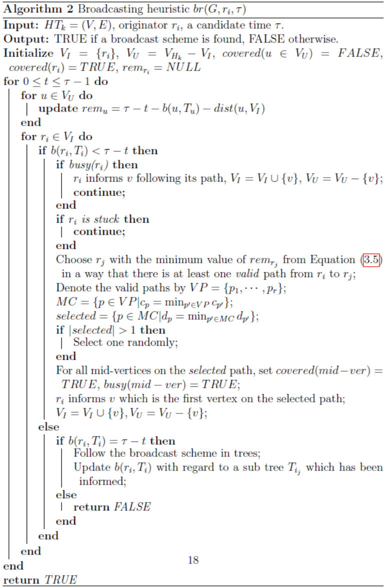
HT_k - Proposed Heuristic - MBS

Algorithm 1 The Modified Binary Search $MBS(G, r_i, lb, ub)$

```
input  : $HT_k = (V, E)$, originator $r_i$, lower bound $lb$, and the upper bound $ub$
output: An improved broadcast time for $b(r_i, HT_k)$ denoted by $b$
$b = ub$
if $ub < lb$ then
    return $FALSE$
end
$mid = lb + \lfloor \frac{ub-lb}{2} \rfloor$
if $br(HT_k, r_i, mid)$ then
    update $b = mid$
    return $MBS(G, r_i, lb, mid - 1)$
else
    return $MBS(G, r_i, mid + 1, ub)$
end
return $b$
```

HT_k - Proposed Heuristic - *br*

- Considering a root vertex r_i at time t:
    - $b(r_i, T_i) > \tau - t$:
        - return FALSE!
    - $b(r_i, T_i) = \tau - t$:
        - r_i broadcasts in its tree T_i.
    - $b(r_i, T_i) < \tau - t$:
        - r_i must contribute in the Hypercube...



Algorithm 2 Broadcasting heuristic $br(G, r_i, \tau)$

```
Input: $HT_k = (V, E)$, originator $r_i$, a candidate time $\tau$.
Output: TRUE if a broadcast scheme is found, FALSE otherwise.
Initialize $V_I = \{r_i\}$, $V_U = V_{H_k} - V_I$, $covered(u \in V_U) = FALSE$,
  $covered(r_i) = TRUE$, $rem_{r_i} = NULL$
for $0 \leq t \leq \tau - 1$ do
    for $u \in V_U$ do
        update $rem_u = \tau - t - b(u, T_u) - dist(u, V_I)$
    end
    for $r_i \in V_I$ do
        if $b(r_i, T_i) < \tau - t$ then
            if $busy(r_i)$ then
                $r_i$ informs $v$ following its path, $V_I = V_I \cup \{v\}$, $V_U = V_U - \{v\}$;
                continue;
            end
            if $r_i$ is stuck then
                continue;
            end
            Choose $r_j$ with the minimum value of $rem_{r_j}$ from Equation (3.5)
              in a way that there is at least one valid path from $r_i$ to $r_j$;
            Denote the valid paths by $VP = \{p_1, \cdots, p_r\}$;
            $MC = \{p \in VP | c_p = \min_{p' \in VP} c_{p'}\}$;
            $selected = \{p \in MC | d_p = \min_{p' \in MC} d_{p'}\}$;
            if $|selected| > 1$ then
                Select one randomly;
            end
            For all mid-vertices on the selected path, set $covered(mid-ver) =$
              $TRUE$, $busy(mid - ver) = TRUE$;
            $r_i$ informs $v$ which is the first vertex on the selected path;
            $V_I = V_I \cup \{v\}, V_U = V_U - \{v\}$;
        else
            if $b(r_i, T_i) = \tau - t$ then
                Follow the broadcast scheme in trees;
                Update $b(r_i, T_i)$ with regard to a sub tree $T_{i_j}$ which has been
                  informed;
            else
                return $FALSE$
            end
        end
    end
end
return $TRUE$
```

18

HT_k - Proposed Heuristic - br - cont.

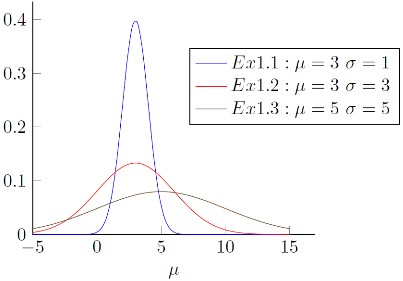
- We want to send a message from a root vertex r_i to another root vertex r_j
- Two questions must be answered:
    - How to choose r_j?
        - ◇ The one with the minimum value of rem_{r_j}:
          $rem_{r_j} = \tau - t - b(r_j, T_j) - dist(r_j, V_I)$

HT_k - Proposed Heuristic - *br* - cont.

- We want to send a message from a root vertex r_i to another root vertex r_j
- Two questions must be answered:
  - How to choose r_j?
    - ◇ The one with the minimum value of rem_{r_j}:
      $rem_{r_j} = \tau - t - b(r_j, T_j) - dist(r_j, V_I)$
  - How to choose a path between r_i and r_j ?
    - ◇ A path P with the minimum value of c_P and d_P:
      $c_P = \min_{r_m \in P}\{rem_{r_m}\}$
      $d_P = \sum_{r_m \in P} rem_{r_m}$
    - ◇ More "critical" vertices on such a path!

HT_k - Evaluation - Setup

- Generate 1000 random HT_k for each k.
  - Following Gaussian distribution $p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$



- In terms of:
  - *success rate*: How many times we performed better than [3]?
  - *gain*: How much better?
  - $|V|$

HT_k - Evaluation - Ex 1.1

Table: Numerical Results for Ex1.1: $\mu = 3, \sigma = 1$

k	average $\lvert V \rvert$	success rate	average gain
3	115.86	29.79%	7.55%
4	233.55	28.49%	5.84%
5	466.09	21.3%	3.71%
6	932.49	5.7%	0.85%
7	1862.98	1.3%	0.16%
8	3729.59	0.2%	0.02%

HT_k - Evaluation - Ex 1.2

Table: Numerical Results for Ex1.2: $\mu = 3, \sigma = 3$

k	average $\|V\|$	success rate	average gain
3	614.28	73.5%	24.45%
4	1189.54	78.6%	23.31%
5	2481.77	74.5%	19.81%
6	5008.22	69.2%	15.55%
7	10003.63	63.2%	12.5%
8	19927.99	54.7%	9.76%

HT_k - Evaluation - Ex 1.3

Table: Numerical Results for Ex1.3: $\mu = 5, \sigma = 5$

k	average $\|V\|$	success rate	average gain
3	185255.48	81.20%	28.65%
4	280823.85	86.00%	28.82%
5	704372.23	88.10%	27.35%
6	1313690.28	89.30%	26.70%
7	3532669.06	90.50%	25.85%
8	5245921.21	90.10%	24.10%

HT_k - Evaluation - Ex 2.1

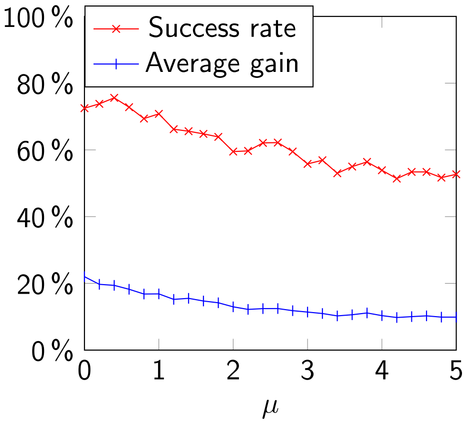


Figure: Ex2.1: $k = 5, \sigma = 2$

HT_k - Evaluation - Ex 2.2

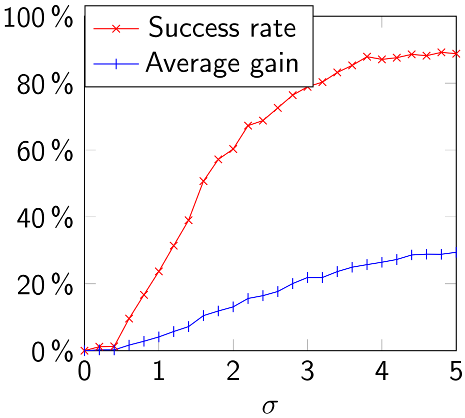


Figure: Ex2.2: $k = 5, \mu = 2$

HT_k - Conclusion

- We proposed a heuristic for broadcasting in a hypercube of trees.
- Theoretically:
    - ◇ 2-approximation
- Practically:
    - ◇ outperform the best-known algorithm for the same problem in up to 90% of the experiments while speeding up the process up to 30%

Fully Connected Trees

- A Fully Connected Tree FCT_n:
  - ◇ A Clique of size n +
  - ◇ n arbitrary trees.
- Previous result: A $O(|V|\log|V|)$ algorithm [8]

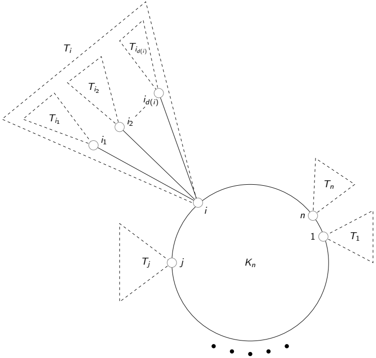


Figure: A Fully Connected Tree FCT_n

FCT_n - Broadcast Algorithm for Root Vertices

- Instead of finding $b(i, FCT_n)$, solve this:
  - ◇ $b(i, FCT_n) \leq \tau$?

FCT_n - Broadcast Algorithm for Root Vertices

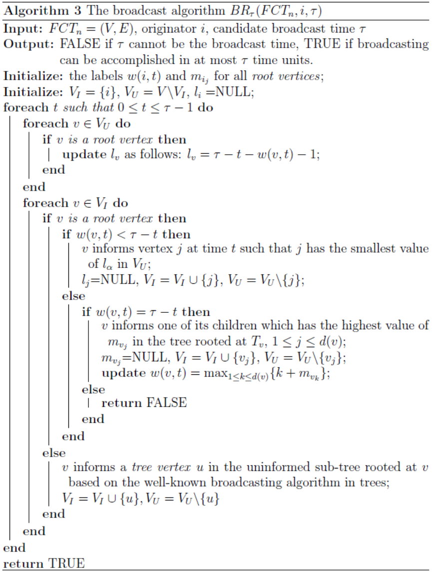
- Instead of finding $b(i, FCT_n)$, solve this:
  - ◇ $b(i, FCT_n) \leq \tau$?
- **Lemma:**
  - ◇ $\underbrace{\max\left\{\lceil \log n \rceil, \max\{b(i, T_i)\}\right\}}_{lb} \leq b(i, FCT_n) \leq \underbrace{\lceil \log n \rceil + \max\{b(i, T_i)\}}_{ub}$

FCT_n - Broadcast Algorithm for Root Vertices

- Instead of finding $b(i, FCT_n)$, solve this:
    - ◇ $b(i, FCT_n) \leq \tau$?
- **Lemma:**
    - ◇ $\underbrace{\max\left\{\lceil \log n \rceil, \max\{b(i, T_i)\}\right\}}_{lb} \leq b(i, FCT_n) \leq \underbrace{\lceil \log n \rceil + \max\{b(i, T_i)\}}_{ub}$
- Do a binary search on this range (MBS).
    - ◇ Invoke the main algorithm (BR_τ) within this function.

FCT_n - Broadcast Algorithm for Root Vertices - cont.

- Considering a root vertex i at time t:
  - $b(i, T_i) > \tau - t$:
    - return FALSE!
  - $b(i, T_i) = \tau - t$:
    - i broadcasts in its tree T_i.
  - $b(i, T_i) < \tau - t$:
    - i must contribute in the Clique...



Algorithm 3 The broadcast algorithm $BR_\tau(FCT_n, i, \tau)$

```
Input: $FCT_n = (V, E)$, originator $i$, candidate broadcast time $\tau$
Output: FALSE if $\tau$ cannot be the broadcast time, TRUE if broadcasting
        can be accomplished in at most $\tau$ time units.
Initialize: the labels $w(i, t)$ and $m_{i_j}$ for all root vertices;
Initialize: $V_I = \{i\}$, $V_U = V \backslash V_I$, $l_i$ =NULL;
foreach $t$ such that $0 \leq t \leq \tau - 1$ do
    foreach $v \in V_U$ do
        if $v$ is a root vertex then
            update $l_v$ as follows: $l_v = \tau - t - w(v, t) - 1$;
        end
    end
    foreach $v \in V_I$ do
        if $v$ is a root vertex then
            if $w(v, t) < \tau - t$ then
                $v$ informs vertex $j$ at time $t$ such that $j$ has the smallest value
                 of $l_\alpha$ in $V_U$;
                $l_j$=NULL, $V_I = V_I \cup \{j\}$, $V_U = V_U \backslash \{j\}$;
            else
                if $w(v, t) = \tau - t$ then
                    $v$ informs one of its children which has the highest value of
                     $m_{v_j}$ in the tree rooted at $T_v$, $1 \leq j \leq d(v)$;
                    $m_{v_j}$=NULL, $V_I = V_I \cup \{v_j\}$, $V_U = V_U \backslash \{v_j\}$;
                    update $w(v, t) = \max_{1 \leq k \leq d(v)} \{k + m_{v_k}\}$;
                else
                    return FALSE
                end
            end
        else
            $v$ informs a tree vertex $u$ in the uninformed sub-tree rooted at $v$
             based on the well-known broadcasting algorithm in trees;
            $V_I = V_I \cup \{u\}, V_U = V_U \backslash \{u\}$
        end
    end
end
return TRUE
```

FCT_n - Broadcast Algorithm for Root Vertices - cont.

- We want to send a message from a root vertex i to another root vertex j
- Two questions must be answered:
  - How to choose j?
    - ◇ The one with the minimum value of l_j:
      $l_j = \tau - t - w(j, t) - 1,$
      $w(i, t) = \max_{1 \leq j \leq d(i)}(j + m_{i_j}),$
      $m_{i_j} = b(i_j, T_{i_j}).$

FCT_n - Broadcast Algorithm for Root Vertices - cont.

- We want to send a message from a root vertex i to another root vertex j
- Two questions must be answered:
  - How to choose j?
    - ◇ The one with the minimum value of l_j:
      $l_j = \tau - t - w(j,t) - 1$,
      $w(i,t) = \max_{1 \leq j \leq d(i)}(j + m_{i_j})$,
      $m_{i_j} = b(i_j, T_{i_j})$.
  - How to choose a path between r_i and r_j ?
    - ◇ Easy! They are connected by an edge (i,j).

FCT_n - Proof of Correctness

Theorem 1

If Algorithm $BR_\tau(FCT_n, i, \tau)$ *outputs TRUE, then* $b(i, FCT_n) \leq \tau$.

Theorem 2

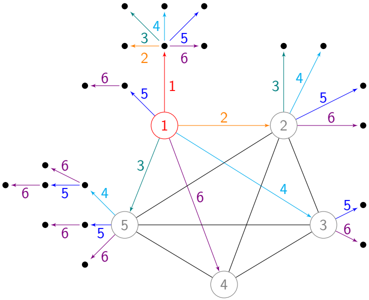
if Algorithm $BR_\tau(FCT_n, i, \tau)$ *returns FALSE, then* $b(i, FCT_n) > \tau$.

- **Lemma:** If under broadcast scheme S_τ, a root vertex i informs a tree vertex i_j at time t_1, then it is necessary for any other scheme S for broadcasting in the same graph to inform i_j by the time t' where $t' \leq t_1$.
- **Lemma:** Assume $b(i, FCT_n) = \tau$. Let S_{opt} be an optimum broadcast scheme different than S_τ. Then, at any time t, $|V_t(S_\tau)| \geq |V_t(S_{opt})|$.
    - ◇ $V_t(S)$: The set of informed root vertices by the time t under the broadcast scheme S.

FCT_n - Time Complexity

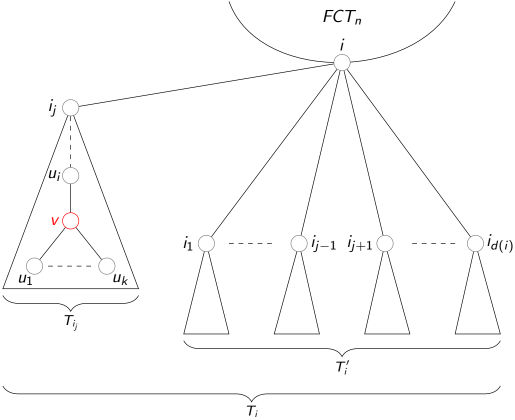
- $O(\text{Algorithm}) \times O(\log(ub - lb))$
  - ◇ The Algorithm is linear $O(|V|)$,
  - ◇ $$\begin{aligned} ub - lb &= \lceil \log n \rceil + \max\{b(i, T_i)\} - \max\left\{\lceil \log n \rceil, \max\{b(i, T_i)\}\right\} \rightarrow \\ ub - lb &= \min\left\{\lceil \log n \rceil, \max\{b(i, T_i)\}\right\} \rightarrow \\ ub - lb &\leq \lceil \log n \rceil. \end{aligned} \tag{1}$$
- Complexity: $O(|V| \log \log n)$

FCT_n - Example



Labels after $t = 1$						
root vertex i	$w(i,t)$	m_{i_1}	m_{i_2}	m_{i_3}	m_{i_4}	l_i
1	$\cancel{6}$ 2	$\cancel{5}$ -	1	-	-	-
2	4	0	0	0	0	$\cancel{1}$ 0
3	2	0	0	-	-	$\cancel{3}$ 2
4	0	-	-	-	-	$\cancel{5}$ 4
5	3	2	1	0	-	$\cancel{2}$ 1

FCT_k - Broadcast Algorithm for Tree Vertices



Theorem 3

This Algorithm generates the optimal broadcast time for a tree vertex in an FCT_n.

- Complexity = The same.

FCT_n - Conclusion

- An optimal algorithm for broadcasting in an FCT_n,
- Proof of correctness.
- Complexity: $O(|V| \log \log n)$,

Outline

1 Introduction

2 Preliminaries and Literature Review

3 Contribution
    Hypercube of Trees
    Fully Connected Trees

4 Future Works
    Development for our heuristic
    Messy broadcasting
    Broadcasting using universal lists

5 Conclusion and Timeline

Development for our heuristic

- The heuristic proposed for HT_k:
    - ◇ Extending the results for other networks: CCC_d, SE_d, BF_d, DE_d, etc.
    - ◇ Attaching random trees to the vertices of those networks,
    - ◇ Improving the heuristic itself:
        - When the heuristic fails?
          When i wants to inform j, but $dist(i, j)$ is huge.
        - How to resolve it?
          Consider a new label: $rem_{v,u}; v \in V_I, u \in V_U$.

Messy broadcasting

- Limited knowledge over the network,
    - Nodes must be able to act independently.

Messy broadcasting

- Limited knowledge over the network,
  - Nodes must be able to act independently.
- Three models:
  - M_1: A vertex knows the state of its neighbors; informed-uninformed.
  - M_2: A vertex knows from which vertices it received the message and to which vertices it has sent it.
  - M_3: A vertex only knows to which vertices it has sent the message.

Messy broadcasting

- Limited knowledge over the network,
  - ◇ Nodes must be able to act independently.
- Three models:
  - ◇ M_1: A vertex knows the state of its neighbors; informed-uninformed.
  - ◇ M_2: A vertex knows from which vertices it received the message and to which vertices it has sent it.
  - ◇ M_3: A vertex only knows to which vertices it has sent the message.
- Difference with classical model:
  - ◇ $t_i(u)$ is the maximum time units required to finish broadcasting from u, not minimum!

Messy broadcasting - cont.

- Limited results so far [10]:
    - ◇ The exact values of $t_i(G)$ for K_n, P_n, C_n, and Complete $d-$ary trees for $i = 1, 2, 3$,
    - ◇ The exact values of $t_i(G)$ for H_k for only $i = 2, 3$,
    - ◇ Upper bounds for CCC_d, SE_d, BF_d, and DE_d for $i = 1, 2, 3$.

Messy broadcasting - cont.

- Limited results so far [10]:
  - ◇ The exact values of $t_i(G)$ for K_n, P_n, C_n, and Complete $d-$ary trees for $i = 1, 2, 3$,
  - ◇ The exact values of $t_i(G)$ for H_k for only $i = 2, 3$,
  - ◇ Upper bounds for CCC_d, SE_d, BF_d, and DE_d for $i = 1, 2, 3$.
- Open problems:
  - ◇ $t_1(H_k)$,
  - ◇ Tightening the bounds for CCC_d, SE_d, BF_d, and DE_d.
  - ◇ Proposing a heuristic for general graphs.

Broadcasting using universal lists

- In classical model: Each node keeps a single list for each possible originator and need the complete knowledge over the network.

Broadcasting using universal lists

- In classical model: Each node keeps a single list for each possible originator and need the complete knowledge over the network.
- Here, a universal list is maintained for all vertices.
- Two models:
    - ◇ Adaptive: Skip the ones you received from,
    - ◇ Non-adaptive: Skip none; send to all neighbors.

Broadcasting using universal lists

- In classical model: Each node keeps a single list for each possible originator and need the complete knowledge over the network.
- Here, a universal list is maintained for all vertices.
- Two models:
    - ◇ Adaptive: Skip the ones you received from,
    - ◇ Non-adaptive: Skip none; send to all neighbors.
- We want to introduce a third model:
    - ◇ Fully adaptive: Skip all informed vertices.

Broadcasting using universal lists

- In classical model: Each node keeps a single list for each possible originator and need the complete knowledge over the network.
- Here, a universal list is maintained for all vertices.
- Two models:
  - Adaptive: Skip the ones you received from,
  - Non-adaptive: Skip none; send to all neighbors.
- We want to introduce a third model:
  - Fully adaptive: Skip all informed vertices.
- A trade-off between space complexity and the speed of broadcasting:

Model	No. of unnecessary calls	Space Complexity	Speed
Non-adaptive	Many	Very Low	Very Slow
Adaptive	Few	Low	Slow
Fully Adaptive	0	Moderate	Moderate
Classical	0	Very high	Very Fast

Broadcasting using universal lists - cont.

- Next step: Develop a comprehensive framework based on Genetic Algorithm for all three models using universal lists.

Broadcasting using universal lists - cont.

- Next step: Develop a comprehensive framework based on Genetic Algorithm for all three models using universal lists.
- A candidate solution to the problem: A matrix $\sigma_{n \times \Delta(G)}$.
- Generate several random solutions,
- Using crossover and mutation over multiple generations, a relatively good solution could be found.

Broadcasting using universal lists - cont.

- Next step: Develop a comprehensive framework based on Genetic Algorithm for all three models using universal lists.
- A candidate solution to the problem: A matrix $\sigma_{n \times \Delta(G)}$.
- Generate several random solutions,
- Using crossover and mutation over multiple generations, a relatively good solution could be found.
- Pros:
    - ◇ Works for arbitrary graphs,
    - ◇ Works for all three models,
    - ◇ Several fitness functions could be defined,
    - ◇ Efficient in terms of time complexity,
    - ◇ Gives the actual broadcast scheme,
    - ◇ The scheme could be used separately for proving many results,
    - ◇ ...

Outline

1 Introduction

2 Preliminaries and Literature Review

3 Contribution
  Hypercube of Trees
  Fully Connected Trees

4 Future Works
  Development for our heuristic
  Messy broadcasting
  Broadcasting using universal lists

5 Conclusion and Timeline

Conclusion

- Considering broadcast problem,

Conclusion

- Considering broadcast problem,
- So far:
    - ◇ A broadcasting heuristic for HT_k,
        - *Published in [6]*
    - ◇ An optimal broadcast algorithm for FCT_n,
        - *Submitted to [7]*

Conclusion

- Considering broadcast problem,
- So far:
    - A broadcasting heuristic for HT_k,
        - *Published in [6]*
    - An optimal broadcast algorithm for FCT_n,
        - *Submitted to [7]*
- In the future:
    - Development for our heuristic,
    - Working on Messy broadcasting,
    - A new model based on universal lists,
    - A comprehensive framework using GA for broadcasting under universal lists.

Timeline

2021			
Jan.	Feb. T2	Mar. O2	Apr.
May T3	Jun. O3	Jul.	Aug.
Sep. T4	Oct. O4	Nov.	Dec. T5

2022			
Jan. O5	Feb.	Mar. T6	Apr. O6
May.	Jun.	Jul. T7	Aug. O7
Sep.	Oct.	Nov.	Dec.

No.		Description	Date	Done?
T1	Task 1	Broadcasting heuristic in Hypercube of Trees	Nov. 2020	✓
O1	Outcome 1	Conference paper	Dec. 2020	✓ [24]
T2	Task 2	Optimal broadcasting in Fully Connected Trees	Feb. 2021	✓
O2	Outcome 2	Journal paper	Mar. 2021	✓ [25]
T3	Task 3	Introducing a new broadcast model for universal lists	May 2021	
O3	Outcome 3	Conference paper	Jun. 2021	
T4	Task 4	Broadcasting with universal lists with GA	Sep. 2021	
O4	Outcome 4	Journal paper + Ph.D. Seminar	Oct. 2021	
T5	Task 5	Messy broadcasting	Dec. 2021	
O5	Outcome 5	Journal paper	Jan. 2022	
T6	Task 6	Improved heuristic for Hypercube of Trees	Mar. 2022	
O6	Outcome 6	Journal paper	Apr. 2022	
T7	Task 7	Thesis writing	Jul. 2022	
O7	Outcome 7	Ph.D. defense	Aug. 2022	

Important References I

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

[2] Dimitri P Bertsekas, C Özveren, George D Stamoulis, Paul Tseng, and John N. Tsitsiklis.
Optimal communication algorithms for hypercubes.
Journal of Parallel and Distributed Computing, 11(4):263–275, 1991.

[3] Puspal Bhabak and Hovhannes A Harutyunyan.
Broadcast problem in hypercube of trees.
In *International Workshop on Frontiers in Algorithmics (FAW)*, pages 1–12. Springer, 2014.

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

Important References II

[5] Michael Elkin and Guy Kortsarz.
Sublogarithmic approximation for telephone multicast.
Journal of Computer and System Sciences, 72(4):648–659, 2006.

[6] Saber Gholami and H. A. Harutyunyan.
A broadcasting heuristic for hypercube of trees.
In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 355–361, 2021.

[7] Saber Gholami and Hovhannes A Harutyunyan.
Optimal broadcasting in fully connected trees.
Submitted to Networks, 2021.

[8] Hovhannes Harutyunyan and Edward Maraachlian.
Broadcasting in fully connected trees.
In *2009 15th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 740–745. IEEE, 2009.

Important References III

[9] Hovhannes A. Harutyunyan, George Laza, and Edward Maraachlian.
Broadcasting in necklace graphs.
In *Canadian Conference on Computer Science & Software Engineering (C3S2E)*, pages 253–256. ACM, 2009.

[10] Hovhannes A Harutyunyan and Arthur L Liestman.
Messy broadcasting.
Parallel Processing Letters, 8(02):149–159, 1998.

[11] Hovhannes A Harutyunyan and Edward Maraachlian.
On broadcasting in unicyclic graphs.
Journal of combinatorial optimization, 16(3):307–322, 2008.

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

Important References IV

[13] Hovhannes A Harutyunyan and Wei Wang.

Broadcasting algorithm via shortest paths.

In *16th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 299–305. IEEE, 2010.

[14] Hovhannes A Harutyunyan and Wei Wang.

A random heuristic for message broadcasting in arbitrary networks.

In *Proceedings of the 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pages 341–345, 2010.

[15] Ching-Tien Ho and S Lennart Johnsson.

Distributed routing algorithms for broadcasting and personalized communication in hypercubes.

In *International Conference on Parallel Processing (ICPP)*, pages 640–648, 1986.

Important References V

[16] Cory J Hoelting, Dale A Schoenefeld, and Roger L Wainwright.
A genetic algorithm for the minimum broadcast time problem using a global precedence vector.
In *Proceedings of the 1996 ACM Symposium on Applied Computing (SAC)*, pages 258–262, 1996.

[17] Andreas Jakoby, Rüdiger Reischuk, and Christian Schindelhauer.
The complexity of broadcasting in planar and decomposable graphs.
Discrete Applied Mathematics, 83(1-3):179–206, 1998.

[18] Guy Kortsarz and David Peleg.
Approximation algorithms for minimum-time broadcast.
SIAM Journal on Discrete Mathematics, 8(3):401–427, 1995.

[19] Edward Maraachlian.
Optimal broadcasting in treelike graphs.
PhD thesis, Concordia University, 2010.

Important References VI

[20] R Ravi.

Rapid rumor ramification: Approximating the minimum broadcast time.

In *Proceedings 35th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 202–213. IEEE, 1994.

[21] Christian Schindelhauer.

On the inapproximability of broadcasting time.

In *International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 226–237. Springer, 2000.

[22] Peter J. Slater, Ernest J. Cockayne, and Stephen T. Hedetniemi.

Information dissemination in trees.

SIAM Journal on Computing, 10(4):692–701, 1981.

Thanks a bunch!