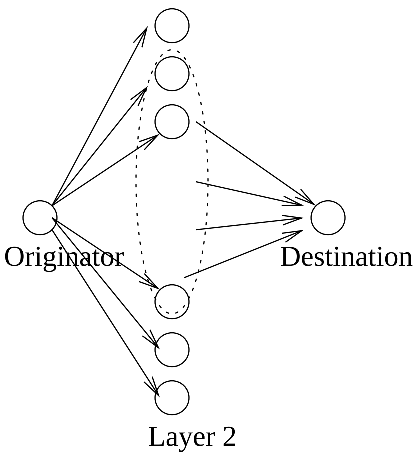
Randomized Lower Bound for Sending Message in $\mathcal{F}_n$ Graph

Nov 7, 2002

1 Introduction

1.1 Preliminaries

Definition 1 ($\mathcal{F}_n$ Graph) *Directed graph $\mathcal{F}_n = (V, E)$ consists of $n+2$ nodes located in 3 layers. One node called* originator*, one node called* destination *locate in 1^{st} and 3^{rd} layer respectively. The other n points labeled from 1 to n, locate in the 2^{nd} layer. There are n edges from originator to each point in layer 2 and some edges from layer 2 to destination.*



Remark 1 *The nodes of the graph represent processors and an edge from u to v implies that u can send message to v.*

Radio Network Property One node can receive message if and only if exactly one of the node's neighbors transmits at time t .

Sending Message Send message from originator to destination.

Time Measurement The time is viewed as divided into slots (or *rounds*). In any given slot, a node can either transmit some message or not.

1.2 Algorithm for Sending Messages

Obviously, at first the originator sends its message to all the n nodes in the second layer(*initial transmission*).

Definition 2 (Deterministic Algorithm) *For any deterministic algorithm Π, after the initial transmission the algorithm runs in m steps according to a family of set $\{S_1,\ S_2,\ \cdots\ S_m\}$, where each set S_i is a subset of $U = \{1,\ 2,\ \cdots n\}$. In step i, the nodes whose labels are in S_i will send message while other keep silence. So m is the running time of this algorithm.*

Definition 3 (Randomized Algorithm) *In the randomized case, the probability rises in choosing the node to send message:*

- *Each node P_i uses a probabilistic procedure Φ_i^j to decide whether the node will transmit at the next time slot j or not, which may depend on the* history
- *The program of P_i is a collection $\Phi_i = (\Phi_i^1, \Phi_i^2, \Phi_i^3, ...)$*
- *A randomized algorithm Π for the task in F_n is simply a collection of n such program and an initial transmission.*
- *Actually we can also get a family of set $\{S_1, S_2, \cdots, S_m\}$ under the universal set U, which is decided by every program of P_i. In step j, the nodes whose labels are in S_j will send message.*
- *After m step, the task complete with bounded error.*

Remark 2 *: An algorithm is* uniform *if all processors use the same program. Otherwise, the algorithm is called* nonuniform.

Success Transmission In some step i of Π, $|S_i \cap X| = 1$

1.3 Bounds for Sending Message in $\mathcal{F}_n$ Graph

In the deterministic case, we can easily get an algorithm running in time $O(n)$. And by the adversary argument, the lower bound is $\Omega(n)$. So the deterministic algorithm's complexity is $\Theta(n)$.

Does the ability to flip a coin add computational power in this task? The answer is positive. The following randomized algorithm runs in $O(logn)$.[4]

each node in layer 2 runs the procedure send(n)

procedure send(n)

for i:=1 to $logn + 1$ do

with probability $\frac{1}{2^{i-1}}$ send

In 1998, Kushilevitz and Mansour gave a 0-sided randomized lower bound in broadcasting in radio network [1]. The main ideas to prove that bound can exactly be used to prove the lower bound in the next section. In addition, a simple proof is given in section 3.

2 Kushilevitz and Mansour's Method

2.1 Uniform Case

For an algorithm Π, we call a round *successful* if exactly one processor which is connected to destination transmits. Let $E(T_\ell^\Pi)$ denote the expected number of rounds until the *first* successful round, given that there are 2^ℓ edges incident to destination.

Theorem 2.1 *: Let Π be an algorithm in graph $\mathcal{F}_n$ to transmit message from originator to destination. Then*

$$E_\ell[E(T_\ell^\Pi)] = \Omega(log(n)),$$

where E_ℓ denotes the expectation when ℓ is chosen uniformly from the range $1 \leq \ell \leq logn$

Ideas:

- The theorem only deals with the *first* success. This, in a sense, allows us to get rid of the dependency in the history.
- Intuitively the algorithm cannot have high probability of success in (a fixed) round s for more than a few values of 2^ℓ.This would imply that since topology between layer 2 and layer 3 is unknown, $\Omega(logn)$ rounds would be required to reach a success for all possible cases.

Proof

Define $p_{s,\ell} = Pr[$ failure in rounds 1,...$s-1$ and success in round s | there is 2^ℓ edges incident to destination]

As the events described in the definition are disjoint and assuming that the algorithm succeeds with probability 1, we get

$$\sum_{s=1}^{\infty} p_{s,\ell} = 1$$

We can get rid of the dependency simply by writing:

$$p_{s,\ell} \leq Pr[success\ \ in\ \ round\ \ s|2^\ell\ edges\ \ to\ \ destination]$$

Formally, we make the following claim to illustrate the second idea.

Claim 2.2 *. For any s,*

$$\sum_{\ell=1}^{logn} Pr[success\ \ in\ \ round\ \ s\ \ |2^\ell edges\ \ to\ \ destination] < 2$$

Proof:

Define $q(s) = Pr[trying\ \ in\ \ round\ \ s]$.Note that $q(s)$ does not depend on ℓ. Recall that a successful round is the one in which exactly one node is trying to transmit. Therefore

$$\text{Pr[success in rounds } s\ |2^\ell] = 2^\ell \cdot q(s) \cdot (1-q(s))^{2^\ell - 1}$$

We get

$$\sum_{\ell=1}^{logn} Pr[success\ \ in\ \ round\ \ s|2^{\ell}] = \sum_{\ell=1}^{logn} 2^{\ell} q(s)(1-q(s))^{2^{\ell}-1}$$

$$= q(s) \sum_{\ell=1}^{logn} 2^{\ell}(1-q(s))^{2^{\ell}-1} \leq 2 \cdot q(s) \sum_{j=1}^{n-1} (1-q(s))^{j}$$

$$= 2 \cdot q(s) \cdot \frac{1-(1-q(s))^{n}}{q(s)} < 2$$

which completes the proof of the claim. *Q.E.D*

We are interested in $\sum_{s=1}^{k} p_{s,\ell}$, which is intuitively the probability that, given that there are 2^{ℓ} edges incident to destination, the algorithm succeeds in one of the first k rounds.

$$\sum_{\ell=1}^{logn} \sum_{s=1}^{k} p_{s,\ell}$$

$$\leq \sum_{s=1}^{k} \sum_{\ell=1}^{logn} Pr[success\ in\ round\ s\ |2^{\ell}] < 2k$$

By definition,

$$E_{\ell}[E(T_{\ell}^{\Pi})] = \sum_{\ell=1}^{logn} \frac{1}{logn} \sum_{s=1}^{\infty} p_{s,\ell} \cdot s \geq \frac{k}{logn} \sum_{\ell=1}^{logn} \sum_{s=k}^{\infty} p_{s,\ell}$$

this equals

$$\frac{k}{logn} \sum_{\ell=1}^{logn} \left(1 - \sum_{s=1}^{k-1} p_{s,\ell}\right)$$

which is greater than

$$\frac{k}{logn} \cdot (logn - 2(2(k-1)).$$

By choosing $k = \frac{1}{4}logn$ we have that

$$E_{\ell}[E(T_{\ell}^{\Pi})] \geq \frac{1}{8}logn + \frac{1}{2}$$

which completes the proof of the lemma.

2.2 Nonuniform Case

We can reduce the nonuniform case into uniform case referring to[1].

2.3 Randomized Lower Bound

From the argument above we get following:

Theorem 2.3 *The randomized lower bound for transmit message from originator to destination in graph $\mathcal{F}_n$ is $\Omega(log(n))$. [1]*

Lemma 2.4 *: If L is accepted by a probabilistic algorithm M with 0-sided error and expected time $T(n)$, then L is accepted by a probabilistic algorithm M' with 1-sided error and worst case time $O(T(n))$.*

The lower bound also holds in *Monte Carlo* case.

3 A Simple Proof for Lower Bound

3.1 Yao's Lemma

Lemma 3.1 *The randomized complexity $R(f)$, if both the set of inputs and the set of algorithm is finite, is equal to the maximum over all probability distributions on inputs of the minimum expected cost of any deterministic algorithm.[3]*

Remark 3 *: So in order to find the randomized lower bound in sending message in $\mathcal{F}_n$ graph, it suffices to find a lower bound on such expected cost of any deterministic algorithm. Such a lower bound can be found by postulating an input distribution $\mathcal{P}$ and see whether every algorithm has expected cost at least equal to the desired lower bound.[2]*

3.2 Input Distribution

Since we do not know which nodes connect to the destination, we use $X \subseteq U$ to denote the edges between layer 2 and destination (*Note that* $X \neq \phi$).

Definition 4 (Input distribution $\mathcal{P}$) *The probability of number of the edges between layer 2 and destination is defined as*

$$p_k = Pr[|X| = k]$$

where

$$0 \leq p_k \leq 1 \ \ and \ \ \sum_{k=1}^{n} p_k = 1$$

3.3 Reasoning Clue

If we can proof that under some distribution $\mathcal{P}$, for each deterministic algorithm Π, its running time m for success sending message is lower bounded by $\Omega(logn)$. Then by the description above, we will conclude that the randomized lower bound for this task is $\Omega(logn)$.

In [5] ,where the author gave an amortized progress is $\Omega(logn)$ then the running time of broadcasting is $O(nlogn)$.

So in this case, we only need to prove that in any step the probability of success transmission is $O(\frac{1}{logn})$.Then the deterministic algorithm should take $\Omega(logn)$ round to complete the mission with the probability of error $1-(1-O(\frac{1}{logn})^{\Omega(logn)} = 1-\Omega(1)$. The same idea appeared in the recent paper[6].

3.4 Analysis:

Suppose we have an input distribution $\mathcal{P}$, the probability $|X|=k$ i.e.the probability when there are k edges between layer 2 and destination, is p_k.

Without lose of generality, we take the i^{th} step of a deterministic algorithm $\Pi = S_1, S_2, \cdots, S_m$.If $|X|=k$, then the probability of success transmission is

$$p_k \cdot \frac{\ell \cdot \binom{n-\ell}{k-1}}{\binom{n}{k}}$$

where $\ell = |S_i|$ is the number of nodes which send message in round i. So the expectation is

$$\sum_{k=1}^{\zeta} p_k \cdot \frac{\ell \cdot \binom{n-\ell}{k-1}}{\binom{n}{k}}$$

where $\zeta = n-\ell+1$ because of $n-\ell \geq k-1$

So the remaining thing is to find a specific distribution s.t.

$$\sum_{k=1}^{n-\ell+1} p_k \cdot \frac{\ell \cdot \binom{n-\ell}{k-1}}{\binom{n}{k}} = O(\frac{1}{logn})$$

I have tried several distribution. Naturally suppose

$$p_k = \frac{\binom{n}{k}}{\sum_{i=1}^{n}\binom{n}{i}}$$

or Uniform distribution

$$p_k = \frac{1}{n}$$

They both failed to satisfy the condition.

3.5 A Nontrivial Distribution

Since we need find a distribution $\mathcal{P}$ to satisfy

$$\sum_{k=1}^{n-\ell+1} p_k \cdot \frac{\ell \cdot \binom{n-\ell}{k-1}}{\binom{n}{k}} = O(\frac{1}{logn})$$

and the right $\frac{1}{logn}$ is quite hard to be eliminated by summation. So I guess

$$p_k = \frac{1}{logn} \cdot q_k$$

where $\sum_{k=1}^{n} q_k = logn$. Recall that

$$\sum_{i=1}^{\infty} \frac{1}{i} \longrightarrow logn$$

So Let $q_k = \frac{\lambda_n}{k}$ where λ_n is a constant defined later(*refer to the Appendix*),what remain we should do is to prove

$$\sum_{k=1}^{n-\ell+1} \frac{\lambda_n}{k} \cdot \frac{\ell \cdot \binom{n-\ell}{k-1}}{\binom{n}{k}} = O(1)$$

that is equivalent to prove

$$\sum_{k=1}^{n-\ell+1} \frac{1}{k} \cdot \frac{\binom{n-\ell}{k-1}}{\binom{n}{k}} = O(\frac{1}{\ell})$$

And let

$$\Phi(k) = \frac{1}{k} \cdot \frac{\binom{n-\ell}{k-1}}{\binom{n}{k}} \quad obviously \quad \Phi(k) > 0$$

Easily for $j \geq 1$ and$k \geq 1$.

$$\frac{\Phi(k+1)}{\Phi(k)} = \frac{n-\ell-k+1}{n-k} \leq \frac{n-\ell+1}{n} \leq 1$$

So, we can get

$$\Phi(k) \leq \Phi(1) \cdot h^{k-1}$$

where $h = \frac{n-\ell+1}{n} \leq 1$.And

$$\sum_{k=1}^{n-\ell+1} \Phi(k) \leq \sum_{k=1}^{n-\ell+1} \Phi(1) \cdot h^{k-1} \leq \sum_{k=1}^{\infty} \Phi(1) \cdot h^{k-1}$$

where $h = \frac{n-\ell+1}{n} \leq 1$. Because $\Phi(1) = \frac{1}{n}$

$$\sum_{k=1}^{\infty} \Phi(1) \cdot h^{k-1} = \frac{\Phi(1)}{1-h} = \frac{1}{\ell - 1}$$

and

$$\frac{1}{\ell - 1} \leq 3 \cdot \frac{1}{\ell} \in O(\frac{1}{\ell})$$

Q.E.D.

Lemma 3.2 *Over the probability distribution $\mathcal{P}$ above, with the probability of error $1 - \Omega(1)$ the expected running time of any deterministic algorithm for sending message in $\mathcal{F}_n$ graph is $\Omega(logn)$*

By Yao's lemma mentioned above, we get

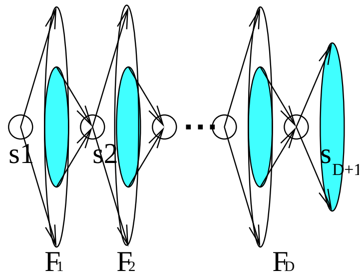
Theorem 3.3 *Each Monte Carlo algorithm in sending message in $\mathcal{F}_n$ graph need expected running time $\Omega(logn)$*

4 Randomized Lower Bound in Broadcast

Now we try to give a lower bound in broadcast algorithms. In broadcast algorithm, we suppose the algorithm knows the N, which is the number of nodes in the radio networks and D which is the *diameter*[1] of the network. we want to show that for every broadcast algorithm that does not know the topology of the network, for every N and D, there exist networks of N nodes and diameter D such that the expected running time is $\Omega(Dlog(N/D))$.

Given N and D, let $n = N/D$, we can construct a network as following:

- It has $2D + 2$ layers
- The $2i^{th}$ layer ($\forall i = 1 \cdots D$) consists of $\ell_i \leq n$ nodes, where ℓ_i is chosen uniformly in the range $1 \cdots n$.
- The $2i - 1^{th}$ layer ($\forall i = 1 \cdots D + 1$) has one node. The first one is the originator of the broadcast. Each node connects all the nodes in the next layer, and each node except originator connects the former layer by some unknown edges .
- The last layer contains all the other nodes ensuring the total number of nodes is N.



We can see that in the i^{th} cell composed of $i^{th}, i + 1^{st}, i + 2^{rd}$ layers ($\forall i = 1, 3, 5, \cdots 2D - 1$) is an $\mathcal{F}_{\ell_i}$ graph. So intuitively, on each cell the broadcast algorithm is expected to run $\Omega(log(N/D))$ round to send message from the node in i^{th} layer to the node in the $i+2^{rd}$ layer. In next round, the message should progressly pass through in the next $\mathcal{F}_{\ell_{i+1}}$ like graph. So the broadcast progresses in a layer-by-layer manner.

But we should note that before the message reaches the i^{th} cell, the nodes may have run some other algorithms and know the topology of the layers. These extra information may help the broadcasting fast. To make the argument simpler, we make the assumption that all the nodes keep silent before they get the first message.

Let t_i be the random variable denoting the time required to send message in i^{th} cell. We need to show that for some choice of $\ell_1, \cdots \ell_D$ we get $E_\Pi(\sum_{i=1}^{D} t_i) = \Omega(Dlog(N/D))$, where the expectation is taken over the random choices of the algorithm Π.

It suffices to show that$E_{\ell_1,\cdots,\ell_D,\Pi}(\sum_{i=1}^{D} t_i) = \Omega(Dlog(N/D))$. and

$$E_{\ell_1,\cdots,\ell_D,\Pi}\left(\sum_{i=1}^{D} t_i\right) = \sum_{i=1}^{D} E_{\ell_1,\cdots,\ell_D,\Pi}(t_i)$$

we just need to bound$E_{\ell_1,\cdots,\ell_D,\Pi}(t_i)$.

$$E_{\ell_1,\cdots,\ell_D,\Pi}(t_i) = \sum_{\ell_1,\cdots,\ell_D} E_{\ell_i,\Pi}(t_i|\ell_1,\cdots\ell_D) \cdot Pr[\ell_1,\cdots,\ell_D]$$

By the **Theorem 3.3**

$$E_{\ell_i,\Pi}(t_i|\ell_1,\cdots\ell_D) \geq clogn$$

for some constant c. Therefore

$$E_{\ell_1,\cdots,\ell_D,\Pi}(t_i) \geq clogn$$

[1] The diameter of a graph is defined by the longest length between any two nodes

which implies

$$E_{\ell_1,\cdots,\ell_D,\Pi}\left(\sum_{i=1}^{D} t_i\right) = \Omega(Dlogn) = \Omega(Dlog(N/D)),$$

which completes the proof.

Theorem 4.1 *Each Monte Carlo algorithm for broadcast in radio network need expected running time* $\Omega(Dlog((N/D))$

References

[1] Kushilevitz,E..Mansour,Y. *An* $\Omega(Dlog(N/D))$ *lower bound for broadcasting in radio networks*,SIAM J.Comput.27(1998),702-712

[2] Sanjeev Arora. Lecture Notes, *Computational Complexity*,2001 Fall

[3] Saks,M.. Wigderson,A *Probabilistic Boolean Decision Trees and the Complexity of Evaluating Game trees* FOCS 1986

[4] Bar-Yehu,R..Goldreich,O..Itai,A *On the Time-Complexity of Broadcast in Multi-Hop Radio Networks: An Exponential Gap between Determinism and Randomization* Distributed Computing 5 (1991)

[5] Chrobak,M.,Gąsieniec,L.,Rytter,W. *Fast Broadcasting and Gossiping in Radio Networks* FOCS 2000

[6] Liu,D.,Prabhakaran,M. *On Randomized Broadcasting and Gossiping in Radio Networks* COCOON 2002

A Appendix: About the Distribution

Because

$$\sum_{k=1}^{n} p_k = 1 \ \ and \ \ 0 \leq p_k \leq 1$$

Suppose

$$p_k = \frac{1}{logn} \cdot q_k$$

where

$$\sum_{k=1}^{n} q_k = logn \ \ and \ \ 0 \leq q_k \leq logn$$

Recall that the Harmonic series $H_n \to ln(n)$. So let $q_k = \lambda_n \cdot \frac{1}{k}$,where

$$\lambda_n \cdot \sum_{i=1}^{n} \frac{1}{i} = logn$$

$$\lambda_n = \frac{logn}{\sum_{i=1}^{n} \frac{1}{i}}$$

They should satisfy that

$$0 \leq \lambda_n \cdot \frac{1}{k} \leq logn$$

which means

$$0 \leq \frac{logn}{k \cdot \sum_{i=1}^{n} \frac{1}{i}} \leq logn$$

trivially

$$\frac{1}{k} \leq \sum_{i=1}^{n} \frac{1}{i}$$

So, our distribution is right and when $n \to \infty$ then $\lambda_n \to loge$.