CS294-1 On-line Computation & Network Algorithms **Spring 1997**

Lecture 15: March 18

Lecturer: Yair Bartal *Scribe: Zhendong Su*

In this lecture we talk about competitive analysis of distributed algorithms. In particular we study some useful tools in the constrcution of local algorithms including the mobile user problem and the data tracking problem [AP91, BFR92, Bartal96]. We also discuss Sparse Partition [AP90] and how to apply it for Distributed Tracking. These machanisms have uses, for example, for Distributed Paging problems.

One way to deal with competitive analysis of distributed algorithms is to compare their total cost including all communication costs to the cost of a global control algorithm while taking into account inherent costs of any distributed algorithm [BFR92]. A stronger definition does this directly [AADW94].

15.1 The Mobile User and Data Tracking Problems

The Mobile User Tracking problem is about distributively tracking a single user moving in a network of processors. The problem can be generalized to the Data Tracking problem, where there are multiple users and the set of users can change over time.

In a network over a set P of n processors, the data tracking problem maintains a set of data items. It must support the following operations:

- insert(u, v), copy u into v and v is inserted into the set of data items.
- delete(u), removes u from the set of data items.
- find(w), find where w is, where w is in the set of data items.

Note, we could also define insert more generally as:

- insert(v), could be implemented by a find and an insert operations. The find is used to see who has data, and the insert is used to insert from what found to v.

We define the cost for each operation as:

- $cost(insert(u, v)) = D \cdot d(u, v)$.
- $cost(delete(u)) = 0$.
- $cost(find(w)) = d(Q, w)$, where Q is the set of data items, and $d(Q, w) = \min_{q \in Q} d(q, w)$.

The data tracking problem is related to the mobile user problem since inserting and them deleting the same element is the same as moving a user.

Definition 15.1 *A distributed on-line data tracking algorithm serves sequences of insert, delete and find operations. The cost of the algorithm is the total cost of messages it sends to conduct those operations.*

Awerbuch and Peleg [AP91] prove the following:

Theorem 15.2 *There exists an $O(\log n \log(\max\{n, \Delta\}))$ competitive ratio for the mobile user tracking problem (and the more general* data tracking*)*

where Δ is the diameter of the network. In [BFR92, Bartal96] this is improved to $O(\log n \log(\min\{n, \Delta\}))$.

15.2 Graph Decompositions (Sparse Partition)

We have the following definitions:

Definition 15.3 *The following definitions are related to Graph Decomposition:*

- *A* <u>*cluster* C</u> *is a subset of vertices.*
- *A* <u>*partition*</u> *is a set of clusters (possibly overlapping). We have $\sigma = \lceil\log(\Delta)\rceil$ levels of partitions, which are denoted by $P = \{P_0, P_1, \ldots, P_\sigma\}$. ($diam(C) = \max_{u,v\in C} d(u,v)$)*
- *For P_i, we say it has a* <u>*stretch*</u> *S if $\forall C \subset P_i, dial(C) \leq (2^i - 1)S$.*
- *We define* <u>$stretch(P)$</u> $= \max_i stretch(P_i)$.
- <u>$deg(P_i)$</u> $= \max_v\{$*number of clusters* $C \in P_i, s.t., v \in C\}$.
- <u>$deg(P)$</u> $= \max_i deg(P_i)$.

What we are looking for is a partition with a small stretch and small degree.

Theorem 15.4 *There exists a partition P such that: For all i and parameter k ($k \approx O(\log n)$),*

- *Every $2^i - 1$ neighborhood is properly contained in some $C \in P_i$.*
- $deg(P_i) \leq 2n^{1/k} \cdot k \leq O(\log n)$.
- $stretch(P_i) \leq 2k - 1 < O(\log n)$. *(If $k = \log n$, then $O(\log n) = 2\log n$.)*

15.2.1 Application to Mobile User Tracking

We now prove Theorem 15.2. We maintain a hierarchical graph decomposition. Let v be the location of the mobile user. We call a level $i+1$ cluster "good" if it contains the $2(2^i-1)$ neighborhood of v. At every time there must be a level $i+1$ cluster that contains the $2^{i+1}-1$ neighborhood of v and therefore also at least one good cluster.

Once at the beginning, we compute for every level some good cluster (the cost for this is an additive term). This information is kept at the cluster's leader, as well as a pointer to the location of the user.

Let C denote the accumulative distance the user has moved over the entire process. The user maintains this quanantity modulu the diameter Δ as it moves. It also leaves pointer in every node it leaves to which neighbor node it moved.

For every $i < log(\Delta)$, if $C \text{mod}(2^i-1) = 0$ we inform the $i+1$ level cluster leader of some good cluster to mark itself as good and sends it a pointer to its current location. Then it lets the former good cluster unmark itself as good. The cost for this is $O(2^i \cdot stretch(P)) = O(2^i \log n)$. Therefore the total cost incured in this process is at most $O(C \log n)$ per level and thus $O(C \log n \log \Delta)$ over all level.

Now when a find occurs in some node u we iterate over levels $j = 0, 1, \ldots$. For each level j, we communicate with all leaders of clusters containing u to check if they are marked as good. When it finds such a cluster it follows the pointer is has of the location of the user at the time that it was marked and then follows the pointers that the user left at the individual nodes until it reaches the final location of the user.

Let i be minimal such that $d(u,v) \leq 2^i - 1$. Every good level $i+1$ cluster contains both u and v, and thus the process above will stop at level $j \leq i+1$. The cost incurred for stage j in the process is bounded by $O(2^j stretch(P) deg(P)) = O(2^j \log^2 n)$. Notice that a user is always within distance $O(2^i \log n)$ from the leader of the good level $i+1$ cluster and therefore the cost for following the pointer until the user is found is bounded by $O(2^i \log n)$. This give a total cost of $O(d(u,v)\log^2 n)$ for the find operation. This together with the bound on the update cost completes the proof.

References

[AADW94] M. Ajtai, J. Aspnes, C. Dwork, and O. Waarts. A Theory of Competitive Analysis for Distributed Algorithms. In *Proc. of the 35th Ann. IEEE Symp. on Foundations of Computer Science*, pages 401-411, October 1994.

[AP91] B. Awerbuch and D. Peleg. Concurrent On-line Tracking of Mobile Users, *Proc. SIGCOMM*. Zurich, Sept. 1991.

[AP90] B. Awerbuch and D. Peleg. Sparse Partitions. In *Proc. of the 31st Ann. Symp. on Foundations of Computer science*, pages 503-513, October 1990.

[Bartal96] Y. Bartal. Probabilistic Approximation of Metric Spaces and its Algorithmic Applications. To appear in *Proc. of the 37rd Ann. IEEE Symp. on Foundations of Computer Science*, 1996.

[BFR92] Y. Bartal, A. Fiat, and Y. Rabani. Competitive algorithms for distributed data management. In *Proc. 24th ACM Symp. on Theory of Computing*, pages 39–50, 1992. Also accepted for publication in the STOC 92 special issue *Journal of Computer and Systems Sciences*, 1993.