Distributed Algorithms as Combinatorial Structures

Keren Censor-Hillel, Technion
ADGA 2014
Today I want to inspect distributed algorithms
Not their code
Their execution
Design algorithms for \textit{synchronous} systems but run them in \textit{asynchronous} systems [Awerbuch 1985]

- Send \((r+1)\)-message after receiving all \(r\)-messages

\(v\) will wait forever if \(u\) never sends an \(r\)-message

Sending an empty message increases message complexity
Send ACKs:
- $v$ informs about receiving all ACKs ($v$ is safe)
- $v$ sends $(r+1)$-message when all neighbors are safe

Still need a message from every neighbor

Message overhead is $M = O(|E|)$
Time overhead is $T = O(1)$
In a \textbf{k-spanner} $S$ of $G$:
\[ d_S(u,v) \leq k \cdot d_G(u,v) \]
$k$ is the \textbf{stretch}

[Peleg, Ullman 1989]
**Synchronizers with Spanners**

$S$ is a $k$-spanner with $m$ edges

- Repeat $k$ iterations:
  - Send **safe** messages in the spanner
  - Wait for **safe** messages in the spanner

**2-spanner**

Iteration $t$: nodes within spanner distance $t$ are safe
All neighbors are within distance $k$ in the spanner
Synchronizers with Spanners

$k$-spanner with $m$ edges $\rightarrow$
synchronizer with $M=O(km)$, $T=O(k)$

[Peleg, Ullman 1989]
Sync is a synchronizer

Mark all edges used

Information has to pass between each pair of neighbors

Gives a spanner $S$ with $m \leq M$ and $k \leq T$
Synchronizers vs. Spanners

- A $k$-spanner with $m$ edges is equivalent to a synchronizer with $M = O(km)$, $T = O(k)$.

- A synchronizer with parameters $M$ and $T$ is equivalent to a spanner with $m \leq M$ and $k \leq T$.

[Peleg, Ullman 1989]
Synchronizers vs. Spanners

Synchronizers

Synchronizer Code for node v
1. Upon receiving:
2. Send ACK to u
3. ...

Spanners
Algorithm for PROBLEM
Code for node v
1. In round r do:
2. Send MSG to u
3. ...

Distributed Algorithms

Graph Structures
Multi-Message Broadcast
Multi-Message Broadcast
The GOSSIP model

LOCAL round: contact all neighbors
[Linial 1987]

GOSSIP round: contact a single neighbor
[Demers et al. 1987]
Multi-Message Broadcast in GOSSIP

- The challenge: bottlenecks
  - Simple round-robin performs poorly
  - Random choices depend on conductance: $\Theta(\log n/\varphi)$
    [Giakkoupis 2011, Chierichetti, Lattanzi, Panconesi 2010]
Neighbor Exchange

- T rounds for neighbor exchange → DT rounds for multi-message broadcast

- T-spanner with O(nT) edges and max degree T in T rounds
Neighbor exchange in $T$ rounds $\Rightarrow$ spanner with $m=\eta T$, $k=T$ and max degree $T$

- Neighbor exchange in $O(\log^3 n)$ rounds
  [Censor-Hillel, Hauepler, Kelner, Maymounkov 2012]

- Neighbor exchange improved to $O(\log^2 n)$
  [Hauepler 2013]
**Multi-Message Broadcast with Spanners**

$S$ is a $k$-spanner with max degree $d$
- Repeat $k$ iterations:
  - Pick new neighbor in $S$ (round-robin)

**Iteration $t$:** reach nodes within spanner distance $t$

All neighbors are within distance $k$ in the spanner

- Neighbor exchange in $dk$ rounds
- mm-broadcast in $dkD$ rounds
Not every graph has a small degree spanner. Instead, use a spanner that is “sparse enough”

- Takes $O(1)$ rounds to reach the next hop

Hereditary density $\delta$:
The maximal density of induced subgraphs

- Any induced subgraph has at most $\delta|S|$ edges

Can direct edges of $G$ with hereditary density $\delta$ such that max out-degree is $\delta$, in $O(\delta \log n)$ rounds
In GOSSIP model:

- Guess $\delta$, if have fewer than $\delta$ remaining edges then contact these neighbors, else double $\delta$
  - Terminates when the guess is $O(\delta)$

- All edges directed, out-degree at most $O(\delta)$
- A constant fraction of nodes in each iteration, otherwise density would be greater than $\delta$
- Takes $O(\delta \log(n))$ rounds to direct all edges
Multi-Message Broadcast vs. Spanners

\((\alpha, \beta)\)-spanner \(S\) with hereditary density \(\delta \rightarrow\)

\text{mm-broadcast in } T = \text{polylog} n T_S + \delta \text{log} n + \delta(\alpha D + \beta)

[Censor-Hillel, Hauepler, Kelner, Maymounkov 2012]

\text{mm-broadcast in } \mathcal{O}(D + \text{polylog } n) \text{ rounds}

- By simulating a \((\mathcal{O}(1), \text{polylog } n)\)-spanner with \(\delta = \mathcal{O}(1)\)
  construction in \(\mathcal{O}(\text{polylog} n)\) rounds [Pettie 2009]

- Other \(\alpha, \beta\) trade-offs by simulating other spanner constructions
  [Dubhashi, Mei, Panconesi, Radhakrishnan, Srinivasan 2005] [Derbel, Gavoille, Peleg, Viennot 2008] [Pettie 2010]
Multi-Message Broadcast vs. Spanners

Multi-message broadcast

Multi-message broadcast Code for node $v$
1. In round $r$ do:
2. Send MSG to $u$
3. ...
The Congestion Models

Two congestion models:

**Edge-congested model**
- Message size $O(\log n)$
- Classic CONGEST model [Peleg 2000]

**Node-congested model**
- Message size $O(\log n)$
- Communication by local broadcast
  - Send same message to all neighbors
Toy example: bottleneck graph, throughput $\leq 1$

Time is $O(D+N)$ for $N$ messages [Topkis 1985]
But what if connectivity is larger?
Can we hope for throughput of $k$ and shorter broadcast time?
Graph Connectivity

$\lambda$-connectivity: removal of any $\lambda-1$ edges (vertices) does not disconnect the graph

Menger’s Theorem: $\lambda$ edge-(vertex-) disjoint paths between every pair of nodes
CONGEST: Spanning Tree Packings

$S$ is a spanning tree
- Send msg $M$ on $S$

$N$ messages?
**Spanning tree packing**: set of edge-disjoint spanning trees

Fractional packing: the **weight** on each edge is at most 1
The **size** of a packing is the total weight of all trees

Spanning tree packing of size $N \rightarrow$ throughput of $\Omega(N)$
CONGEST:
Spanning Tree Packings

Spanning tree packing of size $N \to$ mm-broadcast with throughput of $\Omega(N)$

mm-broadcast with throughput of $\Omega(N) \to$ Spanning tree packing of size $N$

[Censor-Hillel, Ghaffari, Kuhn 2014]
**CONGEST:**
Spanning Tree Packings

**MMB-E** is a mm-broadcast algorithm

Mark all edges over which msg **M** is sent

**M** reaches all nodes

Gives a spanning tree **S** (removing cycles)

Repeat for all msgs: Gives a **spanning tree packing** of size **N**
**V-CONGEST:**

**Dominating Tree Packings**

S is a dominating tree
- Send msg M on S

N messages?

**Dominating tree packing:** set of vertex-disjoint dominating trees

Fractional packing: the **weight** on each vertex is at most 1
The **size** of a packing is the total weight of all trees

Dominating tree packing of size **N** $\rightarrow$ throughput of $\Omega(N)$
**V-CONGEST:**
Dominating Tree Packings

**MMB-V** is a mm-broadcast algorithm

Mark all vertices which send msg $M$

$M$ reaches all nodes

Gives a dominating tree $S$

Repeat for all msgs: Gives a **dominating tree packing** of size $N$
V-CONGEST: Dominating Tree Packings

Dominating tree packing of size $N \rightarrow$ mm-broadcast with throughput of $\Omega(N)$

mm-broadcast with throughput of $\Omega(N) \rightarrow$ Dominating tree packing of size $N$

[Censor-Hillel, Ghaffari, Kuhn 2014]
Every spanning tree packing of a \textbf{\(\lambda\)-edge connected} graph has size at most \(\lambda\) (every spanning tree needs to cross a min-cut)

Every \(\lambda\)-edge connected graph has a spanning tree packing of size \(\lceil (\lambda - 1) / 2 \rceil\) (tight)

[\text{Tutte 1961, Nash-Williams 1961, Kundu 1974}]

Centralized algorithms for decomposing unweighted graphs in \(\tilde{O} \left( \min \{ mn, m^2 / \sqrt{n} \} \right)\) time [\text{Gabow and Westermann 1988}]

and weighted graphs in \(\tilde{O}(mn)\) time [\text{Barahona 1995}]

Distributed decomposition of $\lambda$-edge connected graphs into fractionally edge-disjoint spanning trees with total weight $\lceil (\lambda - 1) / 2 \rceil (1 - \varepsilon)$ in $\tilde{O}(D + \sqrt{n\lambda})$ rounds

A lower bound of $\Omega(D + \sqrt{n / \lambda})$ rounds for such decompositions

[Censor-Hillel, Ghaffari, Kuhn 2014]
Distributed decomposition of $k$-vertex connected graphs into fractionally vertex-disjoint dominating trees with total weight $\Omega(k/\log n)$ in $\tilde{O}(D + \sqrt{n})$ rounds

A lower bound of $\Omega(D + \sqrt{n/k})$ rounds for such decompositions
Centralized decomposition of \(k\)-vertex connected graphs into fractionally vertex-disjoint dominating trees with total weight \(\Omega(k/\log n)\) in \(\tilde{O}(m)\) time

Centralized and distributed \(O(\log n)\)-approximation algorithms for the vertex connectivity of a graph in \(\tilde{O}(m)\) time, and \(\tilde{O}(D + \sqrt{n})\) rounds, respectively
Multi-Message Broadcast vs. Tree Packings

Multi-message broadcast

Code for node $v$

1. In round $r$ do:
2. Send MSG to $u$
3. ...

Tree packings
Use $k$ colors, neighbors have different colors

**Acyclic orientation**: no directed cycles

**Length**: length of longest path

$AO$ is an acyclic orientation of length $k$
- For $i=1,\ldots,k+1$
  - Color $i$ each node whose parents have been colored
AO is an acyclic orientation, length $k$ and in-degree $d$

- For $i=1,\ldots,k+1$
  - Color each node whose parents have been colored with an unused color

[Gallai, Hasse, Roy, Vitaver 1960's]
A legal $k$-coloring gives an acyclic orientation with length $k-1$:

Orient edges from smaller color towards larger color

- Used in coloring algorithms by orienting edges
  [Barenboim, Elkin 2008, 2009]
Coloring vs. Acyclic Orientations

Coloring

Code for node \( v \)
1. In round \( r \) do:
2. Send MSG to \( u \)
3. ...

Acyclic orientations
**MIS**: maximal independent set

A $k$-coloring gives an MIS in $k$ rounds.
- In round $i$ all remaining nodes with color $i$ enter the MIS and inform all their neighbors to drop out

**MIS** in $T(n, \Delta)$ rounds gives $(\Delta+1)$-coloring in
$T(n(\Delta+1), 2\Delta)$ rounds

[Luby 1986]
(different graph)
Algorithm for PROBLEM
Code for node v
1. In round r do:
2. Send MSG to u
3. ...

Distributed Algorithms

Graph Structures