On Network-Coding Based Distributed Optimization for Networks

Muriel Médard
Electrical Engineering and Computer Science Department
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

Collaborators

• MIT LIDS: Desmond Lun, Fang Zhao, Anna Lee, Ebad Ahmed, Clifford Choute
• MIT CSAIL: David Karger, Ben Leong
• University of Illinois Urbana-Champaign: Ralf Koetter, Niranjan Ratnakar
• California Institute of Technology: Michelle Effros, Siddharth Jaggi, Tracey Ho (previously LIDS, UIUC, Lucent)
• Lucent Bell Labs: Supratim Deb (previously LIDS)

Supported by NSF ITR, DARPA
Network coding

- Canonical example [ACLY00]

- No longer flows, but information

Randomized network coding

- The effect of the network is that of a transfer matrix from sources to receivers

- To recover symbols at the receivers, we require sufficient degrees of freedom – an invertible matrix in the coefficients of all nodes

\[
Y_{ij} = Y_{ij} + \alpha_{ij}^{km} Y_{km} + \alpha_{km}^{ij} X_{km}
\]
Distributed random network coding

- The realization of the determinant of the matrix will be non-zero with high probability if the coefficients are chosen independently and randomly.

- Probability of success over field $F \approx 1 - \frac{1}{|F|}$

- Randomized network coding can use any multicast subgraph which satisfies min-cut max-flow bound for each receiver [HKMKE03, HMSEK03, WCJ03] for any number of sources, even when correlated [HMEK04]

Robustness to failures and erasures

- For multicast recovery, the code in the interior of the network need not be changed [KM01, HMK03]

- What about packet erasures - probabilistic link failures?

![Diagram of network with nodes 1, 2, and 3 connected]
Erasure reliability

\[ \epsilon_{12} : \text{Erasure probability on link (1, 2).} \]
\[ \epsilon_{23} : \text{Erasure probability on link (2, 3).} \]

End-to-end erasure coding:
– Capacity is \( \frac{12}{10} \) packets per unit time.

As two separate channels:
– Capacity is \( \frac{10}{2} \) packets per unit time.
– Can use block erasure coding on each channel. But delay is a problem.

Random erasure approach

• For erasures, correlated or not, we can in the multicast case deal with average flows uniquely [LME04], [LMK05], [DGPHE04]:
  – Nodes store received packets in memory
  – Random linear combinations of memory contents sent out at every transmission opportunity (without waiting for full block)

• We obtain delay expressions using in effect a generalization of Jackson networks for the innovative packets:
  – Keep track of the propagation of innovative packets - packets whose auxiliary encoding vectors (transformation with respect to the packets injected into the node’s memory) are linearly independent across cuts
  – For Poisson arrivals, propagation of innovative packets through any node forms a stable \( M/M/1 \) queueing system in steady-state

• Scheme can be operated ratelessly - can be run indefinitely until successful reception
The use of distributed approaches

- Distributed approaches lie at the core of current routing schemes
- Enables network to operate with nodes behaving as honest middlemen negotiating with their direct business contacts
- Example: Distributed Bellman-Ford
- Can we use network coding to do distributed optimization?

Relaxing the constraints on trees

Integrality constraint arises in time-sharing between the blue and red trees
Relaxing the constraints on trees

Integrality constraint arises in time-sharing between the blue and red trees
Relaxing the constraints on trees

Index on receivers rather than on processes

\( (z_{ij}, x_{ij}^{(1)}, x_{ij}^{(2)}) \)

Relaxing the constraints on trees

\[
\begin{align*}
\text{minimize} & \quad f(z) \\
\text{subject to} & \quad z \in Z, \\
& \quad z_{ij} \geq x_{ij}^{(t)} \geq 0, \\
& \quad \sum_{j \mid (i, j) \in A} x_{ij}^{(t)} - \sum_{j \mid (j, i) \in A} x_{ij}^{(t)} \\
& \quad = \begin{cases} 
R & \text{if } i \text{ source} \\
-R & \text{if } i \text{ sink} \\
0 & \text{otherwise.}
\end{cases}
\end{align*}
\]

\( (z_{ij}, x_{ij}^{(1)}, x_{ij}^{(2)}) \)
Optimization

- For any convex cost functions
- The vector $z$ is part of a feasible solution for the optimization problem if and only if there exists a network code that sets up a multicast connection in the graph $G$ at average rate arbitrarily close to $R$ from source $s$ to terminals in the set $T$ and that puts a flow arbitrarily close to $z_{ij}$ on each link $(i, j)$
- Proof follows from min-cut max-flow necessary and sufficient conditions
- Polynomial-time
- Can be solved in a distributed way
- Steiner-tree problem can be seen to be this problem with extra integrality constraints

Distributed approach

minimize $\sum_{(i,j) \in A} f_{ij}(z_{ij})$

subject to $z_{ij} \geq x_{ij}^{(t)} \geq 0$,

$$\sum_{\{j \mid (i, j) \in A\}} x_{ij}^{(t)} - \sum_{\{j \mid (j, i) \in A\}} x_{ji}^{(t)} = \sigma_i^{(t)}$$

Special case of strictly convex, monotonically increasing cost function per arc
Distributed approach

$$\text{minimize} \sum_{(i,j) \in A} f_{ij}(z'_{ij})$$

subject to

$$z'_{ij} = \left( \sum_{t \in T} (x^{(t)}_{ij})^n \right)^{1/n}$$

$$\sum_{\{j \in \partial_i \cap A\}} x^{(t)}_{ij} - \sum_{\{j \in \partial_i \cap A\}} x^{(t)}_{ji} = \sigma^{(t)}_i,$$

$$x^{(t)}_{ij} \geq 0.$$ 

Consider approximation in large $n$ limit

Distributed approach

$$U(x) := -\sum_{(i,j) \in A} f_{ij}\left(\left(\sum_{t \in T} (x^{(t)}_{ij})^n \right)^{1/n}\right).$$

yields Lagrangian: $L(x, p, \lambda) =$

$$L(x, p, \lambda) = U(x) - \sum_{t \in T} \left\{ \sum_{i \in N} p_i^{(t)} \left( \sum_{\{j \in \partial_i \cap A\}} x^{(t)}_{ij} - \sum_{\{j \in \partial_i \cap A\}} x^{(t)}_{ji} - \sigma^{(t)}_i \right) - \sum_{(i,j) \in A} \lambda^{(t)}_{ij} x^{(t)}_{ij} \right\}.$$
Distributed approach

\[ \dot{x}^{(t)}_{ij} = k^{(t)}_{ij} \left( x^{(t)}_{ij} \right) \left( \frac{\partial U(x)}{\partial x^{(t)}_{ij}} - q^{(t)}_{ij} + \lambda^{(t)}_{ij} \right), \]

Continuous-time primal-dual algorithm

\[ p^{(t)}_i = h^{(t)}_i \left( p^{(t)}_i \right) \left( y^{(t)}_i - \sigma^{(t)}_i \right), \]

\[ \dot{\lambda}^{(t)}_{ij} = m^{(t)}_{ij} \left( \lambda^{(t)}_{ij} \right) \left( -x^{(t)}_{ij} \right)^+ \lambda^{(t)}_{ij} \]

\[ q^{(t)}_{ij} := p^{(t)}_i - p^{(t)}_j, \]

\[ y^{(t)}_i := \sum_{\{j | (i,j) \in A\}} x^{(t)}_{ij} - \sum_{\{j | (j,i) \in A\}} x^{(t)}_{ji}. \]
Dynamic approach

- We may take into account not only changes in a reactive fashion, but also in a predictive fashion.
- Reactive fashion should exhibit attributes similar to distributed Bellman-Ford – will depend on $N_S$, number of computation iterations within a coherence time of the network topology.
- Predictive fashion is effect relaxation constraints of on-line Steiner tree problem and combinations of trees.
Mobile wireless networks

Movement pattern for a 5-node random network in 20 time units under the Random Direction Mobility Model with \( \text{minspeed} = 0.2 \) and \( \text{maxspeed} = 0.5 \).

Reactive approach

Average cost of a random 4-terminal multicast in a 30-node mobile wireless network, with \( N_s = 50 \). For the modified primal recovery method,
Reactive approach

• Set of new entrants $V$ and departures $W$ to/from multicast group

• We have new flows derive from previous flows

$$z^{(m+1)} = \mu_m(z^{(m)}, T_m)$$

• Feasibility preservation: $z^{(m+1)}$ must lie in

$$U(z^{(m)}, T_m)$$

Pro-active approach

Extra energy required for random multicasts in mobile wireless networks using decentralized subgraph optimization scheme in terms of percentage of the optimal value. There are 30 nodes in the random network.
Pro-active approach

• Consider cost to be expected value of total cost over all times: undiscounted, infinite-horizon dynamic programming problem
• Use Bellman’s equation:

\[ J^*(z, T) = \min_{u \in U(z, T)} \left\{ \sum_{(i, j) \in A} f_{ij}(u_{ij}) + \mathbb{E}[J^*(u, (T \setminus V) \cup W)] \right\} \]

• Optimal policy is:

\[ \mu(z, T) = \arg \min_{u \in U(z, T)} \left\{ \sum_{(i, j) \in A} f_{ij}(u_{ij}) + \mathbb{E}[J^*(u, (T \setminus V) \cup W)] \right\} \]

Issues of queueing

• Queuing issues lead, in broad terms, to two effects: loss and delay, which are tightly coupled
• We may consider the two jointly, taking queue size into account – may in lead in general networks to difficult set-up
• We may separate the two and furthermore subdivide delay:
  – Consider delay due to congestion through convex cost function
  – Consider loss of packets in a parametric fashion if needed
  – Consider delay in reception of degrees of freedom for network coding
Conclusions and further directions

- The ability to perform distributed optimization with network coding may enable new ways of approaching wireless networking
- Many questions remain;
  - How do we incorporate lower layer coding issues with network coding and maintain a distributed approach (wireless multiple access, degraded broadcast channels)?
  - How do delay issues interact with other protocols or should we even consider forfeiting some of these protocols?
  - Do domain-based approaches suggest themselves and should coding be allowed among these domains?
  - What are the pricing consequences of network coding?
  - How can we blend distributed optimization and randomized network coding for downloading purposes?
Decentralized code construction and network coding for multicast with a cost criterion
Overview

- Randomized construction and its error behavior
- Performance of distributed randomized construction - case studies
- Traditional methods based on flows - a review
- Trees for multicasting - a review
- Network coding with a cost criterion - flow-based methods for multicasting through linear programming
• Distributed operation - one approach

• A special case - wireless networks

• Sample ISPs
Linear network coding

\[ X_i \text{ originating at } Y_j \rightarrow Y_k \rightarrow \text{ receiver } \]

\[ Z_{i;k} = b_{i;k} Y_j + b_{i;k} Y_k \]

Given network-constrained transfer matrices \((A; F; B)\), a network code is a matrix \(M\) such that

\[
\begin{bmatrix} X_1 & X_2 & \cdots & X_r \end{bmatrix} M = \begin{bmatrix} Z_{1} & Z_{2} & \cdots & Z_{r} \end{bmatrix}
\]
Linear network coding for multicast

\[
\begin{align*}
Y_j & \quad Y_k \\
\text{source } X_i & \quad \text{originating at } v
\end{align*}
\]
Linear network coding for multicast

\[ Y_j \quad Y_k \]

source \( X_i \) originating at \( v \)

\[ Y_l = a_{1,3}X_i + f_{1,3}Y_j + f_{2,3}Y_k \]
Linear network coding for multicast

\[ Y_j \rightarrow Y_k \rightarrow Y_j \]

source \( X_i \) originating at \( v \)

\[ Y_l = a_{1,3}X_i + f_{1,3}Y_j + f_{2,3}Y_k \]

receiver \( \beta \)
Linear network coding for multicast

\[ Y_j \quad Y_k \quad Y_j \quad Y_k \]

source $X_i$ originating at $v$

$Y_l = a_{1,3}X_i + f_{1,3}Y_j + f_{2,3}Y_k$

receiver $\beta$

$Z_{\beta,i} = b_{\beta,i,k}Y_j + b_{\beta,i,k}Y_k$
Linear network coding for multicast

\[ Y_j \xrightarrow{v} Y_k \xrightarrow{\beta} Y_j \xrightarrow{v} Y_k \]

source \( X_i \) originating at \( v \)

\[ Y_l = a_{1,3}X_i + f_{1,3}Y \text{output} + f_{2,3}Y_k \]

receiver \( \beta \)

\[ Z_{\beta,i} = b_{\beta,i,k}Y_j + b_{\beta,i,k}Y_k \]

- Coefficients \( \{a_{i,j}, f_{l,j}, b_{\beta,i,l}\} \) give network-constrained transfer matrices \( (A, F, \{B_{\beta}\}) \), a network code

- Matrix \( M_{\beta} = A(I - F)^{-1}B_{\beta}^T \) gives transfer function from sources to outputs [KM01]:

\[
\begin{bmatrix} X_1 & X_2 & \ldots & X_r \end{bmatrix} M_{\beta} = \begin{bmatrix} Z_{\beta,1} & Z_{\beta,2} & \ldots & Z_{\beta,r} \end{bmatrix}
\]
Linear network coding for multicast

\[
\begin{align*}
Y_j & \quad Y_k & \quad Y_j & \quad Y_k \\
\text{source } X_i & \quad \text{originating at } v & \quad \text{receiver } \beta \\
Y_l &= a_{1,3}X_i + f_{1,3}Y_{\text{output}} + f_{2,3}Y_k \\
& \quad + b_{\beta_{i,k}}Y_j + b_{\beta_{i,k}}Y_k
\end{align*}
\]

- Coefficients \( \{a_{i,j}, f_{l,j}, b_{\beta_{i,l}}\} \) give network-constrained transfer matrices \((A, F, \{B_\beta\})\), a network code.

- Matrix \( M_\beta = A(I - DF)^{-1}B_\beta^T \) gives transfer function from sources to outputs [KM01]:

\[
[X_1 \ X_2 \ldots \ X_r] \ M_\beta = [Z_{\beta,1} \ Z_{\beta,2} \ldots \ Z_{\beta,r}]
\]
Feasibility and code construction

Determining feasibility

- min-cut max-flow bound satisfied for each receiver [ACLY00]

- transfer matrix \( A(I - F')^{-1}B^T_\beta \) for each receiver \( \beta \) is non-singular [KM01]

Constructing linear solutions

- Centralized
- Direct algebraic solution using transfer matrix of [KM01]
- Algorithms using subgraph consisting of flow solutions to individual receivers [SET03, JCJ03]

- Decentralized
  - A distributed randomized network coding approach [HKMKE03]
Randomized network coding

- Interior network nodes independently choose random linear mappings from inputs to outputs

- Coefficients of aggregate effect communicated to receivers
Randomized network coding

- Interior network nodes independently choose random linear mappings from inputs to outputs.

- Coefficients of aggregate effect communicated to receivers.

- Receiver nodes can decode if they receive as many independent linear combinations as the number of source processes.
[HKMKE03, HMSEK03] For a feasible $d$-receiver multicast connection problem on a network with

- independent or linearly correlated sources

- a network code in which code coefficients $a_{i,j}$, $f_{l,j}$ for $\eta$ links are chosen independently and uniformly over $\mathbb{F}_q$

the success probability is at least $(1 - d/q)^\eta$ for $q > d$. Error bound is of the order of the inverse of the field size, so error probability decreases exponentially with codeword length.
Proof outline

- Recall transfer matrix \( M_\beta = A(I - F)^{-1}B^T_\beta \) for each receiver \( \beta \) must be non-singular.

- We show an equivalent condition connected with bipartite matching: the Edmonds matrices
  \[
  \begin{bmatrix}
  A & 0 \\
  I - F & B^T_\beta
  \end{bmatrix}
  \]
  (in the acyclic delay-free case) or
  \[
  \begin{bmatrix}
  A & 0 \\
  I - DF & B^T_\beta
  \end{bmatrix}
  \]
  (in the case with delays) are non-singular.

- This shows that if \( \eta \) links have random coefficients, the determinant polynomial
— has maximum degree $\eta$ in the random variables $\{a_{x,j}, f_{i,j}\}$

— is linear in each of these variables
Proof outline (cont’d)

• We want the product of the $d$ receivers’ determinant polynomials to be nonzero

• We can show inductively, using the Schwartz-Zippel Theorem, that for any polynomial $P \in \mathbb{F}[\xi_1, \xi_2, \ldots]$ of degree $\leq dn$, in which each $\xi_i$ has exponent at most $d$, if $\xi_1, \xi_2, \ldots$ are chosen independently and uniformly at random from $\mathbb{F}_q \subseteq \mathbb{F}$, then $P = 0$ with probability at most $1 - (1 - d/q)^n$ for $d < q$

• Particular form of the determinant polynomials gives rise to a tighter bound than the Schwartz-Zippel bound for general polynomials of the same total degree
Utility of distributed network coding

- Decentralized scenarios

<table>
<thead>
<tr>
<th>Receiver position</th>
<th>(2,4)</th>
<th>(4,4)</th>
<th>(8,10)</th>
<th>(10,10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomized flooding upper bound</td>
<td>0.563</td>
<td>0.672</td>
<td>0.667</td>
<td>0.667</td>
</tr>
<tr>
<td>Randomized Coding $P_{26}$ lower bound</td>
<td>0.882</td>
<td>0.827</td>
<td>0.604</td>
<td>0.567</td>
</tr>
<tr>
<td>$P_{28}$ lower bound</td>
<td>0.969</td>
<td>0.954</td>
<td>0.882</td>
<td>0.868</td>
</tr>
</tbody>
</table>
Another case study

- Results of Chou, Wu and Jain 2003
- Implemented event-driven simulator in C++
- Six ISP graphs from Rocketfuel project (UW)
  - SprintLink: 89 nodes, 972 bidirectional edges
  - Edge capacities: scaled to 1 Gbps / “cost”
  - Edge latencies: speed of light x distance
- Sender: Seattle; Receivers: 20 arbitrary (5 shown)
  - Broadcast capacity: 450 Mbps; Max 833 Kbps
  - Union of maxflows: 89 nodes, 207 edges
- Sent 20000 packets in each experiment:
  - field size: $2^{16}$; generation size (group of packets): 100;
    interleaving length: 100
Throughput

Chicago        (450 Mbps)
Pearl Harbor (525 Mbps)
Anaheim       (625 Mbps)
Boston          (733 Mbps)
SanJose       (833 Mbps)
Running the LP on sample ISPs (Rocketfuel)

Telstra, $V = 108$, $E = 306$
Ebene, $V = 88$, $E = 323$
Exodus, $V = 79$, $E = 294$