Computing Bounds on Network Capacity Regions as a Polytope Reconstruction Problem

Anthony Kim and Muriel Médard
August, 2011
Main Results

• Define a notion of network capacity region that generalizes the notion of network capacity defined by Cannons et al.

• Show the network routing capacity region is a computable rational polytope and provide exact algorithms and approximation heuristics.

• Define and show the semi-network linear coding capacity region is a computable rational polytope that inner-bounds the corresponding network linear coding capacity region.

• Show connections between computing these regions and a polytope reconstruction problem.

NOTE that our results apply when the finite field is fixed!
NOTE that our algorithms are not necessarily polynomial time!
Outline

• Selected Previous Works
• Fractional Network Coding Model
• Network Capacity Regions
• Main Results
• Graphs
• Further Discussion
Selected Previous Works

• Well known that max-flow/min-cut bounds are sufficient in the case of single-source multiple multicast networks.

  – Combined information and graph theoretic techniques to provide a computable outer bound on the network coding capacity regions.

  – Gave an explicit outer bound that improves upon the max-flow/min-cut bounds and showed its connection to a minimum-cost network coding problem.

  – Defined a notion of network routing capacity that is computable with a linear program and showed that every rational number in (0,1] is the routing capacity of some solvable network.

  – Extended Farkas’ Lemma to reduce an infinite set of linear constraints to a finite set in terms of minimal Steiner trees and applied the results to obtain the routing capacity region of undirected ring networks.
Fractional Network Coding Model

- Allows resource sharing of edge bandwidth among multiple independent information flows, as opposed to a snapshot of one such flow in the general network coding model.

- Network $\mathcal{N}$ is a 7-tuple $(\nu, \epsilon, \mu, c, \mathcal{A}, S, R)$:
  - $\nu$ is a node set;
  - $\epsilon$ is an edge set;
  - $\mu$ is a message set;
  - $c : \epsilon \rightarrow \mathbb{Z}^+$ is an edge capacity function;
  - $\mathcal{A}$ is an alphabet;
  - $S : \nu \rightarrow 2^\mu$ is a source mapping (to the messages each node generates);
  - $R : \nu \rightarrow 2^\mu$ is a receiver mapping (to the messages each node demands).

- Fractional network code solution is a 5-tuple $(F, \hat{k}, n, \mathcal{F}_e, \mathcal{F}_d)$:
  - $F$ is a finite field;
  - $\hat{k}$ is a source dimension vector;
  - $n$ is an edge dimension;
  - $\mathcal{F}_e$ is a collection of fractional edge functions (at intermediate nodes);
  - $\mathcal{F}_d$ is a collection of fractional decoding functions (at sink nodes).
Example

Network $\mathcal{N}$
$\mathcal{A} = \{a_1, a_2\}; c(e_i) = 1, \forall i.$

A Fractional Solution $(F, \hat{k}, n, \mathcal{F}_e, \mathcal{F}_d)$
$F = \mathbb{F}_2; \hat{k} = (k_1, k_2) = (1, 2); n = 2.$
Example breakdown

\((F, \hat{k}, n, \mathcal{F}_c, \mathcal{F}_d)\)

\(F = \mathbb{F}_2; \hat{k} = (k_1, k_2) = (1, 2); n = 2.\)

On the “first coordinates” of edges

On the “second coordinates” of edges
Network Capacity Regions

• Given a network $\mathcal{N}$, we define 
  Achievable coding rate vector: 
  $$\left( \frac{k_1}{n}, \ldots, \frac{k_{|\mu|}}{n} \right) \in \mathbb{Q}_{+}^{\mu}$$ 
  for which there exists a 
  fractional solution with such 
  $\hat{k} = (k_1, \ldots, k_{|\mu|})$ and $n$.

Network capacity region $\mathcal{C}$: closure of all achievable coding rate vectors in $\mathbb{R}^{\mu}$.

We similarly define $\mathcal{C}_r$ and $\mathcal{C}_l$ for routing capacity region and linear coding 
  capacity region for achievable routing and linear coding rate vectors.

• Thm:
  Let $\mathcal{N}$ be a capacitated network and $F$ a fixed finite field. The corresponding 
  network capacity region $\mathcal{C}$ is a closed, bounded and convex set in $\mathbb{R}^{\mu}_{+}$.

• Proof Sketch:
  – Straightforward to show closedness and boundedness based on definitions.
  – Use rate-sharing over the fixed finite field for convexity.
  – Note, however, the rate-sharing argument fails in the case of linear coding if 
    the finite field is not fixed.
Network Routing Capacity Region

• Thm:
The network routing capacity region $C_r$ is a bounded rational polytope in $\mathbb{R}^{|\mu|}_+$ and is computable.

• Proof Sketch (polytope)
  – Sufficient to consider “minimal” fractional routing solutions, in the usage of edge coordinates, i.e. edge bandwidth.
  – Any such solution consists of routing messages along Steiner trees $\mathcal{T}$, where $\mathcal{T}_i$ is the set of Steiner trees for message $i$ and $\mathcal{T} = \mathcal{T}_1 \cup \ldots \cup \mathcal{T}_{|\mu|}$.
  – Note that such solution satisfies the following:

$$\sum_{T \in \mathcal{T}} T(e) \cdot x(T) \leq c(e)n, \quad \forall e \in \epsilon$$
$$\sum_{T \in \mathcal{T}_i} x(T) = k_i, \quad \forall 1 \leq i \leq |\mu|$$
$$x \geq 0,$$

Where

$x(T)$ is the number of times Steiner tree $T$ is used in the solution, and $T(e)$ is an indicator that is 1 if $T$ uses edge $e$, or 0 otherwise.
Network Routing Capacity Region

• Proof Sketch (cont’d)
  – After scaling by \( n \), any such solution satisfies:
    \[
    \sum_{T \in \mathcal{T}} T(e) \cdot x(T) \leq c(e), \quad \forall e \in \mathcal{E} \\
    \sum_{T \in \mathcal{T}} x(T) \geq 0.
    \]
  – The constraints define a bounded rational polytope \( \mathcal{P}_r \), with rational extreme points in the \( |\mathcal{T}| \)-dimensional space.
  – Note that the set of minimal routing solutions map onto the dense subset of rational points inside \( \mathcal{P}_r \).
  – Routing capacity region \( C_r \) is the image of \( \mathcal{P}_r \) under the affine map:
    \[
    \psi_r : (x(T))_{T \in \mathcal{T}} \mapsto \left( \sum_{T \in \mathcal{T}_1} x(T), \ldots, \sum_{T \in \mathcal{T}_{|\mathcal{E}|}} x(T) \right).
    \]
  – Hence, a rational polytope.
Network Routing Capacity Region

• Proof Sketch (computable)
  – Compute the vertices of polytope $\mathcal{P}_r$ by any vertex enumeration algorithm: $v_1, \ldots, v_h$.
  – Compute the image of the vertices under the map $\psi_r$.
  – The network routing capacity region $C_r$ is given by the convex hull of points $\psi_r(v_1), \ldots, \psi_r(v_h)$. 
Network Routing Capacity Region

• Note the algorithm given in the proof may not be efficient:
  – Polytope $\mathcal{P}_r$ is defined in the $|\mathcal{T}|$-dimensional space, where $|\mathcal{T}|$ could be exponential in the problem description size.
  – Vertex enumeration and convex hull algorithms are expensive.

• We provide exact algorithms and approximation heuristics that might be more efficient in practice, using polytope reconstruction algorithms.

• Note that the algorithm is not necessarily polynomial-time.
Polytope Reconstruction Problem

- Given a polytope $Q$ and a ray oracle $O_{Ray}$, compute a polytope description of $Q$, using finite number of calls to the oracle.
- Gritzmann et al. gives a reconstruction algorithm with the number of calls at most $f_0(Q) + (|\mu| - 1)f_{|\mu|-1}(Q) + (5|\mu| - 4)f_{|\mu|-1}(Q)$ where $f_i(Q)$ denotes the number of i-dimensional faces of $Q$.
- Applicable to our problem.

\[ \hat{x} = \hat{r} \cdot t, t \geq 0 \]
Network Routing Capacity Region

• Our goal: construct the ray oracle $O_{Ray}$ for our problem.
  
  – From an earlier Thm: $\sum_{T \in T} T(e) \cdot x(T) \leq c(e), \quad \forall e \in \epsilon$
  
  – Given a ray $\hat{x} = \hat{q}t, t \geq 0$, we construct a linear program
    
    $\max \quad \lambda$
    
    s. t. $\sum_{T \in T} T(e) \cdot x(T) \leq c(e), \quad \forall e \in \epsilon$
    
    $\sum_{T \in T_i} x(T) \geq \lambda q_i, \quad \forall i$
    
    $x, \lambda \geq 0.$

  – Then the intersection point is exactly $\lambda_{max} \hat{q}$.
  
  – Using exact linear programming algorithm to solve the above program
    $\Rightarrow$ obtain an exact oracle $O_{Ray}$.
  
  – Using approximation algorithm to solve the above program $\Rightarrow$ obtain an approximate oracle $O_{Ray}$.
Network Routing Capacity Region

• Constructing the ray oracle $O_{Ray}$ (cont’d):
  – Using Gritzmann et al.’s reconstruction algorithm and exact oracle $O_{Ray}$
    $\Rightarrow$ we can recover the routing capacity region exactly.
  – However, Gritzmann et al.’s algorithm might not terminate if using
    approximate oracle $O_{Ray}$ because the set of approximate
    intersection points might not form a polytope.
Network Routing Capacity Region

• Constructing the ray oracle $O_{Ray}$ (cont’d):
  – Instead, the set of approximate intersection points and scaled version of it gives an envelope around the boundary of the capacity region.
  – Hence, we get approximation heuristics with no guarantees on the approximation of the region, with approximate oracle $O_{Ray}$.
  – Simple approximation heuristics might be useful; for instance, take sufficiently widespread rays from origin and take approximate intersection points.
Network Routing Capacity Region

• In our paper, we gave a combinatorial approximation algorithm for $O_{Ray}$.
• Based on the minimum-cost directed Steiner tree problem and Garg and Könemann’s work on the multi-commodity flow problem.
• Main Idea:
  – Concurrently pack Steiner trees corresponding to messages according to the ratio given by $\hat{q}$ in the ray $\hat{x} = \hat{q}t, t \geq 0$.
  – Choose the minimum-cost Steiner tree for message $i$, given the message and an intermediate result.
• Thm:
  There exists an $(1 + \omega)A$-approximation algorithm for the oracle in time $O(\omega^{-2} (|\mu| \log A |\mu| + |\epsilon|) A \log |\epsilon| \cdot T_{DSteiner})$, where $T_{DSteiner}$ is the time required to solve the minimum-cost directed Steiner tree problem with oracle $O_{DSteiner}$ within an approximation guarantee $A$.
• For instance, Charikar et al. give a set of algorithms with $A = i(i - 1)|\nu|^{1/i}$ and time $O(|\nu|^{3i})$ for any $i$ for $O_{DSteiner}$. $A = 1$ is possible with a brute-force algorithm.
Linear Coding Capacity Regions

- Linear coding capacity region $C_l$: closure of all achievable linear coding rate vectors in $\mathbb{R}^{\mu}$.
- Difficulty:
  - For routing:
    - Each minimal routing solution is an aggregate of Steiner trees for the messages, i.e. the Steiner trees form the “building blocks” for the routing solutions.
    - Since there are a finite number of Steiner trees, given a network, the edge constraints naturally define a polytope in some finite dimensional space.
    - Hence, the proof that the routing capacity region is a polytope
Linear Coding Capacity Regions

• Difficulty (cont’d):
  – For linear coding:
    • Not known whether there exists a finite set of such “building blocks” for minimal linear coding solutions.
    • Problem: coding/mixing across the messages lead to potentially infinite number of such atomic “building blocks”.
    • Possibly many \((\hat{k}, n)\) pairs for which we have solutions and which does not reduce to a set of smaller solutions, because of nontrivial coding across all the coordinates of the messages.
  • Our approach: Focus on a reasonable finite set of “building blocks” => obtain a finite dimensional polytope and, therefore, an inner bound.
Linear Coding Capacity Regions

• Given a network $\mathcal{N}$ and a fixed finite field $F$, we define:
  – Weight vectors associated with $\mathcal{N}$: vectors $w$ in $\{0, 1\}^{\mu}$ such that there exists a unit-edged scalar-linear solution where only messages $m_i$ with $w_i = 1$ are considered.
  – Partial scalar-linear code solutions: unit-edged scalar-linear solutions corresponding to the weight vectors.
  – Simple fractional linear code solutions: solutions that can be written as an aggregate of partial scalar-linear code solutions and are linear over the fixed finite field $F$.
  – Semi-network linear coding capacity region $C'_i$: closure of all coding rate vectors achievable by simple fractional linear solutions.

• Note that we could have chosen a bigger set, such as $\{0, 1, 2\}^{\mu}$, for the weight vectors, and obtain a bigger set of “building blocks” and, thus, better inner bounds; in this case, you want to send $n$ copies of messages with $w_i = n$ for $n = 0, 1, 2$. 
Linear Coding Capacity Regions

• The set of “building blocks” is finite and we follow the same argument as in the routing case.

• Thm:
  Assume a finite field $F$ is given. The semi-network linear coding capacity region $C'_l$, with respect to $F$, is a bounded rational polytope in $\mathbb{R}_{+}^{|\mu|}$ and is computable.

• Instead of the minimum-cost directed Steiner problem, we have two other associated problems in constructing the approximate oracle $\mathcal{O}_{Ray}$ in the linear coding case.
  - Fractional covering with box constraints; for averaging across weight vectors when we pack the “building blocks” according to some ratio.
  - Minimum-cost scalar-linear code problem; for computing the best possible “building block” for certain weight vector, given an intermediate packing result.
Linear Coding Capacity Regions

• Thm:
  There exists an \( (1 + \omega)B \)-approximation algorithm for oracle \( O_{Ray} \), for \( C'_l \), with time \( O(\omega^{-2}(\log A|\mu| + |\epsilon|)B \log |\epsilon| \cdot (T_{FCover} + k'T_{SLinear})) \), where \( T_{FCover} \) is the time required to solve the fractional covering problem by \( O_{FCover} \) within an approximation guarantee \( B \), \( T_{SLinear} \) is the time required to solve the minimum cost scalar-linear network code problem exactly by \( O_{SLinear} \), and \( k' \) is the total number of weight vectors associated with \( N \).

• For instance, Fleischer gives an approximation algorithm for \( O_{FCover} \) and there is a dynamic programming algorithm for \( O_{SLinear} \).

• We do not know how good of an inner bound \( C'_l \) is to \( C_l \).
Graphs

• Computed the routing capacity region and semi-network linear coding capacity region with both exact and approximate ray oracles, with fixed finite field $\mathbb{F}_2$.
• Used a 2D-polytope reconstruction algorithm.
• Hard-coded the linear programs and solved via linprog in MATLAB for exact ray oracles.
• Used our combinatorial approximation algorithm for approximate oracles for reasonable $\omega$ and constants.
Example 1

Network routing capacity region (two inner curves)

Semi-network linear coding capacity region (two outer curves)
Example 2

Network routing capacity region (two inner curves)

Semi-network linear coding capacity region (two outer curves)
Example 3

Network routing capacity region (two inner curves)

Semi-network linear coding capacity region (two outer curves)
Further Discussion

• From our work, we also obtain
  – Membership testing algorithm: given a rate vector, determines whether or not there exists a fractional solution that at least achieves the rate for routing and semi-network linear coding capacity regions.
  – Generalizes to directed networks with cycles and undirected networks.

• Our multi-dimensional notion of network capacity region generalizes the single-dimensional notion of network capacity defined by Cannons et al.

\[ \hat{x} = (1, \ldots, 1) \cdot t, t \geq 0 \]

Cannons et al.’s routing capacity

Our routing capacity region $C_r$
Further Discussion

- To answer some questions posed by Cannons et al.:
  - Efficiently algorithm to compute their notion of network routing capacity?
    => Showed combinatorial approximation algorithm for the ray oracle that essentially computes the network routing capacity.
  - Algorithm to compute network linear coding capacity?
    => Showed an algorithm for the ray oracle to compute a lower bound via semi-network linear coding capacity region.
  - Note these are not necessarily polynomial-time algorithms.

- We refer to our paper for more details.
REFERENCES


Thank you!