Multicommodity Flows in Polymatroidal Capacity Networks

Chandra Chekuri, Sreeram Kannan, Adnan Raja and Pramod Viswanath
University of Illinois, Urbana-Champaign, IL 61801
Email: {chekuri,kannan1, araja2, pramodv}@illinois.edu

Abstract

A classical result in undirected edge-capacitated networks is the approximate optimality of routing (flow) for multiple-unicast: the min-cut upper bound is within a logarithmic factor of the number of sources of the max flow [2, 3]. In this paper we focus on extending this result to a more general network model, where there are joint polymatroidal constraints on the rates of the edges that meet at a node.

For directed polymatroidal networks with symmetric demands we show that the maximum concurrent flow is within $O(\log^3 k)$ of the sparsest edge cut. We also show that for bidirected polymatroidal networks (with general demands), the maximum concurrent flow is within $O(\log^3 k)$ of the sparsest edge cut. We finally show that the rate region achievable by flows is within a factor of $O(\log^3 k)$ of the rate region defined by the cut-set bounds.

1 Introduction

Consider a communication network represented by a directed graph $G = (V, E)$. In the commonly studied scenario, each edge is labeled by a capacity, and the amount of information flowing on that edge should not exceed the capacity of the edge. A classical result in undirected edge-capacitated networks is the approximate optimality of routing (flow) for multiple-unicast: the min-cut upper bound is within a logarithmic factor of the number of sources of the max flow [2, 3].

In this paper we focus on extending this result to a more general network model, where there are joint constraints on the rates of the edges that meet at a node. This network, which we call as the polymatroidal network, was studied originally by Lawler and Martel [6], and is closely related to the submodular flows model studied by Edmonds and Giles [5].

We study directed polymatroidal networks with symmetric demands, and establish that the maximum concurrent flow is within $O(\log^3 k)$ of the sparsest edge cut. This result is based on an uncrossing argument and a directed tree representation of laminar families, using which we connect the directed polymatroidal network problem to a directed edge-capacitated problem for which a flow-cut approximation was established by [1].
Next, we study bidirected polymatroidal networks (with general demands), and establish that the maximum concurrent flow is within $O(\log^3 k)$ of the sparsest edge cut. We show that a similar result holds for approximating the sparsest partitioned “cut-set” by the maximum concurrent flow. We finally show that the rate region achievable by flows is within a factor of $O(\log^3 k)$ of the rate region defined by the cut-set bounds.

2 Flows and Cuts

2.1 Flows

Consider a communication network represented by a directed graph $G = (V,E)$, and a set $S \subseteq V$ of $k$ sources $s_1, .., s_k$ and a set $T \subseteq V$ of $k$ destinations $t_1, .., t_k$. Source $s_i$ wants to send “commodity” $i$ to destination $t_i$. Without loss of generality, we can assume $S$ and $T$ are non-overlapping, i.e., $S \cap T = \phi$. A flow of commodity $i$, is defined as a mapping $f_i : E \to \mathbb{R}^+$, which satisfies the following conservation conditions:

$$\sum_{e \in \text{In}(v)} f_i(e) = \sum_{e \in \text{Out}(v)} f_i(e) \forall v \in V \setminus \{S \cup T\},$$

$$R_i := \sum_{e \in \text{Out}(s_i)} f_i(e) = \sum_{e \in \text{In}(t_i)} f_i(e).$$

Here $R_i$ is interpreted as the rate of commodity flow $i$.

The total flow over an edge $e$ is defined as

$$f(e) := \sum_{i=1}^{k} f_i(e).$$

In polymatroidal capacity networks, we will require a flow to satisfy the following additional conditions:

$$\sum_{e \in \text{In}(v)} f(e) \leq \rho^\text{in}_v(S_v) \forall S_v \subseteq \text{In}\{v\}$$

$$\sum_{e \in \text{Out}(v)} f(e) \leq \rho^\text{out}_v(S_v) \forall S_v \subseteq \text{Out}\{v\},$$

where $\rho^\text{in}_v(\cdot)$ and $\rho^\text{out}_v(\cdot)$ are sub-modular functions on the corresponding input sets $\text{In}\{v\}$ and output set $\text{Out}\{v\}$.

A rate tuple $(R_1, ..., R_k)$ is said to be achievable if commodities 1 to $k$ can be sent at rates $R_1, ..., R_k$ simultaneously between the corresponding source-destination pairs.

The maximum concurrent flow is defined as follows: each source-destination pair $i$ has a requested demand vector $D_i$. The maximum concurrent flow is defined as the maximum value of $\lambda$ such that the rate tuple $(\lambda D_1, ..., \lambda D_k)$ is achievable.
2.1.1 Network with symmetric demands

For a polymatroidal network with symmetric demands, node $s_i$ intends to send commodity $i$ to $t_i$ and node $t_i$ intends to send commodity $s_i$ at the same rate. Therefore, in the flow conservation equations we have the following instead of (2):

$$ R_i := \sum_{e \in \text{Out}(s_i)} f_i(e) = \sum_{e \in \text{In}(t_i)} f_i(e), \quad (6) $$

$$ R'_i := \sum_{e \in \text{Out}(t_i)} f_i(e) = \sum_{e \in \text{In}(s_i)} f_i(e), \quad (7) $$

$$ R_i = R'_i \quad \forall i = 1 \ldots k. \quad (8) $$

Here $R_i$ is the rate of flow from $s_i$ to $t_i$ and $R'_i$ is the rate of flow from $t_i$ to $s_i$.

The maximum concurrent flow is again defined as the maximum value of $\lambda$ such that the rate tuple $(\lambda D_1, \ldots, \lambda D_k)$ is achievable.

Conceptually, this problem can be viewed as a regular maximum concurrent flow problem with $2k$ sources, $s_1, \ldots, s_k, s'_1, \ldots, s'_k$, where $s'_i = t_i$ and corresponding destinations $t_1, \ldots, t_k, t'_1, \ldots, t'_k$ with $t'_i = s_i$, and the symmetric rate constraints can be enforced by setting the demands as $(D_1, \ldots, D_k, D_1, \ldots, D_k)$.

2.2 Cuts

Consider a directed network with polymatroidal constraints.

**Definition 1.** Partitioned edge cut A partitioned edge cut $F$ is defined as a set of edges $F \subseteq E$, along with a partition $\mathcal{F} = \{F_1, \ldots, F_l\}$ such that

$$ F = \biguplus_{i=1}^l F_i, \quad (9) $$

where in any given partition $F_i$, all edges share either the head or the tail, i.e.,

$$ \text{head}(e) = u_i \quad \forall e \in F_i, \quad (10) $$

or

$$ \text{tail}(e) = v_i \quad \forall e \in F_i. \quad (11) $$

The value of a partition $F$ is given by

$$ \rho(F_i) := \begin{cases} 
\rho_{u_i}^{\text{in}}(F_i), & \text{if } \text{head}(F_i) = u_i \\
\rho_{v_i}^{\text{out}}(F_i), & \text{if } \text{tail}(F_i) = v_i 
\end{cases}. \quad (12) $$

We will use the similarly defined shorthand $\rho(E_i)$ to denote the capacity constraint on any set of edges $E_i$ that either share a head or a tail. Here the subscript $v$, and the superscript “in” or “out” are suppressed, and can be understood from the argument $E_i$. 

3
The value of a partitioned edge cut is then given by
\[ v(F) := \sum_{i=1}^{l} \rho(F_i). \] (13)

A source-destination pair \( i \) is said to be disconnected by a partitioned edge cut if
\[ p \cap F \neq \emptyset \quad \forall p \in \mathcal{P}_{(s_i, t_i)}, \] (14)
where \( \mathcal{P}_{(s_i, t_i)} \) is the set of paths from \( s_i \) to \( t_i \). Set \( d_i = 1 \) if \( (s_i, t_i) \) is disconnected by the edge cut, otherwise set \( d_i = 0 \).

The sparsity of a partitioned edge cut \( \mathcal{F} \) is defined as
\[ u(\mathcal{F}) = \frac{v(\mathcal{F})}{\sum_i d_i D_i}. \] (15)

The “sparsest partitioned edge cut” problem is to find the partitioned edge cut with minimum sparsity,
\[ \mathcal{F}^* = \arg \min_\mathcal{F} s(\mathcal{F}). \] (16)

Given a partitioned edge cut, the total rate of the disconnected \( (s_i, t_i) \) pairs is upper bounded by the value of the partitioned edge cut,
\[ \sum_i d_i R_i \leq v(F). \] (17)

For the concurrent flow problem, \( R_i = \lambda D_i \), and we get
\[ \lambda \leq \frac{v(F)}{\sum_i d_i D_i} = s(\mathcal{F}). \] (18)

We can summarize this as the following.

**Lemma 1.** The value of the maximum concurrent flow is upper bounded by the value of the sparsest partitioned edge cut.

### 2.2.1 Network with symmetric demands

For a network with symmetric demands, the sparsest partitioned edge cut can be defined as follows. Given a partitioned edge cut \( \mathcal{F} \), if \( (s_i, t_i) \) is disconnected, then set \( d_i = 1 \), if not, set \( d_i = 0 \) (as before), now additionally define \( d'_i = 1 \) if \( (t_i, s_i) \) is disconnected, if not, set \( d'_i = 0 \). The sparsity of the partitioned edge cut can be given as
\[ s(\mathcal{F}) = \frac{v(\mathcal{F})}{\sum_i (d_i + d'_i) D_i}. \] (19)

We can view the sparsest edge cut problem in the network with symmetric demands as a special case of a general network problem with \( 2k \) sources and \( 2k \) destinations as done in Sec. 2.1.1 to show the following lemma:

**Lemma 2.** For a network with symmetric demands, the value of the maximum concurrent flow is upper bounded by the value of the sparsest partitioned edge cut.
3 Main Result: Max Concurrent Flow to Sparsest Partitioned Edge Cut Ratio

Theorem 1. For a directed network with polymatroidal constraints, with \( k \) source-destination pairs, and symmetric demands from \( s_k \) to \( t_k \) and \( t_k \) to \( s_k \), the ratio between the sparsest partitioned edge cut and the maximum concurrent flow is \( O(\log^3 k) \).

Proof. The proof can be outlined as follows.

- We begin with the observation that max concurrent flow is a linear optimization program. This allows us to write the dual of this program.
- We analyze the properties of the optimal solution set of the dual-of-flow program, and show that only constraints corresponding to a certain laminar family are active.
- Next, we obtain a “tree representation” for a node having a laminar family of constraints on the input.
- We then use the tree representation to construct an edge-capacitated network for which the dual of concurrent flow problem with symmetric demands corresponds exactly to \( \tilde{\mathcal{P}}_{d.s.} \).
- We proceed to show that the flow and edge-cut in the constructed edge-capacitated network are related to the flow and cut of the original polymatroidal network.
- Finally, we use existing results [1] to obtain an edge-cut in the wireline network whose sparsity which is within \( O(\log^3 k) \) of the maximum concurrent flow.

This completes the proof. In the following, we formally derive the steps involved in the proof of our main result.

3.1 Flow Optimization Problem

We begin by writing the optimization problem for the maximum concurrent flow in directed network with symmetric demands.

For a path \( p \in \mathcal{P}_{(s_i, t_i)} \), we will use the variable \( f(p) \) to denote the amount of flow from \( s_i \) to \( t_i \) through the path \( p \). The end points of the path \( p \) uniquely specify \( s_i \) and \( t_i \). The max
concurrent flow problem can now be written mathematically as

$$\max \lambda$$  \hspace{1cm} (20)

$$\text{s.t.} \quad \sum_{p \in P_{(s_i, t_i)}} f(p) \geq \lambda D_i, \forall i$$  \hspace{1cm} (21)

$$\sum_{e \in E} \sum_{p \in p \subseteq S_v} f(p) \leq \rho_v^{in}(S_v) \forall S_v \subseteq \text{In}(v) \forall v$$  \hspace{1cm} (22)

$$\sum_{e \in E} \sum_{p \in p \subseteq S_v} f(p) \leq \rho_v^{out}(S_v) \forall S_v \subseteq \text{Out}(v) \forall v$$  \hspace{1cm} (23)

$$\sum_{p \in P_{(s_i, t_i)}} f(p) = \sum_{p \in P_{(t_i, s_i)}} f(p) \forall i$$  \hspace{1cm} (24)

$$f(p) \geq 0 \forall p.$$  \hspace{1cm} (25)

### 3.1.1 Dual of Flow

We now write the dual of the flow linear program to shed some insight on the structure of the optimal flow, and its connection to the cut.

Let the dual variables be $l_i, d_v^{in}(S_v), d_v^{out}(S_v), m_i$ corresponding to the constraints (21), (22), (23) and (24) respectively.

$$\min \sum_{v \in V} \sum_{S_v \subseteq \text{In}(v)} d_v^{in}(S_v) \rho_v^{in}(S_v) + \sum_{v \in V} \sum_{S_v \subseteq \text{Out}(v)} d_v^{out}(S_v) \rho_v^{out}(S_v)$$  \hspace{1cm} (26)

$$\text{s.t.} \quad \sum_{i} l_i D_i \geq 1$$  \hspace{1cm} (27)

$$\sum_{e \in E} \left\{ \sum_{S_v \subseteq \text{In}(v); e \in S_v} d_v^{in}(S_v) + \sum_{S_v \subseteq \text{Out}(v); e \in S_v} d_v^{out}(S_v) \right\} + m_i \geq l_i \forall p \in P_{(s_i, t_i)} \forall i$$  \hspace{1cm} (28)

$$\sum_{e \in E'} \left\{ \sum_{S_v \subseteq \text{In}(v); e \in S_v} d_v^{in}(S_v) + \sum_{S_v \subseteq \text{Out}(v); e \in S_v} d_v^{out}(S_v) \right\} - m_i \geq 0 \forall p' \in P_{(t_i, s_i)} \forall i$$  \hspace{1cm} (29)

$$d_v^{in}(S_v) \geq 0 \forall v \in V \forall S_v \subseteq \text{In}(v)$$  \hspace{1cm} (30)

$$d_v^{out}(S_v) \geq 0 \forall v \in V \forall S_v \subseteq \text{Out}(v).$$  \hspace{1cm} (31)

Now, we can use the fact that

$$a_i + m_i \geq l_i \text{ and } b_i - m_i \geq 0$$  \hspace{1cm} (32)

$$\iff a_i + b_i \geq l_i,$$  \hspace{1cm} (33)
if \( m_i \) is not constrained in sign, in order to simplify the equations (28) and (29) into a single equation. This gives us the equivalent program

\[
\min \sum_{v \in V} \sum_{S_v \subseteq \text{In}(v)} d^\text{in}_v(S_v) \rho^\text{in}_v(S_v) + \sum_{v \in V} \sum_{S_v \subseteq \text{Out}(v)} d^\text{out}_v(S_v) \rho^\text{out}_v(S_v) \tag{34}
\]

s.t.

\[
\sum_i l_i D_i \geq 1 \tag{35}
\]

\[
\sum_{e \in P} \sum_{S_v \subseteq \text{In}(v): e \in S_v} d^\text{in}_v(S_v) + \sum_{S_v \subseteq \text{Out}(v): e \in S_v} d^\text{out}_v(S_v) \geq \sum_{e \in P^\prime} \sum_{S_v \subseteq \text{In}(v): e \in S_v} d^\text{in}_v(S_v) + \sum_{S_v \subseteq \text{Out}(v): e \in S_v} d^\text{out}_v(S_v) \geq l_i \quad \forall p \in P(s_i, t_i), p' \in P(t_i, s_i) \forall i \tag{36}
\]

Let us define

\[
d(e) := \sum_{S_v \subseteq \text{In}(v): e \in S_v} d^\text{in}_v(S_v) + \sum_{S_v \subseteq \text{Out}(v): e \in S_v} d^\text{out}_v(S_v) \tag{39}
\]

Then the constraint (36) becomes,

\[
\min_{p \in P(s_i, t_i), p' \in P(t_i, s_i)} \sum_{e \in P} d(e) + \sum_{e \in P^\prime} d(e) \geq l_i \tag{40}
\]

\[
\iff \min_{p \in P(s_i, t_i)} \sum_{e \in P} d(e) + \min_{p' \in P(t_i, s_i)} \sum_{e \in P^\prime} d(e) \geq l_i. \tag{41}
\]

The dual-of-flow program can now be written as

\[
\min \sum_{v \in V} \sum_{S_v \subseteq \text{In}(v)} d^\text{in}_v(S_v) \rho^\text{in}_v(S_v) + \sum_{v \in V} \sum_{S_v \subseteq \text{Out}(v)} d^\text{out}_v(S_v) \rho^\text{out}_v(S_v) \tag{42}
\]

s.t.

\[
\sum_i l_i D_i \geq 1 \tag{43}
\]

\[
\sum_{S_v \subseteq \text{In}(v): e \in S_v} d^\text{in}_v(S_v) + \sum_{S_v \subseteq \text{Out}(v): e \in S_v} d^\text{out}_v(S_v) =: d(e) \tag{44}
\]

\[
\min_{p \in P(s_i, t_i)} \sum_{e \in P} d(e) + \min_{p' \in P(t_i, s_i)} \sum_{e \in P^\prime} d(e) \geq l_i \tag{45}
\]

\[
d^\text{in}_v(S_v) \geq 0 \quad \forall v \in V \quad \forall S_v \subseteq \text{In}(v) \tag{46}
\]

\[
d^\text{out}_v(S_v) \geq 0 \quad \forall v \in V \quad \forall S_v \subseteq \text{Out}(v). \tag{47}
\]

The feasibility of this program depends only on the value of \( d(e) \), and not on the finer specification of the values of \( d^\text{in}_v(\cdot) \) and \( d^\text{out}_v(\cdot) \).
3.2 Properties of the Dual of Flow program

**Lemma 3.** There exists an optimal solution such that \( \forall v, \{S_v : d_v^{in}(S_v) > 0\} \) is a laminar family; and \( \forall v, \{S_v : d_v^{out}(S_v) > 0\} \) is also a laminar family.

**Proof.** Among all the \( \{d_v^{in}(\cdot), d_v^{out}(\cdot)\} \) which are feasible and have the same optimal objective function value in the dual of flow program, choose \( d_v^{in}(\cdot), d_v^{out}(\cdot) \) such that

\[
\sum_v \sum_{S_v \subseteq \text{In}(v)} d_v^{in}(S_v) \cdot |S_v| \cdot |\text{In}(v) \setminus S_v| + \sum_v \sum_{S_v \subseteq \text{Out}(v)} d_v^{out}(S_v) \cdot |S_v| \cdot |\text{Out}(v) \setminus S_v|, \tag{48}
\]

is minimized.

We will show that such a \( d_v(\cdot) \) the set \( F_v^{in} := \{S_v : d_v^{in}(S_v) > 0\} \) and \( F_v^{out} := \{S_v : d_v^{out}(S_v) > 0\} \) should be laminar for every \( v \). Suppose not, let \( F_v^{in} \) be non-laminar for a specific \( v \). We will now show that there exists a \( d_v^{in} \) which is feasible, the objective function is no greater and the value of

\[
\sum_{S_v \subseteq \text{In}(v)} d_v^{in}(S_v) \cdot |S_v| \cdot |\text{In}(v) \setminus S_v|, \tag{49}
\]

is strictly less than that of \( d_v^{in} \).

By assumption of non-laminarity of \( F_v^{in} \), there exists \( A, B \in F_v^{in} \) such that \( A \cap B \neq \emptyset, A \not\subseteq B \) and \( B \not\subseteq A \). Let, without loss of generality, \( d_v^{in}(A) \geq d_v^{in}(B) > 0 \).

Define

\[
\begin{align*}
d_v^{in}(A) & := d_v^{in}(A) - d_v^{in}(B) \quad \tag{50} \\
d_v^{in}(B) & := d_v^{in}(B) - d_v^{in}(B) = 0 \quad \tag{51} \\
d_v^{in}(A \cap B) & := d_v^{in}(A \cap B) + d_v^{in}(B) \quad \tag{52} \\
d_v^{in}(A \cup B) & := d_v^{in}(A \cup B) + d_v^{in}(B) \quad \tag{53}
\end{align*}
\]

and for the other sets \( d_v^{in}(C) = d_v^{in}(C) \).

Claim: \( d_v^{in}(\cdot) \) is feasible. In particular \( d(e) = d'(e), \forall e \).

Proof: It is sufficient to show that

\[
\sum_{S_v \subseteq \text{In}(v) : e \in S_v} d_v^{in}(S_v) = \sum_{S_v \subseteq \text{In}(v) : e \in S_v} d_v^{in}(S_v). \tag{54}
\]

This implies that \( d(e) = d'(e) \), which in turn implies that the feasibility of the constraints are unaltered.

Case 1: Let \( e \in A, e \not\in B \), then \( e \in A \cup B, e \not\in A \cap B \). Now (54) is true, since,

\[
d_v^{in}(A) + d_v^{in}(A \cup B) = d_v^{in}(A) + d_v^{in}(A \cup B). \tag{55}
\]

Case 2: Let \( e \in A, e \in B \). Now (54) is true since,

\[
d_v^{in}(A) + d_v^{in}(B) + d_v^{in}(A \cap B) + d_v^{in}(A \cup B) = d_v^{in}(A) + d_v^{in}(B) + d_v^{in}(A \cap B) + d_v^{in}(A \cup B). \tag{56}
\]
The optimal solution of

3.2.1 Reduced Flow Program

the value of the program

families for all the sets. Therefore even if we add the constraint that

families. Let us call this set Ξ, which is the union of the incoming and outgoing laminar

v node laminar polymatroidal constraint

Lemma 4.

by the submodularity of $\rho^v$.

Claim:

$$\sum_{S_v \in \text{In}(v)} d^{\text{in}}_v(S_v) |S_v||\text{In}(v) \setminus S_v| < \sum_{S_v \in \text{In}(v)} d^{\text{in}}_v(S_v) |S_v||\text{In}(v) \setminus S_v|. \quad (62)$$

Proof: Let us recall the following lemma (Chapter 2, Theorem 2.1 in [4]):

**Lemma 4.** [4] If $T$ and $U$ are subsets of a set $S$ with $T \not\subseteq U$ and $U \not\subseteq T$, then

$$|T||S \setminus T| + |U||S \setminus U| > |T \cap U||S \setminus (T \cap U)| + |T \cup U||S \setminus (T \cup U)|. \quad (63)$$

The difference between the two terms is

$$\sum_{S_v \in \text{In}(v)} (d^{\text{in}}_v(S_v) - d^{\text{in}}_v(S_v)) |S_v||\text{In}(v) \setminus S_v| \quad (64)$$

$$= -d^{\text{in}}_v(B)\{|A||\text{In}(v) \setminus A| + |B||\text{In}(v) \setminus B| - |A \cup B||\text{In}(v) \setminus (A \cup B)||\text{In}(v) \setminus A \cap B|\} \quad (65)$$

$$< 0. \quad (66)$$

where (66) follows from Lemma 4.

This is a contradiction to the assumption that $d^{\text{in}}_v(\cdot)$ has the minimum value of

$$\sum_{S_v \in \text{In}(v)} d^{\text{in}}_v(S_v) |S_v||\text{In}(v) \setminus S_v|, \quad (67)$$

among all functions that are feasible and have the same optimal objective function value in the dual of flow program.

This completes the proof of the lemma. \hfill \square

3.2.1 Reduced Flow Program

The optimal solution of $\mathcal{P}^{\text{d.s.}}_I$ has the structure that for any given $v$, only for $S_v \in \mathcal{F}^{\text{in}}_v$, $d^{\text{in}}_v(S_v) > 0$ and similarly only for $S_v \in \mathcal{F}^{\text{in}}_v$, $d^{\text{out}}_v(S_v) > 0$, where $\mathcal{F}^{\text{in}}_v$ and $\mathcal{F}^{\text{in}}_v$ are laminar families. Let us call this set $\Xi$, which is the union of the incoming and outgoing laminar families for all the sets. Therefore even if we add the constraint that

$$d_v(S_v) = 0, S_v \not\subseteq \Xi, \quad (68)$$

the value of the program $\mathcal{P}^{\text{d.s.}}_I$ is unchanged. This is equivalent to a flow problem, where, node $v$ has a laminar polymatroidal constraint $\rho^{\text{in}}_v(E_v), E_v \in \mathcal{F}^{\text{in}}_v$ on its incoming edges and a laminar polymatroidal constraint $\rho^{\text{out}}_v(E_v), E_v \in \mathcal{F}^{\text{in}}_v$ on the outgoing edges.

Let us call this program with additional constraints as $\tilde{\mathcal{P}}^{\text{d.s.}}_I$. 


3.3 A Tree Representation for the Laminar Constraints

Consider a node $v$ having incoming constraints $\rho_v^{in}(E), E \in \mathcal{F}_v^{in}$, where $\mathcal{F}_v^{in}$ is a laminar family of constraints.

The connection between laminar family of sets and trees is well known and was developed by Edmonds and Giles [5]. For the discussion in this paper, it will be convenient to have an explicit construction of an edge-capacitated directed tree corresponding to a node’s incoming laminar constraints, and this will be developed here.

If the singleton subsets $\{e_1\}, \ldots, \{e_l\}$ of In$(v)$ are not already present in $\mathcal{F}_v^{in}$, let us add them to the laminar family. This does not affect the laminarity of $\mathcal{F}_v^{in}$.

Let us construct a graph with nodes $v_E, E \in \mathcal{F}_v^{in}$ and the node $v$. Let the set of edges in this graph be $T$, and add the edges $(v_E_1, v_E_2) \in T$ if $E_1 \subset E_2$, and $E_2$ is the minimal such set in $\mathcal{F}_v^{in}$, with the capacity of the edge given by

$$c_{(v_E_1, v_E_2)} := \rho(E_1),$$

and the edge is labeled $e_{E_1}$.

Further add the edges

$$(v_{E_1}, v) \in T$$

if $E_1$ is maximal in $\mathcal{F}_v^{in}$, with capacity

$$c_{(v_{E_1}, v)} := \rho(E_1),$$

and these edges are labeled $e_{E_1}$.

This completes the definition of a directed graph $(V, T)$. We will now show that it is a tree. Since any edge connects a set of smaller cardinality to an edge of larger cardinality, the graph is acyclic. Since each node has only one outgoing edge, the acyclic graph is a directed tree.
Note that since the graph is a tree there is a unique path from any node $v_{E_1}$ to $v$, denoted by $p_{E_1,v}$ comprised of the edges $\{e_{E_v} : E_1 \subseteq E_v \in \mathcal{F}^\text{in}_v\}$.

**Example:** Consider a node that has incoming polymatroidal constraints on $e_5, e_{12}, e_{34}, e_{1234}$ given by $\rho_5, \rho_{12}, \rho_{34}, \rho_{1234}$ respectively. For this example, the directed network is shown in Fig. 1.

### 3.4 Procedure Constructing an Edge-Capacitated Network

- Each node $v$ in the original network has a laminar polymatroidal constraint $\rho_v^{\text{in}}(E_v), E_v \in \mathcal{F}^\text{in}_v$ on its incoming edges and a laminar polymatroidal constraint $\rho_v^{\text{out}}(E_v), E_v \in \mathcal{F}^\text{in}_v$ on the outgoing edges.

- Since this is a directed network, we can think of all the incoming edges being grouped together (say, on the left side), and all the outgoing edges grouped together (on the right side). On the incoming side, the node $v$ is supplemented by $|\mathcal{F}^\text{in}_v|$ nodes, named $v_{E_v}, E_v \in \mathcal{F}^\text{in}_v$ using the directed tree representation of Sec 3.3. Similarly on the outgoing side, the node $v$ is supplemented by $|\mathcal{F}^\text{in}_v|$ nodes, named $v_{E_v}, E_v \in \mathcal{F}^\text{in}_v$ using the directed tree representation of Sec 3.3 with the edges reversed. Thus, each node $v$ is replaced by $|\mathcal{F}^\text{in}_v| + 1 + |\mathcal{F}^\text{in}_v|$ nodes.

- Now we need to specify how the connections between various nodes are made in the new network. An edge $e_i = (u,v)$ in the polymatroidal network is now represented twice in the polymatroidal network once in the incoming tree representation for $v$, and once in the outgoing tree representation for $u$. There are corresponding nodes $v_{e_i}$ and $u_{e_i}$. We will now collapse these two nodes into a single node.

- There is a one-to-one correspondence between the paths in the original network and the constructed network. Corresponding to any edge $e$ coming into a node $v$ in a path in the polymatroidal network, we can identify a path in the edge-capacitated network by using the unique path $p_{\{e\},v}$ in the tree representation. Doing this for every edge in the path gives us a corresponding path in the edge-capacitated network. On the other hand, given any path composed of $e_{E_i}, i = 1, 2, ..., l$ in the edge-capacitated network, taking the edges in the union of these sets $\cup_i E_i$, we can construct a path in the original network.

### 3.5 Relationship Between Flows and Cuts in Original and Constructed Network

Let the flow value in the constructed edge-capacitated network be given by $P^n$.

**Lemma 5.** The flow value in the constructed edge-capacitated network is identical to the flow value in the original polymatroidal network:

$$P^n_f = P^{d,s}_f.$$  \hspace{1cm} (73)
Proof. We already know that $P_{d.s.}^f = ˜P_{d.s.}^f$, i.e., we can consider a network with laminar constraints corresponding to $Ξ$ instead of considering all the polymatroidal constraints.

Let the edges in the edge capacitated network be denoted by $˜e$ and the dual variable constraining $c_e$ be denoted as $d_\tilde{e}$.

The dual of the maximum concurrent flow for the edge capacitated network with symmetric demands is given by,

$$\begin{align*}
\min \sum_{\tilde{e}} d_{\tilde{e}} c_{\tilde{e}} \\
\text{s.t.} \\
\sum_i l_i D_i \geq 1 \\
\min_{p \in \tilde{P}(s_i, t_i)} \sum_{\tilde{e} : \tilde{e} \in p} d_{\tilde{e}} + \min_{p' \in \tilde{P}(t_i, s_i)} \sum_{\tilde{e} : \tilde{e} \in p'} d_{\tilde{e}} \geq l_i \\
\tilde{d}_{\tilde{e}} \geq 0 \forall v \in V \forall \tilde{e}.
\end{align*}$$

Any edge $\tilde{e}$ can be represented as $e_{E_v}$ for some $E_v \in F_v^\text{in}$ or $E_v \in F_v^\text{out}$ in the original network, and the capacity of this edge is given by $\rho(E_v)$. Using the one-to-one correspondence between paths in the polymatroidal and edge-capacitated network, we can see that any path $p \in P(s_i, t_i)$ in the polymatroidal network can be represented in the edge capacitated network by replacing edge edge $e = (u, v) \in p$ with $\{E_v \in F_v^\text{in}, E_u \in F_u^\text{out}\}$. We can further substitute $\tilde{d}_{\tilde{e}}$ by $\tilde{d}_{E_v}$ if $\tilde{e} = e_{E_v}$, to rewrite the program as

$$\begin{align*}
\min \sum_{E_v : E_v \in F_v^\text{in}} \tilde{d}_{E_v} \rho_v^\text{in}(E_v) + \sum_{E_v : E_v \in F_v^\text{out}} \tilde{d}_{E_v} \rho_v^\text{out}(E_v) \\
\text{s.t.} \\
\sum_i l_i D_i \geq 1 \\
\sum_{e \in p} \sum_{S_v \in F_v^\text{in} : e \in S_v} \tilde{d}_{v}^\text{in}(S_v) + \sum_{S_v \in F_v^\text{out} : e \in S_v} \tilde{d}_{v}^\text{out}(S_v) \\
\sum_{e' \in p'} \sum_{S_v \in F_v^\text{in} : e' \in S_v} \tilde{d}_{v}^\text{in}(S_v) + \sum_{S_v \in F_v^\text{out} : e' \in S_v} \tilde{d}_{v}^\text{out}(S_v) \geq l_k \forall p \in P(s_i, t_i), p' \in P(t_i, s_i) \forall k
\end{align*}$$

This program $P_{inf}^f$ is identical to the program $\tilde{P}_{d.s.}^f$, and therefore

$$P_{inf}^f = \tilde{P}_{d.s.}^f = P_{d.s.}^f.$$
Lemma 6. Any edge cut in the edge-capacitated network specifies a partitioned edge-cut in the polymatroidal network of the same sparsity. Therefore,

\[ \mathcal{P}_{d.s.}^{e.c.} \leq \mathcal{P}_n^{e.c.}. \]  

(84)

Proof. Each edge cut \( \mathcal{E} \) in the edge-capacitated network specifies a set of edges \( e_{E_i}, i = 1, 2, ..., l \) where each edge \( e_{E_i} \) corresponds to some directed tree representation of the laminar constraints. The value of the edge cut in the edge-capacitated network is given as the sum of the edge capacities, therefore the value of the specified edge cut is \( \sum_i c_{E_i} = \sum_i \rho(E_i) \).

Now we can define a partitioned edge cut in the polymatroidal network as

\[ \mathcal{F} = \{ F_1, F_2, ... F_l \} \], where \( F_i := E_i, i = 1, 2, ..., l \).

(85)

The value of the partitioned edge cut is given by

\[ v(\mathcal{F}) = \sum_i \rho(F_i) = \sum_i \rho(E_i), \]

(86)

which is the value of the edge cut in the edge-capacitated network.

We now claim that the set of source-destination pairs disconnected by the edge cut \( \mathcal{E} \) in the edge-capacitated network is the same as the set of source-destination pairs disconnected by the partitioned edge cut \( \mathcal{F} \) in the polymatroidal network. Consider the contrapositive of this statement: “If there is a path between some \((s_i, t_i)\) pair in the polymatroidal network after removing the edges specified by \( \mathcal{F} \), then there is a path between the same \((s_i, t_i)\) pair in the edge-capacitated network after removing the edges specified by \( e_{E_i} \).” This statement can be shown easily by considering the one-to-one correspondence between the paths in the polymatroidal and edge capacitated network.

Since the value of the edge cut \( \mathcal{E} \) is equal to the value of the partitioned edge cut \( \mathcal{F} \), and the sum of the demands disconnected is the same in the case of the edge-capacitated network and the polymatroidal network, it follows that the sparsity of the edge cut \( \mathcal{E} \) is the same as the sparsity of the partitioned edge cut \( \mathcal{F} \).

Now, we can use the existing result to connect the flow and cuts in the edge-capacitated network:

Theorem 2. [1] For a directed edge-capacitated network, with \( k \) source-destination pairs, and symmetric demands from \( s_k \) to \( t_k \) and \( t_k \) to \( s_k \), the ratio between the sparsest edge cut and the maximum concurrent flow is \( \mathcal{O}(\log^3 k) \).

This implies that

\[ \frac{\mathcal{P}_f^n}{\mathcal{P}_n^{e.c.}} \geq \frac{1}{\mathcal{O}(\log^3 k)}. \]

(87)

Combining this with the earlier equations relating the original and new networks (83), (84), which are summarized in Fig. 2, we get that

\[ \frac{\mathcal{P}_n^{d.s.}}{\mathcal{O}(\log^3 k)} \leq \mathcal{P}_f^{d.s.} \leq \mathcal{P}_n^{d.s.}, \]

(88)

which concludes the proof of the theorem.
\[
\mathcal{P}_{f}^{d.s.} \leq \mathcal{P}_{e.c.}^{d.s.} \\
\mathcal{P}_{f}^{n} \geq \mathcal{P}_{f}^{n} \\
\frac{1}{O(\log^3 k)}
\]

Figure 2: Relationship between flows and cuts of new and original networks

4 Bidirected Polymatroidal Networks with General Demands

In this section, we show that if the network is bi-directed, then the assumption of symmetric demands can be relaxed. We begin by defining bidirected polymatroidal networks.

A bi-directed edge-capacitated network is a directed edge-capacitated where, for every edge \((i, j)\) there is a corresponding reverse edge \((j, i)\) such that \(c(i, j) = c(j, i)\).

A bi-directed polymatroidal network is a directed polymatroidal network in which

- Every edge \(e = (i, j)\) has a corresponding reverse edge \(\tau(e) := (j, i)\).
- For any vertex \(v\), the polymatroidal constraint \(\rho_v^{\text{in}}(.)\) on the incoming edges \(\text{In}(v)\) is the same as the polymatroidal constraint \(\rho_v^{\text{out}}(.)\) on the outgoing edges \(\text{Out}(v)\). More concretely,

\[
\rho_v^{\text{in}}(E_v) = \rho_v^{\text{out}}(\tau(E_v)) =: \rho_v(E_v) =: \rho_v(\tau(E_v)). \tag{89}
\]

Theorem 3. For a bidirected network with polymatroidal constraints, with \(k\) source-destination pairs, the ratio between the sparsest partitioned edge cut and the maximum concurrent flow is \(O(\log^3 k)\).

Proof. Let us consider a bi-directed polymatroidal network, with \(k\) specified sources \(s_1, ..., s_k\) and the \(k\) corresponding destinations \(t_1, ..., t_k\). In the problem with general demands, the flow requirements is \(D_i\) between \(s_i\) to \(t_i\), and the flow requirement between \(t_i\) to \(s_i\) is zero. However, in order to tackle this problem, we consider the problem with symmetric demands, where both \(s_i\) wants to send \(D_i\) flow to \(t_i\) and symmetrically \(t_i\) wants to send \(D_i\) flow to \(s_i\).

Let us call the maximum concurrent flow for the bi-directed problem with symmetric demands as \(P_{f}^{b.s.}\) and the sparsest partitioned edge-cut as \(P_{e.c.}^{b.s.}\). Similarly, let us call the maximum concurrent flow achievable for the bi-directed problem with general demands as \(P_{f}^{b.g.}\) and the sparsest edge cut as \(P_{e.c.}^{b.g.}\).

For a bi-directed polymatroidal network with symmetric demands, the gap between max concurrent flow and the sparsest partitioned edge-cut is a special case of Theorem 1. Therefore

\[
P_{f}^{b.s.} \geq \frac{P_{e.c.}^{b.s.}}{O(\log^3 k)}. \tag{90}
\]
Clearly, any rate achievable with symmetric demands is also achievable with general demands, since the former is a harder problem. So
\[ P_{b.s.}^{f} \subseteq P_{b.g.}^{f}. \] (91)

Now we will consider a partitioned edge cut with symmetric demands. This would in general disconnect either a path from \( s_i \) to \( t_i \) or a path from \( t_i \) to \( s_i \), for any given \( i \) or disconnect both. For the general demands problem, the obtained edge cut should disconnect paths from \( s_i \) to \( t_i \), \( \forall i \). The edge-cuts of the symmetric and general demands problems are connected in the following lemma:

**Lemma 7.** Given a partitioned edge-cut for the problem with symmetric demands, there exists a partitioned edge cut for the general demands problem, whose sparsity is at most a factor of 4 bigger than the sparsity of the cut for the symmetric demands problem.

**Proof.** Let us start with a partitioned edge-cut \( F \), recall that this need not disconnect all source-destination pairs. We can make this edge-cut \( F \) bi-directed by adding the reverse edges that are not already present \( F \) to this edge-cut, let us call this bidirected edge cut as \( F_{b.g.} \),

\[
F_{b.g.} = F \cup \{ \tau(F) \setminus F \},
\]

\[
v(F_{b.g.}) = v(F) + v(\tau(F) \setminus F),
\]

\[
\leq v(F) + v(\tau(F)),
\]

\[
= 2v(F).
\]

Thus the value of the obtained edge cut could at most increase the cut value by a factor of two.

Claim: If the original edge cut disconnects paths from \( t_i \) to \( s_i \) for \( i = 1, 2, ..., l \), then the bidirected edge cut \( F_{b.g.} \) disconnects all paths from \( s_i \) to \( t_i \) and all paths from \( t_i \) to \( s_i \) for all \( i = 1, 2, ..., l \).

Proof: We will show that if an edge cut \( F \) disconnects paths from \( t_i \) to \( s_i \) for all \( i \), then the edge cut \( \tau(F) \) will disconnect paths from \( t_i \) to \( s_i \). This will then imply that \( F \cup \tau(F) \) will disconnect both \( s_i \) to \( t_i \) and \( t_i \) to \( s_i \), and prove the claim. We will show the contrapositive of the requisite statement: If there is a path from some \( t_i \) to \( s_i \) among edges \( \tau F^c \), then there is a path from \( s_i \) to \( t_i \) among edges \( F^c \) for the same \( i \). Indeed this is true, since, if there is a path from \( s \) to \( t \) among edges \( A \), then there is a path from \( t \) to \( s \) among edges \( \tau(A) \), and the fact that

\[
\tau(\tau F^c) = \tau(F^c) = F^c,
\]

where we have used the fact for a set of edges the complement and \( \tau \) operators commute.

Now we are ready to prove the statement of the lemma. Let the given partitioned edge cut, for \( i = 1, 2, ..., l \), disconnect either paths from \( s_i \) to \( t_i \) (given by \( d_i = 1 \)) or \( t_i \) to \( s_i \) (given by \( d'_i = 1 \)) or both. The sparsity of the edge cut is given by

\[
s(F) = \frac{v(F)}{\sum_{i=1}^{l} (d_i + d'_i)D_i}
\]

(97)
Now the sparsity of the partitioned bidirected edge cut for the general demands problem is
\[
\begin{align*}
\sum_{i=1}^{l} D_i & \leq 2 v(F) \sum_{i=1}^{l} D_i \\
& \leq 4 \frac{v(F)}{\sum_{i=1}^{l} (d_i + d'_i) D_i} = 2 s(F),
\end{align*}
\]
(99)
since \(d_i + d'_i \leq 2\). Thus the sparsity of the obtained partitioned edge cut for the general demands problem is atmost four times the sparsity of the partitioned edge cut for the symmetric problem.

Now we relate the various variables in the problem in Fig. 3. This leads to the following relationship,
\[
\frac{P_{b.g.}^{b.g.}}{O(\log^3 k)} \leq P_{f}^{b.g.} \leq P_{e.c.}^{b.g.},
\]
(101)
which concludes the proof of the theorem.

### 4.1 Approximating Sparsest Partitioned Cut-set by Maximum Concurrent Flow

For a bidirected polymatroidal network with general demands, we proved that the sparsest partitioned edge cut is approximated by the maximum concurrent flow. In this section, we will connect the sparsest partitioned edge cut to the sparsest edge-partitioned "cut-set".

A cut-set is a partition of the vertices into two sets \(\Omega\) and \(\Omega^c\). The edges corresponding to the partition are given by
\[
E(\Omega) = \{e = (u, v) : u \in \Omega, v \in \Omega^c\}.
\]
(102)
An edge-partitioned cut-set is given by a partition of the edges $E(\Omega)$ into subsets $E_i$ such that all edges in any subset has either a common head or a common tail. Thus an edge-partitioned cut-set is a special type of partitioned edge cut where the set of edges $F = E(\Omega)$ for some $\Omega$. The sparsity of the edge-partitioned cut-set is defined as the sparsity of the partitioned edge cut $E = \bigcup_i E_i$, and this is a bound on the maximum concurrent flow $\lambda$. Note that an edge-partitioned cut-set is, by definition, directional in that only edges going from $\Omega$ to $\Omega^c$ are included in the edge cut.

We are now ready to show the relationship between sparsest partitioned edge cut, and the sparsest edge-partitioned cut-set.

**Lemma 8.** Given a partitioned bidirected edge cut for a bidirected polymatroidal network, we can obtain an edge-partitioned cut-set whose sparsity is atmost $4$ times the sparsity of the partitioned edge cut.

**Proof.** Let $\lambda$ be the maximum concurrent flow rate. Let the given edge cut be $F$. Since the edge cut is bidirected, the graph obtained after removing the edge cut is also bidirected. We can treat the graph as an undirected graph, and let us we partition the edges into sets falling in the same connected component. This induces a natural “vertex multi-partition”, $V = \cup_i V_i$, and let us define the edge cut corresponding to a vertex multi-partition as

$$E(\cup_i V_i) := \{ e = (u,v) : u \in V_i, v \in V_m, l \neq m. \} \tag{103}$$

Let us call the partitioned edge cut corresponding to the multi-partition as $F_m$. The demand separated by $F_m$ is the same as the demands separated by $F$, and the value of $F_m$ is lesser than or equal to the value of $F_m$ since $F_m \subseteq F$. Therefore the sparsity of $F_m$ is lesser than or equal to the sparsity of the partitioned edge cut $F$, i.e.,

$$s(F_m) \leq s(F). \tag{104}$$

For a given vertex multi-partition, let us define

$$D_{ij} = \sum_{m:s_m \in V_i, t_m \in V_j} D_m, \tag{105}$$

be the sum of demands between sources in $V_i$ and destinations in $V_j$. Clearly, some of the $D_{ij}$ can be zero. Thus the original demands $D_i$ are relabeled according to the vertex multi-partition.

In this terminology, the vertex multi-partition bound suggests a bound on the maximum concurrent flow,

$$\lambda \sum_{(i,j) \in |I| \times |I|} D_{ij} \leq v(F_m), \tag{106}$$

since the edges corresponding to the vertex multi-partition disconnect all these source destination pairs.

Note that the cut-set differs from vertex bi-partition (special case of vertex multi-partition with 2 partitions) in the sense that in a vertex bi-partition, the edges going from $V_1$ to $V_2 = V_1^c$ and the edges from $V_1^c$ to $V_1$ are both included in the cut, whereas in a cut-set given by a set $V_1$, only the edges going from $V_1$ to $V_1^c$ are included.
4.1.1 The Averaged Edge-Partitioned Cut-Set Bound

We will compute the average bound implied by several cut-set bounds, which will indicate that one of the cut-sets will be at least as strong as the average. Let us define a set
\[ \Omega(R) := \{ \cup_{i \in R} V_i \} \],
and observe that \( E_{\Omega(R)} \) is a subset of \( F_m \). Let us consider the partitioned cut-set termed \( F_{\Omega(R)} \) induced by choosing the partition induced on this subset of edges \( E_{\Omega(R)} \) by the partition on \( F_m \) by \( F_m \). Thus we restrict ourselves to choosing the cut-set and then partition the edges in the same way they were partitioned in \( F_m \).

This cut-set separates a set of demands,
\[ D_{\Omega(R)} = \{(i, j) : i \in R, j \in R^c\}, \]
and implies the following bound on the maximum concurrent flow,
\[ \lambda \left\{ \sum_{(i, j) \in S_{\Omega(R)}} D_{ij} \right\} \leq v(F_{\Omega}) \]
\[ \Rightarrow \lambda \left\{ \sum_{(i, j) \in S_{\Omega(R)}} D_{ij} \right\} \leq v(F_m), \]
since \( v(F_{\Omega}) \leq v(F_m) \).

Now let us average the cut-set bound \( \Omega(R) \) over subsets \( R \) of \( \{1, 2, ..., l\} \) of size \( \frac{l}{2} \). The total number of such subsets is \( \binom{l}{l/2} \).

This will imply the following bound,
\[ \lambda \left\{ \sum_{R \subseteq \{l:|R| = \frac{l}{2}\}} \sum_{(i, j) \in S_{\Omega(R)}} D_{ij} \right\} \leq \binom{l}{l/2} v(F_m). \]

The term inside the brackets in the LHS can be rewritten as
\[ \{\text{No. of cuts that each demand appears in}\} \sum_{(i, j) \in [l] \times [l]} D_{ij}, \]

Now this term can be further simplified using the equation,
\[ \{\text{No. of cuts per demand}\} \times \{\text{No. of demands}\} = \{\text{No. of demands per cut}\} \times \{\text{No. of cuts}\} \]
\[ \Leftrightarrow \{\text{No. of cuts per demand}\} l^2 = \binom{l}{l/2} \binom{l}{l/2}. \]
Substituting this in the averaged cut-set bound (111), we get,
\[ \lambda \left( \binom{l}{l/2} \right)^2 \sum_{(i, j) \in [l] \times [l]} D_{ij} \leq l^2 \binom{l}{l/2} v(F_m) \]
\[ \Rightarrow \lambda \sum_{(i, j) \in [l] \times [l]} D_{ij} \leq 4v(F_m). \]
Comparing this with (106), we can see that the averaged cut-set bound presents an upper bound on $\lambda$ at most four times larger than that of the corresponding vertex multi-partition bound, which is itself smaller than the original partitioned bidirected edge cut bound.

This implies that there exists an edge-partitioned cut-set, for which the sparsity is at most four times the sparsity of the given partitioned bidirected edge cut.

The following theorem follows immediately from Theorem 3 and Lemma 8:

**Theorem 4.** For a bidirected network with polymatroidal constraints, with $k$ source-destination pairs, the ratio between the sparsest edge-partitioned cut-set and the maximum concurrent flow is $O(\log^3 k)$.

### 4.2 Connection between Rate Regions

Since, for every value of demands $D_1, ..., D_k$, the maximum concurrent flow is within a factor $O(\log^3 k)$ of the edge-partitioned cut-set bound, the following result follows:

**Theorem 5.** For a bidirected network with polymatroidal constraints, with $k$ source-destination pairs, the rate region achievable by the flow is within a factor $O(\log^3 k)$ of the rate region defined by the edge-partitioned cut-set bounds.

### References


