The Projected Pairwise Multicommodity Flow Polyhedron

M. Queiroz and C. Humes, Jr.
Department of Computer Science
University of São Paulo
Caixa Postal 66.281, 05315-970, São Paulo - SP, Brazil
<mq@humes@ime.usp.br

(Received and accepted April 2000)

Communicated by M. Iri

Abstract—The object of this paper is a simple characterization of the vertices and extreme rays of the undirected multicommodity flow polyhedron \( P \), which has one variable for each channel. This polyhedron is seen as a linear projection of the high-dimensional directed multicommodity flow polyhedron \( F \), which has variables corresponding to each possible tuple of the form (channel, direction, origin, destination). Along with the characterization of vertices and rays of \( F \), a computationally verifiable necessary condition for the vertices is given. It is shown that no polynomially bounded analytical description of \( F \) exists, by exhibiting exponentially many facets of \( F \). © 2001 Elsevier Science Ltd. All rights reserved.

Keywords—Optimization, Multicommodity flows, Projected polyhedra.

1. INTRODUCTION

The motivation for the results here presented was the need, for design purposes, of characterizing the polyhedron of actual physical flows in a computer network with full-duplex channels. The physical flow in a communication channel with extremes \( k \) and \( l \) is the sum of all message flows from \( k \) to \( l \) and vice-versa, over all types of messages. The characterization of a message type (commodity) is given by its origin and destination. Technically, this is coherent with packet switching (see [1]). We shall refer to this particular type of multicommodity flow as pairwise, due to the peculiar characterization of different commodities.

The flows of individual commodities (messages with origin \( r \) and destination \( s \)) are classically modeled by flow polyhedra, both directed \( (\bar{F}^{rs}) \) and undirected \( (F^{rs}) \). The polyhedron we shall study \( (\bar{F}) \) is the sum of all undirected flow polyhedra \( F^{rs} \), and we shall refer to it as the projected pairwise multicommodity flow polyhedron (PMCF).

The PMCF appears in a wide range of network flow applications. Our motivation was the continuous capacity and flow assignment, where one wants to find routings for each pair of nodes, minimizing the cost of the channels (depending on transmission speeds, to be determined), subject to a constraint on the mean delay (caused by queueing processes), as in [2–5].

Research supported by FAPESP Grant No. 97/06227-2.

0893-9659/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved. Typeset by Amino-TeX
A natural motivation for analyzing the polyhedron PMCF is the dimensionality issue. In a computer network with $m$ channels (undirected arcs) and $n$ message processors (nodes), PMCF is $m$-dimensional, although derived from $2mn(n - 1)$-dimensional directed flows.

In general, a flow in PMCF may be obtained as sums of different single-commodity directed flows. These $2mn(n-1)$-dimensional representations shall be referred to as feasible decompositions of this flow.

We shall show that each vertex of PMCF corresponds to an unique feasible decomposition consisting of $n$ rooted trees, that has the property of coherency, to be defined later. And conversely, each flow with an unique feasible decomposition, consisting of $n$ rooted trees, must be a vertex.

It is also shown that a subset of the facets of PMCF is associated to the inequalities obtained from feasibility constraints on the minimal cuts of the graph. This implies that, in general, there is no polynomially bounded analytical representation for PMCF.

2. THE PAIRWISE MULTICOMMODITY FLOW POLYHEDRON

In order to describe the PMCF, let $G = (N, A)$ be the graph where $N = \{1, 2, \ldots, n\}$ is the set of nodes and $A = \{1, 2, \ldots, m\}$ is the set of arcs. For each commodity, i.e., pair of nodes $(r, s)$, the flow demand is given by $q_{rs}$. For the motivating problem, computer networks, this demand is typically measured in KBytes/sec. We shall assume for simplicity that $q_{rs} > 0, \forall r, s$.

For directed flows, we shall use the same notation $G = (N, A)$, though in this case $|A| = 2m$, since every undirected arc $\{k, l\}$ begets two directed arcs $(k, l)$ and $(l, k)$. The context will always make clear whether we are considering directed or undirected flows.

For each commodity $(r, s)$, we define the polyhedron of directed flows from $r$ to $s$ in the usual manner

$$\bar{F}^{rs} = \left\{ f^{rs} \in \mathbb{R}^{2m}_+ \mid J f^{rs} = q_{rs} [e^r - e^s] \right\},$$

where $J$ denotes the (unreduced) incidence matrix of the directed graph $G$, and $e^j$ denotes the $j$th element of the canonical basis.

The polyhedron of directed flows leaving node $r$ is given by

$$\tilde{f}^{r} - \sum_{s \neq r} \bar{F}^{rs} = \left\{ \left( f^r - \sum_{s \neq r} f^{rs} \right) \mid f^{rs} \in \tilde{F}^{rs}, \forall s \neq r \right\}. $$

Analogously, the polyhedron of directed multicommodity flows is given by

$$\tilde{F} = \sum_{r \in N} \tilde{f}^{r} = \left\{ f = \sum_{r \in N} f^r \mid f^r \in \tilde{F}^r, \forall r \in N \right\}. $$

Using the previous notation, PMCF is defined by

$$\mathcal{F} = \left\{ f \in \mathbb{R}^m_+ \mid f_e = f_{kl} + f_{lk} \text{, for } e = \{k, l\} \text{ and } f \in \tilde{F} \right\}. $$

We denote the linear operator that carries $\tilde{F}$ onto $\mathcal{F}$ by $P$. It is easy to see that $P$ is indeed a projection map. When necessary, we shall refer to the polyhedra $\bar{F}^{rs} = P(\bar{F}^{rs})$ and $F^r = P(\tilde{F}^r)$.

The directed flow polyhedra $\bar{F}^{rs}$ and $\tilde{F}^r$ are extensively studied in the literature. See, for example, [6,7]. We summarize the main results about them in Lemma 2.1. From now on, we will allow ourselves to say "the tree (path) $f^r$" if the support of the flow $f^r$ is a tree (path) and to denote the cone associated with the polyhedron $P$ by $C(P)$.

**Lemma 2.1.** The following characterizations hold:

$$V \left( \bar{F}^{rs} \right) = \left\{ f^{rs} \in \bar{F}^{rs} \mid f^{rs} \text{ is a directed path from } r \text{ to } s \right\} \neq \emptyset;$$

$$V \left( \tilde{F}^r \right) = \left\{ f^r \in \tilde{F}^r \mid f^r \text{ is a directed tree with root } r \right\} \neq \emptyset;$$

$$C \left( \tilde{F} \right) = C \left( \bar{F}^{rs} \right) = C \left( \tilde{F}^r \right) = \left\{ h \in \mathbb{R}^{2m}_+ \mid h = \sum_{i=1}^k \lambda_i h^i, \lambda \geq 0 \text{ e } h^i \in C' \right\};$$
where
\[ C' = \{ h \in \mathbb{R}_+^{2m} \mid h \text{ is a directed cycle in } G \} . \]

For the study of the multicommodity flow polyhedron, we shall use the concept of a feasible decomposition of a flow \( f \in \mathcal{F} \) with respect to components \( f^r \in \tilde{\mathcal{F}}^r \) as follows.

**DEFINITION 2.2.** The vector \((f^1, \ldots, f^n) \in \prod_{r \in N} \tilde{\mathcal{F}}^r \) is a feasible decomposition of \( f \in \mathcal{F} \) if \( f = \sum_{r \in N} f^r \). For the undirected case, \((f^1, \ldots, f^n) \in \prod_{r \in N} \tilde{\mathcal{F}}^r \) is a feasible decomposition of \( f \in \mathcal{F} \) if \( f = \mathcal{P}(\sum_{r \in N} f^r) \). Note that \( f^1, \ldots, f^n \) are directed flows.

We denote by \( \tilde{\mathcal{F}} \subseteq \mathcal{F} \) and \( 
\mathcal{F} \subseteq \mathcal{F} \) the set of flows with an unique feasible decomposition.

We often abuse notation and refer to the feasible decomposition \( f = \sum_{r \in N} f^r \), for both the directed and the undirected case.

The following lemma is a particularization of a well-known fact about linear transformations of polyhedra.

**LEMMA 2.3.** Any feasible decomposition of a vertex of \( \mathcal{F} \) or \( \mathcal{F} \) is a vector of vertices of \( \tilde{\mathcal{F}}^r \).

This corresponds to the fact that if \( f = \sum_{r \in N} f^r \) is a vertex, then for all \( r \), \( f^r \) is a directed tree with root \( r \). The converse statement is not true, i.e., there are flows with a feasible decomposition of trees that are not vertices. For instance, with \( a_{ij} = 1 \), \( \forall i, j \), the following trees do not correspond to a vertex of \( \tilde{\mathcal{F}} \) or \( \mathcal{F} \):

\[
\begin{array}{c}
\text{1}\cdot \\
2 \\
3\
\end{array} \quad \begin{array}{c}
\text{2}\cdot \\
2 \\
3\
\end{array} \quad \begin{array}{c}
\text{3}\cdot \\
2 \\
2\
\end{array}
\]

This example lacks one necessary condition of being a vertex, namely the property of coherency (to be defined later). This property roughly demands that within the same feasible decomposition any directed (descending) paths from \( i \) to \( j \) be the same, for all \( (i, j) \). Clearly this does not happen in the example above.

We denote by \( \mathcal{T} \subseteq \mathcal{F} \) and \( \mathcal{T} \subseteq \mathcal{F} \) the flows that have some feasible decomposition of trees.

It is natural to search for conditions that are either necessary or sufficient to ensure that a particular sum of rooted trees \( f^r \in V(\tilde{\mathcal{F}}^r) \) is indeed a vertex of \( \tilde{\mathcal{F}} \). The following result shows that \( f \in \tilde{\mathcal{F}} \) is a vertex if and only if \( f \) has an unique feasible decomposition which is formed by trees.

**THEOREM 2.4.** \( V(\tilde{\mathcal{F}}) = \mathcal{T} \cap \tilde{\mathcal{U}} \).

**PROOF.**

(\( \subseteq \)) Let \( f \in V(\tilde{\mathcal{F}}) \). By Lemma 2.3, \( f \in \tilde{T} \). Assume that \( f \) has two feasible decompositions \( f = \sum g^i = \sum h^i \). Then \( f = \sum (1/2)g^i + (1/2)h^i \) and, by Lemma 2.3, \( (1/2)g^i + (1/2)h^i \) are trees. But this implies that \( g^i = h^i \), and therefore, \( f \) has an unique decomposition.

(\( \supseteq \)) Let \( f \in \tilde{T} \cap \tilde{\mathcal{U}} \). If \( f = (1/2)g + (1/2)h \), then \( f = \sum (1/2)g^i + (1/2)h^i \). As, by assumption, \( f \) has an unique decomposition which is formed by trees, then \( (1/2)g^i + (1/2)h^i \) are trees, and therefore, \( g^i = h^i \), and \( f \) is a vertex.

The same proof applies to the following undirected case.

**COROLLARY 2.5.** \( V(\mathcal{F}) = \mathcal{T} \cap \mathcal{U} \).

Note that uniqueness of decomposition alone is not sufficient to ensure that \( f \in \tilde{\mathcal{F}} \) is a vertex. The following example corresponds to a flow that is not a vertex of \( \tilde{\mathcal{F}} \), but has an unique decomposition.
Actually a stronger result is available for the undirected case. It is a straightforward consequence of the following lemma.

**Lemma 2.6.** $U \subset T$. 

**Proof.** Suppose that $f \in \mathcal{F}$ has an unique decomposition $(f^1, \ldots, f^n)$ but $f^i$ is not a tree. It is nevertheless a sum of paths from $i$ to the other nodes (no cycling of messages is possible because of uniqueness of decomposition). In this case, there must be two distinct paths $\Pi^i_j$ and $\Pi^j_k$ carrying flow from $i$ to the same node $j$. Let $\Pi$ be some path from $j$ to $i$ in $f^j$. Without loss of generality, and $g^i = f^i - \epsilon \Pi_1 + \epsilon \Pi, g^j = f^j + \epsilon \Pi_1 - \epsilon \Pi, g^k = f^k, \forall k \neq i, j$ is another feasible decomposition for $f$, for $\epsilon$ sufficiently small.

**Corollary 2.8** and **Lemma 2.6** together imply the following.

**Theorem 2.7.** $V(\mathcal{F}) = U$.

Though $V(\mathcal{F}) \subseteq \mathcal{P}(V(\mathcal{F}))$ obviously holds, we must note that there are vertices of $\mathcal{F}$ whose projection is not a vertex of $\mathcal{F}$. The simplest example is

<table>
<thead>
<tr>
<th>$f^1$</th>
<th>$f^2$</th>
<th>$f^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 1</td>
<td>2 3 2</td>
<td>3 1 2</td>
</tr>
</tbody>
</table>

that has an unique decomposition as a directed flow, but has other decompositions as an undirected flow.

Theorem 2.7 has a very interesting computational implication. Though, in general, $\mathcal{F}$ has no polynomially-sized analytical description, the question whether $f$ belongs to $V(\mathcal{F})$ can be answered in time which is polynomial in $n$, the number of nodes.

**Corollary 2.8.** There exists an algorithm, with running time polynomial in $n$, that decides whether $f \in V(\mathcal{F})$ or not.

**Proof.** Due to Theorem 2.7, $f \in V(\mathcal{F})$ if and only if it has an unique feasible decomposition. This is equivalent to the fact that there is only one solution to the linear system below,

$$x^r \in \tilde{\mathcal{F}}, \quad r \in N,$$

$$\mathcal{P}(x) = f,$$

whose dimensions are polynomial in $n$. As a polyhedron $X \subset \mathbb{R}^p$ has an unique feasible point if and only if all the $2p$ linear programming problems $\max_{x \in X} \langle \phi^i, x \rangle$ and $\min_{x \in X} \langle \phi^j, x \rangle$, for $i = 1, \ldots, p$, have the same solution, the algorithm corresponds to perform the above test on the polyhedron defined by the equations above. Since solving a linear programming problem requires a polynomial number of operations, the overall construction leads to a polynomial algorithm for verifying if $f \in V(\mathcal{F})$.

Though polynomially realizable, this verification of uniqueness of decomposition is a costly operation. A very handy and easily verifiable property of vertices, that was originally motivated
by the well-known characterization of vertices as unique minimizers of some linear function, is 
coherency, defined below.

**Definition 2.9.** A flow \( f \in \overline{T} \) is **coherent** if it has a feasible decomposition \( (f^1, \ldots, f^n) \) such 
that for each pair of nodes \((i, j)\), every directed path from \( i \) to \( j \) in the trees \( f^1, f^2, \ldots, f^n \) is 
the same. If, in addition, every directed path from \( j \) to \( i \) is the reverse path from \( i \) to \( j \), \( f \) is 
strongly coherent.

**Theorem 2.10.** If \( f \in V(\overline{F}) \), then \( f \) is coherent.

**Proof.** By Theorem 2.4, \( f \) has an unique decomposition of trees. Suppose, for the sake of 
contradiction, that \( f \) is not coherent. Then there must be two distinct paths \( \Pi^1 \) and \( \Pi^2 \) carrying 
flow from \( i \) to \( j \). Then, for \( \epsilon \) sufficiently small, \( g = f - \epsilon \Pi^1 + \epsilon \Pi^2 \) and \( h = f + \epsilon \Pi^1 - \epsilon \Pi^2 \) 
are distinct feasible flows such that \( f = (1/2)g + (1/2)h \). This contradicts the fact that \( f \) is a 
vertex. \( \square \)

**Theorem 2.11.** If \( f \in V(\overline{F}) \), then \( f \) is strongly coherent.

**Proof.** By Corollary 2.5, \( f \) has an unique decomposition of trees, and this decomposition is 
coherent, since the directed flow \( f' \) associated to \( f \) is a vertex of \( \overline{F} \). If \( f \) were not strongly coherent, 
there should be two paths \( \Pi^1 \) and \( \Pi^2 \), one not being the reverse of the other, carrying flow from \( i \) 
to \( j \) and from \( j \) to \( i \), respectively. In this case, for \( \epsilon \) sufficiently small, \( g = P(f^1 - \epsilon \Pi^1 + \epsilon (\Pi^2)^{-1}) \) 
and \( h = P(f^1 + \epsilon \Pi^2 - \epsilon (\Pi^1)^{-1}) \) would satisfy \( f = (1/2)g + (1/2)h \) and \( g \neq h \). \( \square \)

The above necessary conditions are not sufficient, as can be checked from the following example. 
In a 6-node graph with edges \((13, 14, 23, 24, 35, 36, 45, 46)\), the following decomposition 
corresponds to a strongly coherent flow that is not a vertex of \( \overline{F} \).

We now present a partial description of the set of facets of \( \overline{F} \), associated with minimal cuts 
in \( \mathcal{G} \). The description is not complete, but it suffices to show that the number of facets grows 
superexponentially with the number of nodes. Let \( \delta(S) = \{e \mid e = \{k, l\}, k \in S, l \in N \setminus S\} \).

**Proposition 2.12.** Let \( S_1, S_2, \ldots, S_k \) be all the subsets of \( N \) such that \( \emptyset \neq S_i \neq N \) and \( \delta(S_i) \) be 
a minimal cut in the graph \( \mathcal{G} \). Then

\[
\mathcal{F} \subseteq \left\{ x \in \mathbb{R}^m_+ \left| \sum_{e \in \delta(S_i)} x_e \geq \sum_{r \in S_i, s \in N \setminus S_i} q_{rs} + q_{sr}, \ i = 1, \ldots, k \right. \right\}.
\]

Moreover, all these inequalities define facets of \( \mathcal{F} \).

**Proof.** That the inequalities above are all valid for \( \mathcal{F} \) is a straightforward consequence of a 
well-known theorem by Iri [8], also proved by Onaga and Kakusho [9], known as the Japanese 
Theorem (see also, for instance, the survey by Avis and Deza [10]).

To show that each of these inequalities is a facet of \( \mathcal{F} \), we show that for each cut \( \delta(S_i) \), there 
are \( m + 1 \) points \( x_0, \ldots, x^m \) of the polyhedron, lying on the hyperplane defined by the inequality 
associated with \( S_i \), and such that \( \{x^1 - x_0, \ldots, x^m - x_0\} \) are linearly independent.

Let \( i \in \{1, \ldots, k\} \), and suppose \( \delta(S_i) = \{e_0, \ldots, e_l\} \). For each arc \( e_j \), choose some tree that 
uses \( e_j \) but no other arc in \( S_i \) (if there were no such tree, the cut would not be minimal). Let \( x^j \) be 
the multicommodity flow corresponding to routing every message through the arcs of this 
tree. Thus, we construct \( l \) points satisfying the constraint of \( S_i \) with equality. For each arc
e_j \not\in \delta(S_i), we construct points of the form x^j = x^0 + M e_j, choosing M > 0 large enough such that \{x^0, \ldots, x^m\} be linearly independent. Each of these points satisfies the constraint of S_i with equality, and this completes the proof.

Since the number of minimal cuts may grow superexponentially with the number of nodes, as in the case of complete n-node graphs, there cannot be a computationally feasible upper bound for the number of cuts in the general case.

3. CONCLUSION

We have given a characterization of the vertices and extreme rays of the projected pairwise multicommodity flow polyhedron, as well as a computationally verifiable necessary condition for the vertices. These results revealed the importance of the concepts of uniqueness of decomposition, coherency and strong coherency for the characterization of PMCF, through the analysis of the projection P. For applications of these results to the Capacity and Flow Assignment Problem, we refer the reader to [5].

REFERENCES