1 Introduction

A network may be shared by many different services or commodities. For example:

- telephone network: each call is a commodity with its own source and sink
- transportation network: each trip is its own commodity
- supply chain: different goods move through the network

The network can be directed or undirected. We’ll assume it is a directed network $D = (V, E)$ and let $A$ denote the node-arc incidence matrix. We have $K$ commodities, each with its own sources and sinks. The vector $b^k$ gives the net demand at each node for commodity $k$. Transporting one unit of commodity $k$ on arc $(i, j) \in E$ costs $c^k_{ij}$.

Each arc $(i, j) \in E$ has a capacity $u_{ij}$. The total amount shipped through arc $(i, j) \in E$ of all the commodities must be no larger than $u_{ij}$. Without this shared bound, the problem would separate: we could solve separate min cost circulation problems for each commodity.

We’ll give two formulations: an arc-based formulation and a path-based formulation. Each formulation can be modified straightforwardly to handle the case that the network is undirected. For more details, see for example Chapter 17 of the text by Ahuja et al [1].

2 Arc-based formulation

We let $x^k_{ij}$ denote the amount of commodity $k$ shipped on arc $(i, j) \in E$. The arc-based LP formulation for the multicommodity network flow problem is:

$$\begin{align*}
\min_{x \in \mathbb{R}^{|K| \times K}} & \quad \sum_{k=1}^{K} \sum_{(i,j) \in E} c^k_{ij} x^k_{ij} \\
\text{subject to} & \quad Ax^k = b^k \\
& \quad \sum_{k=1}^{K} x^k_{ij} \leq u_{ij} \quad \forall (i, j) \in E \\
& \quad x \geq 0
\end{align*}$$
Even if all the data is integer, this problem may well have a fractional optimal solution, in contrast with the single commodity case.

Let \( n = |E| \) and \( m = |V| \). The arc-based formulation has \( nK \) variables and \( n + mK \) constraints (excluding nonnegativity constraints). For example, in a graph with 1000 nodes and 5000 arcs and with one commodity for each pair of nodes, we have on the order of \( 10^6 \) variables and constraints. In principle, the problem can be solved using an LP solver, but in practice the size of the problem can be daunting.

3 A path-based formulation

We assume all the costs \( c^k_{ij} \) are nonnegative, so in particular there are no negative length cycles. Any feasible flow in commodity \( k \) can be represented as a sum of flows on paths. We assume each commodity has a single source \( s^k \) and a single sink \( t^k \); this can be done without loss of generality, by introducing a supersource and/or a supersink for each commodity and constructing capacities and costs appropriately for the edges linking the supersource to the sources and the sinks to the supersink.

We let \( d^k \in \mathbb{R} \) denote the (scalar) demand for commodity \( k \) and we let \( P^k \) denote the set of all paths from the source for commodity \( k \) to the sink for commodity \( k \). Each path \( p \in P^k \) has a cost per unit shipped:

\[
c^k_p = \sum_{(i,j) \in p} c^k_{ij}.
\]

We define an indicator parameter to capture which edges are on a path \( p \):

\[
\delta_{ij}(p) = \begin{cases} 
1 & \text{if } (i,j) \in p \\
0 & \text{otherwise}
\end{cases}
\]

Our variables are \( f^k_p \) for each \( p \in P^k \), for each \( k = 1, \ldots, K \), which gives the amount of commodity \( k \) shipped on path \( p \). The path-based formulation of the multicommodity problem is then:

\[
\begin{align*}
\min_f & \quad \sum_{k=1}^K \sum_{p \in P^k} c^k_p f^k_p \\
\text{subject to} & \quad \sum_{k=1}^K \sum_{p \in P^k} \delta_{ij}(p) f^k_p \leq u_{ij} \quad \forall (i,j) \in E \\
& \quad \sum_{p \in P^k} f^k_p = d^k \quad \text{for } k = 1, \ldots, K \\
& \quad f^k_p \geq 0 \quad \forall p \in P^k, k = 1, \ldots, K
\end{align*}
\]

The path-based formulation has \( n + K \) constraints (excluding nonnegativity constraints). For the earlier numbers of 1000 nodes and 5000 arcs and with one commodity for each pair of nodes, this results in on the order of \( 10^6 \) constraints. The drawback is that the path-based formulation has an exponential number of variables. Thus, the paths are not all enumerated \textit{a priori}; instead, they are generated as needed. They can be found by solving shortest path problems.

References