Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

ANDREAS BLEY

An Integer Programming Algorithm for Routing Optimization in IP Networks

ZIB-Report 08-30 (July 2008)

An Integer Programming Algorithm for Routing Optimization in IP Networks

Andreas Bley

Zuse Institute Berlin
Takustr. 7, D-14195 Berlin, Germany
bley@zib.de

Abstract Most data networks nowadays use shortest path protocols to route the traffic. Given administrative routing lengths for the links of the network, all data packets are sent along shortest paths with respect to these lengths from their source to their destination.
In this paper, we present an integer programming algorithm for the minimum congestion unsplittable shortest path routing problem, which arises in the operational planning of such networks. Given a capacitated directed graph and a set of communication demands, the goal is to find routing lengths that define a unique shortest path for each demand and minimize the maximum congestion over all links in the resulting routing. We illustrate the general decomposition approach our algorithm is based on, present the integer and linear programming models used to solve the master and the client problem, and discuss the most important implementational aspects. Finally, we report computational results for various benchmark problems, which demonstrate the efficiency of our algorithm.
Keywords: Shortest Path Routing, Integer Programming

1 Introduction

In this paper, we present an integer programming algorithm to optimize the routing in communication networks based on shortest path routing protocols such as OSPF [22] or IS-IS [16], which are widely used in the Internet. With these routing protocols, all end-to-end traffic streams are routed along shortest paths with respect to some administrative link lengths (or routing weights), that form the so-called routing metric. Finding a routing metric that induces a set of globally efficient end-to-end routing paths is a major difficulty in such networks. The shortest path routing paradigm enforces rather complicated and subtle interdependencies among the paths that comprise a valid routing. The routing paths can be controlled only jointly and only indirectly via the link lengths. In this paper, we consider the unsplittable shortest path routing variant, where the lengths must be chosen such that the shortest paths are unique and each traffic stream is sent unsplit via its single shortest path.

One of the most important operational planning tasks in such networks is traffic engineering. Its goal is to improve the service quality in the existing network by (re-)optimizing the routing of the traffic, but leaving the network topology and hardware configuration unchanged. Mathematically, this can be formulated as the minimum congestion unsplittable shortest path routing problem (MIN-CON-USPR). The problem input consists of a digraph $D = (V, A)$ with arc capacities $c_a \in \mathbb{Z}$ for all $a \in A$, and a set of directed commodities $K \subseteq V \times V$ with demand values $d_{st} \in \mathbb{Z}$ for all $(s,t) \in K$. A feasible solution is an unsplittable shortest path routing (USPR) of the commodities, i.e., a metric of link lengths $w_a \in \mathbb{Z}$, $a \in A$, that induce a unique shortest (s,t)-path for each commodity $(s,t) \in K$. Each commodity's demand is sent unsplit along its shortest path. The objective is to minimize the maximum congestion (i.e., the flow to capacity ratio) over all arcs. The maximum congestion is a good measure and typically used as a key indicator for the overall network service quality.

Due to their great practical relevance, shortest path routing problems have been studied quite intensively in the last decade. Ben-Ameur and Gourdin [3], Broström and Holmberg [13,14] studied the combinatorial properties of path sets that correspond to shortest (multi-)path routings and devised linear programming models to find lengths that induce a set of presumed shortest paths (or prove that no such lengths exist). Bley [5,9], on the other hand, showed that finding a smallest shortest-path conflict in a set of presumed shortest paths or the smallest integer lengths inducing these paths is $\mathcal{NP}$-hard. Bley [6,7] also proved that MIN-CON-USPR is inapproximable within a factor of $\Omega(|V|^{1-\epsilon})$ for any $\epsilon > 0$, presented examples where the smallest link congestion that can be obtained with unsplittable shortest path routing exceeds the congestion that can be obtained with multicommodity flow or unsplittable flow routing by a factor of $\Omega(|V|^2)$, and proposed polynomial time approximation algorithms for several special cases of MIN-CON-USPR and related network design problems. The minimum congestion shortest multi-path routing problem has been shown to be inapproximable within a factor less than 3/2 by Fortz and Thorup [18].

Various approaches for the solution of network design and routing problems in shortest path networks have been proposed. Algorithms using local search, simulated annealing, or Lagrangian relaxation techniques with the routing lengths as primary decision variables are presented in [4,10,15,17,18], for example. These length-based methods work well for shortest multi-path routing problems, where traffic may be split among several equally long shortest paths, but they often produce only suboptimal solutions for hard unsplittable shortest path routing problems. As they deliver no or only weak quality guarantees, they cannot guarantee to find provenly optimal solutions.

Using mixed integer programming formulations that contain variables for the routing lengths as well as for the resulting shortest paths and traffic flows, shortest path routing problems can – in principle – be solved to optimality. Formulations of this type are discussed in [10,19,24,26,29], for example. Unfortunately, the relation between the shortest paths and the routing length always leads to quadratic or very large big-M models, which are computationally extremely hard and not suitable for practical problems.

In this paper, we present an integer programming algorithm that decomposes the routing problem into the two tasks of first finding the optimal end-to-end routing paths and then, secondly, finding a routing metric that induce these paths. As we will show, this approach permits the solution of real-world problems. An implementation of this algorithm [11,9] is used successfully in the planning of the German national education and research network for several years. Variants of this decomposition approach for shortest multi-path and shortest path multicast routing problems are discussed in [12,20,27,28,29].

The remainder of this paper is organized as follows. In Section 2, we formally define the problem addressed in this paper and introduce the basic notion and notation. The overall decomposition algorithm, the integer and linear programming models and sub-algorithms used for the solution of the master and the client problem, and the most important aspects of our implementation are described in Section 3. In Section 4, we finally report on numerical results obtained with this algorithm for numerous real-world and benchmark problems and illustrate the relevance of optimizing the routing in practice.

2 Notation and Preliminaries

Let $D = (V, A)$ be a directed graph with arc capacities $c_a \in \mathbb{Z}$ for all $a \in A$ and let $K \subseteq V \times V$ be a set of directed commodities with demand values $d_{st} \in \mathbb{Z}$ for all $(s, t) \in K$. A metric $\mathbf{w} = (w_a) \in \mathbb{Z}^A$ of arc lengths is said to define an *unsplittable shortest path routing (USPR)* for the commodities K, if the shortest (s, t)-path P^*_{st} with respect to $\mathbf{w}$ is uniquely determined for each commodity $(s, t) \in K$. The demand of each commodity is routed unsplit along the respective shortest path. For a metric $\mathbf{w}$ that defines such an USPR, the total flow through an arc $a \in A$ then is

$$f_a(\mathbf{w}) := \sum_{(s,t)\in K:\, a\in P^*_{st}(\mathbf{w})} d_{st} \ . \tag{1}$$

The task in the minimum congestion unsplittable shortest path routing problem Min-Con-USPR is to find a metric $\mathbf{w} \in \mathbb{Z}^A$ that defines an USPR for the given commodity set K and minimizes the maximum congestion $L := \max\{f_a(\mathbf{w})/c_a : a \in A\}$.

Before presenting of our algorithm, we need to introduce some further notation. We say that a metric $\mathbf{w}$ is *compatible* with a set $\mathcal{P}$ of end-to-end routing paths, if each path $P \in \mathcal{P}$ is the unique shortest path between its terminals with respect to $\mathbf{w}$. A metric $\mathbf{w}$ is said to be *compatible* with set of node-arc pairs $F \subset V \times A$, if arc a is on a unique shortest path towards t for all $(t, a) \in F$. If there exists such a metric, we say that the set F is a valid *unique shortest path forwarding (USPF)*, otherwise we call it an *(USPF-) conflict*. One easily verifies that a metric is compatible with a path set $\mathcal{P}$ if and only if it is compatible with the set of node-arc pairs $F := \bigcup_{P\in\mathcal{P}}\{(t, a) : t \text{ is destination of } P,\ a \in P\}$.

Clearly, any subset (including the empty set) of an USPF is an USPF as well. Hence, the family of all USPF in the digraph D forms an independence system (or hereditary family) $\mathcal{I} \subset 2^{V\times A}$. The circuits of this independence system are exactly the irreducible conflicts. The family of all irreducible conflicts is denoted by $\mathcal{C} \subset 2^{V\times A}$.

In general, these set families can be extremely complex and computationally intractable [9]. Given an arbitrary set $F \subset V \times A$, the smallest conflict (with respect to the number of node-arc pairs) in F may be arbitrarily large and even approximating its size within a factor less than $7/6$ is $\mathcal{NP}$-hard. Approximating the size of the largest valid USPF in F within a factor less than $8/7$ is $\mathcal{NP}$-hard as well. However, one can decide in polynomial time whether or not a given set $F \subset V \times A$ is a valid USPF and, depending on that, either find a compatible metric or some (not necessarily minimal) irreducible conflict in F, which is the foundation of the algorithm described in this paper.

3 Integer Programming Algorithm

Similar to Bender's decomposition, our algorithm decomposes the problem of finding an optimal shortest path routing into the master problem of finding the optimal end-to-end paths and the client problem of finding compatible routing lengths for these paths.

The master problem is formulated as an integer linear program and solved with a branch-and-cut algorithm. Instead of using routing weight variables, the underlying formulation contains special inequalities to exclude routing path configurations that are no valid unsplittable shortest path routings. These inequalities are generated dynamically as cutting planes by the client problem during the execution or the branch-and-cut algorithm.

Given a set of routing paths computed by the master problem's branch-and-cut algorithm, the client problem then is to find a metric of routing lengths that induce exactly these paths. As we will see in Section 3.2, this problem can be formulated and solved as a linear program. If the given paths indeed form a valid shortest path routing, the solution of this linear program yields a compatible metric. If the given paths do not form a valid unsplittable shortest path routing, the client linear program is infeasible. In this case, the given routing paths contain a conflict that must not occur in any admissible shortest path routing. This conflict, which can be derived from the dual solution of the infeasible client linear program, then can be turned into an inequality for the master problem, which is valid for all admissible shortest path routings, but violated by the current routing. Adding this inequality to the master problem, we then cut off the current non-admissible routing and proceed with the master branch-and-cut algorithm to compute another candidate routing.

3.1 Master Problem

There are several ways to formulate the master problem of Min-Con-USPR as a mixed integer program. For notational simplicity, we present a variation of the disaggregated arc-routing formulation used in our algorithm, which contains additional artificial variables that describe the unique shortest path forwarding defined by the routing.