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.

The primary decision variables used in this formulation are the variables $x_a^{st} \in \{0,1\}$ for all $(s,t) \in K$ and $a \in A$. These variables describe which arcs are contained in the routing paths. Variable x_a^{st} is supposed to be 1 if and only if arc a is contained in the routing path for commodity (s,t). A single variable $L \in \mathbb{R}$ represents the maximum congestion that is attained by the routing. The additional artificial variables $y_a^t \in \{0,1\}$ for all $t \in V$ and $a \in A$ describe the forwarding defined by the routing paths. Variable y_a^t is supposed to be 1 if there is a routing path towards t that contains arc a. With these variables the master problem of Min-Con-USPR can be formulated as follows:

$$\min \quad L \tag{2a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} x_a^{st} - \sum_{a \in \delta^-(v)} x_a^{st} = \begin{cases} -1 & \text{if } v = s \\ 1 & \text{if } v = t \\ 0 & \text{else} \end{cases} \qquad (s,t) \in K,\, v \in V \tag{2b}$$

$$\sum_{(s,t) \in K} d_{st} x_a^{st} \leq c_a\, L \qquad a \in A \tag{2c}$$

$$x_a^{st} \leq y_a^t \qquad (s,t) \in K,\, a \in A \tag{2d}$$

$$\sum_{a \in \delta^+(v)} y_a^t \leq 1 \qquad t \in V,\, v \in V \tag{2e}$$

$$\sum_{(a,t) \in C} y_a^t \leq |C| - 1 \qquad C \in \mathcal{C} \tag{2f}$$

$$x_a^{st} \in \{0,1\} \qquad (s,t) \in K,\, a \in A \tag{2g}$$

$$L \geq 0. \tag{2h}$$

Subproblem (2a)–(2c) together with the integrality and non-negativity constraints (2g) and (2h) is a standard arc-routing formulation for the unsplittable multicommodity flow problem, whose objective is to minimize the congestion L.

Inequalities (2d) force the artificial variables y_a^t to be (at least) 1 for all arcs a that are contained in some routing path towards destination t. Together with the out-degree constraints (2e) this ensures that, for each destination $t \in V$, the routing paths towards t form an anti-arborescence (a reversely oriented tree). This is clearly necessary for the paths in any valid unsplittable shortest path routing.

Constraints (2f) finally ensure that no integer solution of (2) contains all node-arc pairs of any (irreducible) USPF-conflict $C \in \mathcal{C}$. As the irreducible conflicts are exactly the circuits of the independence system formed by all valid unique shortest path forwarding, this implies that the artificial variables y_a^t describe a valid USPF. Consequently, the routing given by any integer feasible solution of (2) is a valid unsplittable shortest path routing. In general, the number of these conflict constraints (2f) can be exponentially large. They are separated via the client problem during the branch-and-cut solution process.

Note that the model contains no explicit constraints forcing the artificial variables y_a^t to attain only values 0 or 1. These constrains are not necessary. Any solution $(\mathbf{x}, \mathbf{y}, L)$ with $x_a^{st} \in \{0,1\}$ for all $(s,t) \in K$ and $a \in A$ can be easily turned into an equivalent solution with $y_t^a \in \{0,1\}$ for all $t \in V$ and $a \in A$ by setting $y_t^a := \max\{x_a^{st} : s \text{ with } (s,t) \in K\}$ for all t and a.

3.2 Client Problem

Now suppose we are given an integer solution $(\mathbf{x}, \mathbf{y}, L)$ of formulation (2) or, more precisely, of a subsystem of (2) containing only some of the conflict constraints (2f) so far.

Let F be the presumed unique shortest path forwarding given by this solution, i.e., $F = \{(t,a) : y_a^t = 1\}$. Our goal in the client problem is to find a compatible metric $\mathbf{w}$ for F. However, if the given solution $(\mathbf{x}, \mathbf{y}, L)$ violates some of the conflict constraints (2f) that have not yet been added to the master formulation, such a metric does not exist. In this case, the task is to generate one of these violated inequalities.

The first part of this problem can be solved with linear programming techniques. A number of alternative formulations for this so-called *inverse shortest paths problem (ISP)* have been proposed in the literature [3,25]. In the following, we present the aggregated formulation used in our algorithm together with the arc-routing formulation for the master problem.

Let F be the given presumed unique shortest path forwarding. For each pair $(t,a) \in F$, arc $a = (u,v)$ is assumed to be on a *unique* shortest path from u to t. Hence, the arcs $a' \in \delta^+(u)\backslash\{a\}$ must not be on any shortest (u,t)-path. The set of all implied non shortest path node-arc pairs is $\bar{F} = \bigcup_{(t,(u,v)) \in F} (\delta^+(u) \setminus \{(u,v)\})$: For each pair $(t,a) \in \bar{F}$, arc $a = (u,v)$ must not be on a shortest path from u to t. (Note that we cannot simply assume $\bar{F} = V \times A \setminus F$, because F not necessarily prescribes the shortest paths between all node pairs. Arcs that are not relevant for the routing of the given commodities may or may not be on shortest paths.)

Our formulation of the inverse shortest paths problem uses a variable $w_a \in \mathbb{Z}$ for the length of each arc $a \in A$ and a variable $r_v^t \in \mathbb{R}$ for the potential of each node $v \in V$ with respect to each destination $t \in V$ and the metric $\mathbf{w}$. (If $r_t^t = 0$, the smallest possible potential r_v^t of node v is exactly the distance from v to t with respect to the arc lengths w_a.) With these variables, the inverse shortest paths problem for the given forwarding F, can be formulated as follows:

$$
\begin{aligned}
\min \quad & w_{\max} && && \text{(3a)}\\
\text{s.t.} \quad & w_{(u,v)} - r_u^t + r_v^t = 0 && (t,(u,v)) \in F && \text{(3b)}\\
& w_{(u,v)} - r_u^t + r_v^t \geq 1 && (t,(u,v)) \in \bar{F} && \text{(3c)}\\
& w_{(u,v)} - r_u^t + r_v^t \geq 0 && (t,(u,v)) \in (V \times A) \setminus F \setminus \bar{F} && \text{(3d)}\\
& 1 \leq w_a \leq w_{\max} && a \in A && \text{(3e)}\\
& r_v^t \in \mathbb{R} && t \in V,\, v \in V && \text{(3f)}\\
& w_a \in \mathbb{Z} && a \in A\ . && \text{(3g)}
\end{aligned}
$$

Constraints (3b),(3d), and (3e) ensure that the lengths w_a in any solution of (3) form a compatible metric for the given forwarding F. The term $w_{(u,v)} - r_u^t + r_v^t$

is the difference between the length of the shortest path starting in node u, passing through arc (u, v), and ending in node t, and the distance from node v to node t. This difference must be 0 for all arcs (u, v) that are on a shortest path and strictly greater than 0 for all arcs that must not be on a shortest path, as expressed in constraints (3b) and (3c). For all remaining arcs it must be non-negative. Formulation (3) has a solution if and only if there exist a compatible metric for the given forwarding F. Furthermore, there is a compatible metric with lengths in the range $\{1, 2, \ldots, M\}$ if and only if the optimal solution value $w_{\max}$ of formulation (3) is less or equal to M.

Note that formulation (3) is an integer program and may be computationally hard. In fact, Bley [8] proved that it is already $\mathcal{NP}$-hard to approximate its optimum value within a factor less than 9/8 in general.

In our algorithm, we solve the linear relaxation of (3) in a first step and scale and round its optimal fractional solution to an integer feasible solution of (3) afterwards. It is not difficult to verify that the integer program (3) has a solution if and only if its linear relaxation has. Using the rounding scheme proposed by Ben-Ameur and Gourdin [3], we obtain lengths that exceed the minimal ones by a factor of at most $\min(|V|/2, |P_{\max}|)$, where $P_{\max}$ is the longest prescribed shortest path. For practically relevant network sizes, the weights computed with this approximate method easily fit into the admissible range of all modern routing protocols. So, we can safely ignore the integrality constraint (3g) in practice.

If the linear relaxation of (3) is infeasible, then the given solution $(\mathbf{x}, \mathbf{y}, L)$ of the (incomplete) master formulation is not a valid routing. In this case, the presumed forwarding F is no valid unsplittable shortest path forwarding. It contains at least one (irreducible) conflict $C \in \mathcal{C}$, whose corresponding inequality (2f) is violated by the given solution $(\mathbf{x}, \mathbf{y}, L)$. To find one of these conflicts, we iteratively try to remove each node-arc pair from F. In each iteration, we remove one pair (t, a) from F, update the set $\bar{F}$ of implied non-shortest path node-arc pairs, and solve the corresponding linear relaxation of (3). If this linear program remains infeasible, we remove the pair (t, a) permanently from F. Otherwise, we reinsert it into F and keep it permanently. If no more node-arc pair can be removed, the remaining set F defines an irreducible conflict, whose corresponding conflict inequality (2f) for $C = F$ is violated by the given solution $(\mathbf{x}, \mathbf{y}, L)$. In our implementation, we improved the practical performance of this procedure significantly by removing initially all those pairs $(t, (u, v))$ from F, for which the dual variables of the corresponding constraint (3b) and the dual variables of all constraints (3c) implied by $(t, (u, v)) \in F$ are 0. If these constraints are not active in the infeasible subsystem of (3), there is at least one (irreducible) conflict that is not related to the fact that $(t, (u, v)) \in F$.

Note that this iterative method finds an irreducible conflict inequality (2f), but not necessarily the most violated one. Finding the most violated such inequality is $\mathcal{NP}$-hard, even if the given solution of the master problem is integer [9]. Furthermore, note that this approach solves the separation problem over the conflict inequalities (2f) only for integer solutions $(\mathbf{x}, \mathbf{y}, L)$. For fractional solutions $(\mathbf{x}, \mathbf{y}, L)$, the presumed forwarding F is not well-defined. A separation heuristic based on an approximate integer programming model of the separation problem (for shortest multi-path routings), which can be applied for fractional solutions, has been proposed by Tomaszewski et al. [28]. Also, several subclasses of (2f) can be separated in polynomial time; see [9,11,13,14,28,29].

3.3 Implementation

From the theoretical point of view, the branch-and-cut approach presented above seems not very attractive. The integrality gap of the integer programming formulation (2) can be very large and the separation of the conflict inequalities (2f) is $\mathcal{NP}$-hard for fractional solutions of (2). Nevertheless, implemented carefully this approach works surprisingly well in practice. In the following, we briefly discuss the most important aspects of our implementation of this algorithm. Further details, including a description of all used cutting planes and separation algorithms, of the specially tailored branching schemes, and of the problem-specific primal heuristics, can be found in [9].

In our implementation, the initial formulation of the master problem contains only the arc-routing variables x_a^{st}, the congestion variable L, and the flow conservation and capacity constraints (2b) and (2c). All other constraints are separated. The degree constraints (2e) are separated by a simple enumerative algorithm searching through all node pairs $t, v \in V$. The conflict constraints (2f) are separated via the solution of the client problem as described in the previous section. However, the artificial variables y_a^t involved in these constraints and the linking constraints (2d) are not generated explicitly. Instead, we assume

$$y_a^t := \max\{x_a^{st} : s \text{ with } (s,t) \in K\} \text{ for all } t \in V \text{ and } a \in A, \tag{4}$$

disaggregate each of the inequalities (2e) and (2f) into an equivalent set of inequalities on the arc-routing variables x_a^{st} instead of the forwarding variables y_t^a, and separate over the set of these disaggregated inequalities. This is done by applying the separation algorithms for the original inequalities to the values y_t^a defined as in (4). If a violated inequality is found, each variable y_t^a in this inequality is replaced by a variable $x_a^{st} = \arg\max\{x_a^{st} : s \text{ with } (s,t) \in K\}$, which yields one of the most violated disaggregated inequalities corresponding to the violated original inequality.

At each node of the master problem's branch-and-bound tree we solve the current LP relaxation and separate violated out-degree constraint and several other classes of inequalities.

Analogous to the out-degree inequalities (2e), which ensure that the routing paths towards each destination t form an anti-arborescence, we also separate in-degree inequalities, which ensure that the routing paths emanating from each source s form an arborescence. With the implicit, artificial variables $z_a^s := \max\{x_a^{st} : t \text{ with } (s,t) \in K\}$ for all $s \in V$ and $a \in A$, these inequalities can be easily formulated as

$$\sum_{a \in \delta^-(v)} z_a^s \leq 1 \qquad\qquad v, s \in V \ . \tag{5}$$

Using the same disaggregation approach as for the out-degree constraints, we separate the disaggregated version of these inequalities with a simple enumerative algorithm.

Numerous types of valid inequalities can be derived from the so-called Bellman property (or subpath consistency) of shortest path routings. This property basically says the following: If both terminals s_1 and t_1 of a commodity $(s_1, t_1) \in K$ are contained in the routing path of another commodity $(s_2, t_2) \in K$, then the routing path of commodity (s_1, t_1) must be a subpath of commodity (s_2, t_2). Otherwise there is no metric such that the two different (sub-)paths between s_1 and t_1 are both unique shortest paths. In our algorithm, we use the following three types of inequalities that are implied by the Bellman property:

$$x_a^{s,v} - x_a^{s,t} + \sum_{e \in \delta^-(v)} x_e^{s,t} \leq 1 \quad (s,t),(s,v) \in K,\ a \in A\ , \tag{6}$$

$$x_a^{v,t} - x_a^{s,t} + \sum_{e \in \delta^-(v)} x_e^{s,t} \leq 1 \quad (s,t),(v,t) \in K,\ a \in A\ , \tag{7}$$

$$x_a^{s,v} + x_a^{v,t} - x_a^{s,t} - 2(1 - \sum_{e \in \delta^-(v)} x_e^{s,t}) \leq 0 \quad (s,v),(v,t),(s,t) \in K,\ a \in A\ . \tag{8}$$

Although in general none of these inequalities is facet-defining for the polytope associated with (2), they all proved to be very useful in practice. In our implementation, we separate over each of these three classes with a straightforward enumerative algorithm.

In addition to these inequalities, which describe the valid routing path patterns independent of the given traffic demands and link capacities, our algorithm also uses cutting planes that are based on the resulting traffic flows and the link capacities. In practice, induced cover inequalities based on the precedence constrained knapsacks defined by a single arc capacity constraint (2c) and the subpath consistency among the paths across that arc proved to be very effective. Due to the space limitations in this extended abstract, we cannot discuss these inequalities here. A detailed description of these inequalities and the heuristic separation methods used in our algorithm is given in [9].

Whenever an integer solution candidate for the (incomplete) master formulation is found, we must solve the client problem to decide whether or not it defines a valid unsplittable shortest path routing and to find a compatible metric or a violated conflict inequality (2f). In our implementation, we solve the client problem not only for the fully integer solutions at the leaves of the master problem's branch-and-bound tree, but also for non-integer solutions arising within the branch-and-bound tree. In practice, this modification drastically reduced the running time of the overall algorithm.

At each node of the master problem's branch and bound tree, we consider the potential forwarding $F \subseteq V \times A$ defined by the integer and near integer routing variables. In our implementation, we let $F := \{(t,a) : y_a^t \geq 0.8\}$. We solve the client problem whenever this presumed forwarding differs from the one at the parent node in the branch-and-bound tree by more than two node-arc pairs, if

Table 1. Computational results for SNDlib problems.

Problem	Nodes	Links	Demands	LP	LB	Sol	Nodes	Gap (%)	Time (s)
Atlanta	15	22	210	0.65	0.86	0.86	30	0.0	10.3
Dfn-bwin	10	45	90	0.34	0.69	0.69	89	0.0	26.5
Dfn-gwin	11	21	110	0.50	0.51	0.51	521	0.0	16.3
Di-yuan	11	42	22	0.25	0.62	0.62	33	0.0	1.8
France	25	45	300	0.60	0.71	0.74	76	5.0	10000.0
Germany50	50	88	662	0.64	0.64	0.73	56	12.7	10000.0
NewYork	16	49	240	0.44	0.62	0.62	15	0.0	54.9
Nobel-EU	28	41	378	0.44	0.44	0.45	75	0.3	10000.0
Nobel-GER	17	26	121	0.64	0.73	0.73	101	0.0	114.1
Nobel-US	14	21	91	0.48	0.49	0.49	77	0.0	20.4
Norway	27	51	702	0.54	0.54	0.62	99	14.9	10000.0
PDH	11	34	24	0.34	0.80	0.80	85	0.0	6.37
Polska	12	18	66	0.82	0.93	0.93	2149	0.0	200.2
TA1	24	55	396	0.30	0.93	0.93	11	0.0	289.2

the depth of the current node in the branch-and-bound is 2^k for some $k \in \mathbb{Z}$, or if all arc-routing variables are integer. If the linear relaxation of the client problem (3) is feasible for this forwarding F, the computed link lengths define a heuristic solution for the Min-Con-USPR problem, which may improve on the best known solution. Otherwise, if a violated conflict inequality (2f) is found, this inequality may cut off the entire invalid branch at the current node in the branch-and-bound tree.

4 Results

The presented algorithm has been implemented as part of the network optimization library Discnet [2]. The data structures and algorithms are based on the standard c++ library and Leda [1], the linear programs arising in the solution process are solved with Cplex 11.0 [21]. The master problem's branch-and-cut framework and all separation procedures are implemented directly in c++.

Table 1 shows computational results for a collection of benchmark problems taken from the Survivable Network Design Library [23]. All computations were performed on an Intel Pentium 4 machine with 2.66 GHz and 4 GB RAM running Linux 2.6. The algorithm was run with a total CPU time limit of 10,000 seconds on each problem instance. The underlying networks are bidirectional and have the same capacity for both directions of all links. The numbers of nodes, bidirected links and non-zero traffic demands are shown in the first columns of Table 1. Column LP shows the lower bound obtained by solving the initial linear relaxation of (2) at the root node of the master problem's branch-and-bound tree. The columns LB and Sol show the best proven lower bound and the best solution value found by our algorithm within the given time limit. The remaining columns show the number of explored branch-and-bound nodes, the residual optimality gap, and the total CPU time until either optimality was proven or the time limit was exceeded.

The results show that our algorithm can be used to solve real-world size problems. All small and medium size instances have been solved optimally within seconds or minutes. For large problems optimality cannot always be achieved. Instances with dense networks and lots of potential routing paths for most demand pairs are more difficult than those where the underlying networks are fairly sparse. For instances with dense networks, lots of violated conflict constraints are separated during the execution of the algorithms, which often drastically slows down the solution of the linear relaxation. For the most difficult problems, only few branch-and-bound nodes could be explored. Yet, even for those problem that could not be solved to optimality, our algorithm always found better solutions than length-based heuristic and Lagrangian approaches. Our algorithm also clearly outperforms all other integer programming approaches presented in the literature so far, which typically even fail to achieve gaps below 30% for networks larger than 10 nodes.

References

1. Algorithmic Solutions Software GmbH: LEDA – Library of Efficient Data types and Algorithms (2000–2007). `http://www.algorithmic-solution.com/leda`
2. atesio GmbH: DISCNET – Network optimization software library (2000–2007). `http://www.atesio.de`
3. Ben-Ameur, W., Gourdin, E.: Internet routing and related topology issues. SIAM Journal on Discrete Mathematics **17**, 18–49 (2003)
4. Bley, A.: A Lagrangian approach for integrated network design and routing in IP networks. In: W. Ben-Ameur, A. Pertrowski (eds.) 1st International Network Optimization Conference, pp. 107–113. Institut National des Télécommunications, Evry/Paris (2003)
5. Bley, A.: Finding small administrative lengths for shortest path routing. In: L. Gouveia, C. Mourão (eds.) 2nd International Network Optimization Conference, pp. 121–128. Universidade de Lisboa, Lisbon (2005)
6. Bley, A.: On the approximability of the minimum congestion unsplittable shortest path routing problem. In: M. Jünger, V. Kaibel (eds.) 11th Conference on Integer Programming and Combinatorial Optimization, *LNCS*, vol. 3509, pp. 97–110. Springer, Heidelberg (2005)
7. Bley, A.: Approximability of unsplittable shortest path routing problems. Technical report ZR-06-02, Zuse Institute Berlin (2006)
8. Bley, A.: Inapproximability results for the inverse shortest paths problem with integer lengths and unique shortest paths. Networks **50**, 29–36 (2007)
9. Bley, A.: Routing and capacity optimization for IP networks. PhD thesis, Technische Universität Berlin (2007)
10. Bley, A., Grötschel, M., Wessäly, R.: Design of broadband virtual private networks: Model and heuristics for the B-WiN. In: N. Dean, D. Hsu, R. Ravi (eds.) Robust Communication Networks: Interconnection and Survivability, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 53, pp. 1–16. AMS (1998)

11. Bley, A., Koch, T.: Integer programming approaches to access and backbone IP-network planning. In: 3rd International Conference on High Performance Scientific Computing. Hanoi, Vietnam (2006)
12. Bourquia, N., Ben-Ameur, W., Gourdin, E., Tolla, P.: Optimal shortest path routing for Internet networks. In: W. Ben-Ameur, A. Pertrowski (eds.) 1st International Network Optimization Conference, pp. 119–125. Institut National des Télécommunications, Evry/Paris (2003)
13. Broström, P., Holmberg, K.: Determining the non-existence of compatibel OSPF weights. In: Nordic MPS 2004, no. 14 in Linköping Electronic Conference Proceedings, pp. 7–21. Linköping University Electronic Press (2004)
14. Broström, P., Holmberg, K.: Stronger necessary conditions for the existence of a compatible OSPF metric. Technical report LiTH-MAT-R-2004-08, Linköping University (2004)
15. Buriol, L., Resende, M., Ribeiro, C., Thorup, M.: A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. Networks **46**, 36–56 (2005)
16. Callon, R.: Use of OSI IS-IS for routing in TCP/IP and dual environments. IETF Internet RFC 1195 (1990)
17. Ericsson, M., Resende, M., Pardalos, P.: A genetic algorithm for the weight setting problem in OSPF routing. Journal of Combinatorial Optimization **6**, 299–333 (2002)
18. Fortz, B., Thorup, M.: Increasing Internet capacity using local search. Computational Optimization and Applications **29**, 13–48 (2004)
19. de Giovanni, L., Fortz, B., Labbé, M.: A lower bound for the Internet protocol network design problem. In: L. Gouveia, C. Mourão (eds.) 2nd International Network Optimization Conference, pp. 402–408. Universidade de Lisboa, Lisbon (2005)
20. Holmberg, K., Yuan, D.: Optimization of Internet protocol network design and routing. Networks **43**, 39–53 (2004)
21. ILOG CPLEX Division: CPLEX 11.0 (2007). URL `http://www.ilog.com`
22. Moy, J.: OSPF version 2. IETF Internet RFC 2328 (1998)
23. Orlowski, S., Pióro, M., Tomaszewski, A., Wessäly, R.: SNDlib 1.0 – Survivable Network Design Library. In: B. Fortz (ed.) 3rd International Network Optimization Conference. Université Libre de Bruxels, Brussels (2007). `http://sndlib.zib.de`
24. Parmar, A., Ahmed, S., Sokol, J.: An integer programming approach to the OSPF weight setting problem. Optimization Online (2006)
25. Pióro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufmann, San Francisco (2004)
26. Pióro, M., Szentesi, A., Harmatos, J., Jüttner, A.: On OSPF related network optimization problems. In: 8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks, pp. 70/1–70/14. Ilkley, UK (2000)
27. Prytz, M.: On optimization in design of telecommunications networks with multicast and unicast traffic. Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden (2002)
28. Tomaszewski, A., Pióro, M., Dzida, M., Mycek, M., Zagożdżon, M.: Valid inequalities for a shortest-path routing optimization problem. In: B. Fortz (ed.) 3rd International Network Optimization Conference. Université Libre de Bruxels, Brussels (2007)
29. Tomaszewski, A., Pióro, M., Dzida, M., Zagożdżon, M.: Optimization of administrative weights in IP networks using the branch-and-cut approach. In: L. Gouveia, C. Mourão (eds.) 2nd International Network Optimization Conference, pp. 393–400. Universidade de Lisboa, Lisbon (2005)