

# OSPF Routing with Optimal Oblivious Performance Ratio Under Polyhedral Demand Uncertainty\*

Aysegül Altın, Pietro Belotti, and Mustafa Ç. Pınar

Department of Industrial Engineering, Bilkent University, Ankara, Turkey

{aysegula,mustafap}@bilkent.edu.tr

Tepper School of Business, Carnegie Mellon University, Pittsburgh PA

belotti@cmu.edu

## Abstract

We study the *best OSPF style* routing problem in telecommunication networks, where weight management is employed to get a routing configuration with the minimum oblivious ratio. We consider *polyhedral* demand uncertainty: the set of traffic matrices is a polyhedron defined by a set of linear constraints, and the performance of each routing is assessed on its worst case congestion ratio for any feasible traffic matrix in the polyhedron. The problem is an accurate reflection of real world IP networks, not only because it considers the likelihood of having inaccurate demand estimates but also because it models one of the main currently viable traffic forwarding technologies, i.e., Open Shortest Path First (OSPF) routing with equal load sharing. This is an NP-hard problem even for a fixed demand matrix, so the problem considered in the paper is also NP-hard.

First, we prove that the optimal oblivious routing under polyhedral traffic uncertainty on a non-OSPF network can be obtained in polynomial time using a duality-based reformulation. Then we consider the OSPF routing with equal load sharing under polyhedral traffic uncertainty, and present a compact mixed-integer linear programming formulation with flow variables. We propose an alternative tree formulation and a branch-and-price algorithm. Finally, we report and discuss test results for several network instances.

**Keywords:** OSPF, oblivious routing, traffic engineering, ECMP, branch-and-price, traffic uncertainty, duality-based reformulation.

## 1 Introduction

Effective traffic engineering is important in today’s highly information dependent economy. An operator manages network resources by applying protocols that establish the routing paths, according to several criteria. Hence, effective routing strategies are crucial for improved customer satisfaction and an efficient use of network resources. We consider networks with the Open Shortest Path First (OSPF) protocol, where all routing paths are shortest paths according to a predefined metric, while ensuring a “fair” allocation of the network resources, in the particular case when the traffic demand is not known a priori. The “fairness” of a routing can be measured by the maximum utilization (i.e., the fraction of capacity used by data flow) among all links in the network. If data flow is distributed among the links in proportion to their capacities such that none of them becomes the bottleneck link, then this measure would be small and the routing is relatively fair.

OSPF is a *link-state* routing protocol developed for Internet Protocol (IP) networks in which routers send information to each other about the state of their adjacent links. A connection

---

\*Research supported through grant MISAG-CNR-1 jointly from TUBITAK, The Scientific and Technological Research Institution of Turkey, and CNR, Consiglio Nazionale delle Ricerche, Italy.

between two nodes  $s$  and  $t$  of the network is routed along the shortest paths from  $s$  to  $t$  composed of *available* links of the underlying network. These shortest paths are determined based on a metric established prior to network operations. If a tie occurs, i.e. if a node  $i$  of the path from  $s$  to  $t$  has more than one shortest path from  $i$  to  $t$ , several network operators use the Equal Cost Multi-Path (ECMP) routing: the traffic is equally divided among all the shortest paths. There are different approaches for determining these metrics. The traditional approach is to fix link weights in advance based on some criteria like physical distances or the inverse of link capacities (De Giovanni et al. [10]). The management of link weights that optimizes a design and routing criterion is the focus of the most recent references (Fortz and Thorup [12], Holmberg and Yuan [13], Parmar et al. [16], Pióro et al. [17], Tomaszewski et al. [21], and Wang et al. [23]). Recently, the telecommunications industry has tried to overcome the disadvantage of optimizing link weights by developing routing standards which do not rely on shortest path protocols. One example of such non-OSPF protocols is Multi-Path Label Switching (MPLS). We show in Section 3 that the problem of finding an oblivious routing for MPLS networks is polynomially solvable.

Weight management under ECMP is NP-hard (Fortz and Thorup [12], Pióro et al. [17], Wang et al. [23]) and the current technology does not support arbitrary load sharing. In order to tackle this difficulty, either the single path routing assumption or a couple of alternative strategies like the management of next hop selection or edge-based traffic engineering have been used. We cite Bley and Koch [8], Lin and Wang [14], and Tomaszewski et al. [21] as examples for unsplit routing while we refer to Parmar et al. [16], Sridharan et al. [20], and Wang et al. [22] for the latter case. Broström and Holmberg [9], De Giovanni et al. [10], Parmar et al. [16], Pióro et al. [17], and Tomaszewski et al. [21] also show mixed-integer modeling examples for incorporating the ECMP rule. Bley and Koch [8], Broström and Holmberg [9], and Pióro et al. [17] propose two-stage algorithms, which initially find an optimal routing scheme and then look for a metric according to which all optimal paths are shortest paths. The main drawback of these approaches is that not all configurations are guaranteed to be realized as shortest paths. Although Wang et al. [23] show that a specific class of routes can be converted to shortest paths, still no complete description of admissible routing schemes is available. Alternatively, De Giovanni et al. [10], Fortz and Thorup [12], Lin and Wang [14], Parmar et al. [16], and Wang et al. [23] consider the optimization of a design criterion and the link metric, simultaneously.

We focus on a network problem with *traffic uncertainty*: as it is very difficult to measure or predict the traffic demand over a network, considering some level of uncertainty in the definition of demand matrices would strengthen the traffic engineering efforts. In this setting, the traffic demand is not known a priori, but rather a *set*  $\mathcal{S}$  of possible traffic demands is given. The idea of “fairness” must adapt to the uncertainty in traffic demands to provide a routing that is *oblivious*, i.e. it is provably “fair” irrespective of a specific demand. Network optimization under traffic uncertainty has been a hot topic in the past years. Duffield et al. [11] consider an uncertainty model where an upper bound for the incoming and outgoing demand of each node is defined, and all demands within those bounds are allowed. Bertsimas and Sim [6] introduce an uncertainty model where all demands are allowed to take on a lower or an upper value, while imposing a maximum number of demands that are at the upper value simultaneously.

Ben-Ameur and Kerivin [5] propose a very general uncertainty model, whereby the set  $\mathcal{S}$  of traffic demands is a polyhedron defined by a set of linear inequalities. This *polyhedral uncertainty model* encloses as special cases the two models described above. They formulate a semi-infinite linear programming (LP) model where all demands are implicitly associated with a constraint. Rather than solving the semi-infinite LP explicitly, they describe an iterative method that considers an initial set of demands. At each step, after solving the current LP, a *separation* problem is solved to generate a traffic demand that cannot be routed; this demand is added to the set of constraints, and the algorithm continues until the separation gives no new demand, proving that all demands in the polytope are satisfied.

Some network routing problems have been studied from the maximum congestion perspec-

tive. Applegate and Cohen (see [2] and references therein) study an oblivious routing problem with very limited information on traffic demands: they assume that all demands that admit a feasible routing – within the network link capacities – are possible. More recently, Belotti and Pinar [4] study a routing problem with box and ellipsoidal uncertainty representations: in the former case, all traffic demands are assumed to have lower and upper bounds, while in the second case the mean-covariance information on demands is available. The set  $\mathcal{S}$  is defined as the intersection of a box, or an ellipsoid, with the set of all demands that admit a routing within the network capacity. They consider a non-OSPF routing, where paths are not constrained to be shortest w.r.t. any metric, and compare it with an OSPF routing where the weight of each link is heuristically fixed at a value inversely proportional to its capacity. One of their open questions addresses oblivious routing for OSPF networks: how to modify the link weights (the only degree of freedom for such networks) to optimize the routing performance, and hence to provide a better comparison with non-OSPF routing protocols?

We extend the work by Belotti and Pinar [4] in two directions. First, we address the first part of this question by considering an OSPF network where the OSPF weights, which uniquely define the routing paths, are decision variables. Second, we consider the general polyhedral uncertainty, instead of two special cases (box and ellipsoid) of traffic uncertainty, and we seek the most “fair” routing for this general demand set. In the non-OSPF case, we prove that the problem can be modeled with linear programming and hence is polynomially solvable. For the OSPF case, we present a compact mixed-integer linear programming formulation and an alternative tree formulation. We use the latter in a branch-and-price algorithm, which we strengthen with cutting planes. Finally, we provide an answer to the second part of the question above, by comparing the oblivious performance of OSPF and MPLS on a set of real-world network topologies.

The rest of the paper is organized as follows. In Section 2 we give some basic notation and explain the performance measure we use in our models to assess the goodness of different routings. In Section 3, we introduce the optimization models for oblivious routing with general demand uncertainty. We incorporate OSPF routing into our models in Section 4. Section 5 discusses our branch-and-price algorithm, and in Section 6 we describe the computational comparison between the oblivious OSPF routing and the oblivious MPLS routing. We give some final remarks in Section 7.

## 2 Basic definitions and measures of performance

Consider the undirected graph  $G = (V, E)$ . Edges  $\{h, k\} \in E$  are also referred to as *links*. For each link we have the associated directed pairs  $(h, k)$  and  $(k, h)$ , which we call the *arcs* of  $G$ . We denote this set of directed node pairs by  $A$ . Each link  $\{h, k\}$  is assigned  $c_{hk}$  units of capacity, which is available for the total flow on  $\{h, k\}$  in both directions. The estimated traffic flow from the source node  $s \in V$  to the sink node  $t \in V$  is  $d_{st}$ . We define the set of such directed source-sink pairs as  $Q = \{(s, t) : s, t \in V, s \neq t\}$ . The traffic matrix (TM)  $d = (d_{st})_{(s,t) \in Q}$  shows the amounts of traffic flow between all directed pairs in  $Q$ . Although  $d$  is a vector, the term *traffic matrix* is ubiquitous in the Telecommunications literature, and we shall use the term *matrix* throughout to refer to vector  $d$ .

We denote the fraction of  $d_{st}$  routed on the arc  $(h, k)$  by  $f_{hk}^{st}$ . Then  $f = (f_{hk}^{st})_{(h,k) \in A, (s,t) \in Q}$  defines a *routing* if it satisfies the following conditions:

$$\sum_{k:\{h,k\} \in E} (f_{hk}^{st} - f_{kh}^{st}) = \begin{cases} 1 & \text{if } h = s \\ -1 & \text{if } h = t \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in V, (s, t) \in Q \quad (1)$$

$$0 \leq f_{hk}^{st} \leq 1 \quad \forall (h, k) \in A, (s, t) \in Q. \quad (2)$$

Let us denote the set of all routings on  $G$  as  $\Lambda$ . Given a certain demand  $d$ , we say that  $f \in \Lambda$  is

feasible w.r.t.  $d$  if the capacity of none of the links is overloaded, that is, if  $\sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st}) \leq c_{hk} \forall \{h, k\} \in E$ . The *traffic load* assigned by  $f \in \Lambda$  to the undirected link  $\{h, k\} \in E$  for the TM  $d$  is denoted  $L_d^f(hk) = \sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st})$ , its *utilization* is  $U_d^f(hk) = L_d^f(hk)/c_{hk}$ . The fairness of a routing, i.e., the measure of how balanced the allocation of a TM  $d$  is, can be assessed on the maximum link utilization of  $f$ , and we denote it  $MaxU_d^f$ :

$$MaxU_d^f = \max_{\{h,k\} \in E} U_d^f(hk).$$

Then, the problem of finding a routing with the minimum  $MaxU_d^f$ , for a fixed TM  $d$ , is defined as  $BEST_d = \min_{f \in \Lambda} \{MaxU_d^f\}$  and modeled as

$$\min r \tag{3}$$

$$\text{s.t. } r \geq \sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st})/c_{hk} \quad \forall \{h, k\} \in E \tag{4}$$

$$f \in \Lambda. \tag{5}$$

Here,  $r$  is the maximum ratio between total flow on a link and the link capacity. We omit constraint  $\sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st}) \leq c_{hk} \forall \{h, k\} \in E$ , because it is equivalent to  $r \leq 1$ , and  $r$  is minimized. If an optimal solution has  $r > 1$ , then no routing is feasible for  $d$ .

### 3 Oblivious routing under polyhedral demand uncertainty

Let us consider now a different scenario: the traffic matrix is not known a priori, and a set  $D$  of possible matrices is given. The optimal oblivious routing problem may consist in finding the best routing configuration for all demands in  $D$ . However, we disregard those demands that do not admit any routing within the network capacity  $(c_{hk})_{\{h,k\} \in E}$  and instead define a set  $\mathcal{F}(D)$  of *routeable* demands, i.e. those demands for which there exists at least one feasible routing. Our scope is hence to determine the best routing that supports all  $d \in \mathcal{F}(D)$  in the most balanced way. For uncertain demands, it yields a conservative strategy with a worst case approach where the “goodness” of a routing  $f$  is measured by the maximum link utilization ratio it achieves over  $\mathcal{F}(D)$ , i.e.,  $\max_{d \in \mathcal{F}(D)} MaxU_d^f$ . However, a more common measure is the closeness of each  $f$  to optimality for any TM  $d \in \mathcal{F}(D)$  (Applegate and Cohen [2], Belotti and Pinar [4]). It is called the *oblivious ratio* of  $f$  on the set  $\mathcal{F}(D)$  and defined as

$$OR_D^f = \max_{d \in \mathcal{F}(D)} \frac{MaxU_d^f}{BEST_d}.$$

The problem of finding the routing with the best oblivious ratio for all demands in  $\mathcal{F}(D)$  is

$$\min_{f \in \Lambda} \max_{d \in \mathcal{F}(D)} \frac{\max_{\{h,k\} \in E} U_d^f(hk)}{BEST_d}. \tag{6}$$

As  $BEST_d$  does not depend on  $\{h, k\}$ ,  $\frac{\max_{\{h,k\} \in E} U_d^f(hk)}{BEST_d}$  can be written as  $\max_{\{h,k\} \in E} \frac{U_d^f(hk)}{BEST_d}$ . Then, we can swap the two max functions in (6) and get

$$\min_{f \in \Lambda} \max_{\{h,k\} \in E} \max_{d \in \mathcal{F}(D)} \frac{U_d^f(hk)}{BEST_d}. \tag{7}$$

In the sequel, we can model (7) as the following mathematical model:

$$\min r \tag{8}$$

$$\text{s.t. } r \geq \max_{d \in \mathcal{F}(D)} \frac{\sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st})/c_{hk}}{BEST_d} \quad \forall \{h, k\} \in E \tag{9}$$

$$f \in \Lambda \tag{10}$$

where (9) implies that for each link  $\{h, k\} \in E$  and routing  $f \in \Lambda$ , we have a maximization problem over  $\mathcal{F}(D)$ . Hence the definition of  $D$  is important in modeling and solving (8)-(10).

Unlike the case with fixed traffic demands, although here  $d$  is not known it should not be considered as a variable of the optimization model (8)-(10). It is instead a variable of the inner optimization problem on the right-hand side of constraint (9). Due to the max operator in constraint (9), the model (8)-(10) is equivalent to a semi-infinite optimization model with one constraint (9) for each  $d \in \mathcal{F}(D)$ .

Another remark is useful here. In recent works on network design with uncertainty in the traffic demand, there has been an interest towards the set  $D' \subseteq D$  of so-called *dominant* demands (see Oriolo [15]), which are defined as those that suffice to describe the entire uncertainty set. Then, routing all demands in  $D'$  implies that all demands in  $D$  are also routable. For instance, in network design problems where capacity has to be installed to accommodate a set of uncertain traffic demands, it is easy to prove that a demand  $d'$  dominates all  $d$  such that  $d \leq d'$ . Oriolo [15] gives a necessary and sufficient condition for dominance between traffic demands. The idea of dominance is difficult to apply here because the objective function of the inner optimization problem is nonlinear w.r.t.  $d$ . Hence, it remains an open question to prove whether, for two demands  $d$  and  $d'$  with  $d \leq d'$ , a relationship can be established between  $\frac{MaxU_d^f}{BEST_d}$  and  $\frac{MaxU_{d'}^f}{BEST_{d'}}$ .

Bearing in mind that the demand uncertainty can be modeled in various ways, we will consider the case of polyhedral uncertainty (Ben Ameur and Kerivin [5]): traffic demand matrices are not known but are supposed to belong to a polyhedron defined by a set of linear inequalities, for instance modeling the capacity of routers or bounds on the traffic flow between some node pairs. Consequently, we consider the general traffic uncertainty model

$$D = \{d = (d_{st})_{(s,t) \in Q} : Ad \leq a, d \geq 0\} \quad (11)$$

as a set of  $H$  inequalities, with  $A \in \mathbb{R}^{H \times |Q|}$  and  $a \in \mathbb{R}^H$ . We assume w.l.o.g. that  $D$  is bounded and non-empty, and prove that the above semi-infinite optimization model can be reduced to a finite LP problem by LP duality. First, we can write (9) as

$$\max_{d \in \mathcal{F}(D)} \left\{ \sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st}) - rc_{hk}BEST_d \right\} \leq 0 \quad \forall \{h, k\} \in E. \quad (12)$$

The left-hand side of (12) is the following maximization problem  $P_{hk}$  for each  $\{h, k\} \in E$ :

$$(P_{hk}) \quad \max \sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st}) - rc_{hk}\omega \quad (13)$$

$$\text{s.t.} \quad \sum_{j: \{s,j\} \in E} (g_{sj}^{st} - g_{js}^{st}) = d_{st} \quad \forall (s,t) \in Q \quad (14)$$

$$\sum_{j: \{i,j\} \in E} (g_{ij}^{st} - g_{ji}^{st}) = 0 \quad \forall i \in V \setminus \{s, t\}, (s,t) \in Q \quad (15)$$

$$\sum_{(s,t) \in Q} (g_{ij}^{st} + g_{ji}^{st}) \leq c_{ij}\omega \quad \forall \{i, j\} \in E \quad (16)$$

$$\omega \leq 1 \quad (17)$$

$$\sum_{(s,t) \in Q} a_z^{st} d_{st} \leq a_z \quad \forall z = 1, \dots, H \quad (18)$$

$$g_{ij}^{st} \geq 0 \quad \forall (i, j) \in A, (s, t) \in Q \quad (19)$$

$$d_{st} \geq 0 \quad \forall (s, t) \in Q \quad (20)$$

where  $\omega = BEST_d$  and the traffic polytope  $D$  is defined by the inequalities (18) and (20). Applegate and Cohen [2] assume that, at an optimum of the inner optimization problem (9),  $BEST_d = 1$  and hence at least one of the arcs will be used to its full capacity for the worst

case demand. However, as Belotti and Pinar [4] show, this is not a valid assumption if  $D$  is bounded. They give an example of the case where  $d_{st} \leq \alpha \frac{\min_{\{h,k\} \in E} c_{hk}}{|Q|} \forall (s,t) \in Q$  with  $\alpha < 1$ . Then none of the links would be used totally even if all demands were routed on the link with the minimum capacity, implying  $BEST_d < 1$ .

Another remark is in order. Although we omit the capacity constraint in model (3)-(5), where  $d$  is known and  $r$  is the objective function (because  $d$  does not admit a feasible routing if the optimal  $r > 1$ ), this is no longer the case here. The demand polyhedron  $D$  is broadly defined, and may contain demands that do not admit a feasible routing. Nevertheless, we are interested in measuring the performance of a specific routing on  $\mathcal{F}(D)$ , by ensuring that both  $BEST_d$  and the oblivious ratio are computed on  $\mathcal{F}(D)$ . In order to add a notion of demand feasibility to the model, we need to intersect  $D$  with the set of those demands that admit at least one feasible routing. We do so in model (13)-(20) with  $\omega$  and (17), which we interpret as an additional constraint to the polyhedron of demands to exclude the non-routable ones.

For a given  $r$  and routing  $f$ ,  $DP_{hk}$  is a linear programming problem, which admits a dual  $DP_{hk}$  for each link  $\{h,k\} \in E$ . Consider the dual variables  $\pi_{hk}^{st}$ ,  $\sigma_{i,hk}^{st}$ ,  $\eta_{ij,hk}$ ,  $\chi_{hk}$ , and  $\lambda_z^{hk}$  of the constraints (14) - (18). If we let

$$\Pi_{i,hk}^{st} = \begin{cases} \pi_{hk}^{st} & \text{if } i = s \\ 0 & \text{if } i = t \\ \sigma_{i,hk}^{st} & \text{otherwise} \end{cases} \quad \forall i \in V, (s,t) \in Q,$$

then we have:

$$(DP_{hk}) \quad \min \quad \chi_{hk} + \sum_{z=1}^H a_z \lambda_z^{hk} \quad (21)$$

$$\text{s.t.} \quad \Pi_{i,hk}^{st} - \Pi_{j,hk}^{st} + \eta_{ij,hk} \geq 0 \quad \forall (i,j) \in A, (s,t) \in Q \quad (22)$$

$$-\pi_{hk}^{st} + \sum_{z=1}^H a_z \lambda_z^{hk} \geq f_{hk}^{st} + f_{kh}^{st} \quad \forall (s,t) \in Q \quad (23)$$

$$-\sum_{\{i,j\} \in E} c_{ij} \eta_{ij,hk} + \chi_{hk} = -rc_{hk} \quad (24)$$

$$\eta_{ij,hk} \geq 0 \quad \forall \{i,j\} \in E \quad (25)$$

$$\chi_{hk} \geq 0 \quad (26)$$

$$\lambda_z^{hk} \geq 0 \quad \forall z = 1, \dots, H. \quad (27)$$

We use  $DP_{hk}$  and the duality theorems to reduce (9) to an equivalent set of linear inequalities. Let us offer the following interpretation of the objective (21):  $\chi_{hk}$  is defined in (24) as the surplus in a hypothetical constraint  $\sum_{\{i,j\} \in E} c_{ij} \eta_{ij,hk} \geq rc_{hk}$ , and is penalized in the objective function. If we intended to include non-routable demands as well, thus considering  $D$  rather than  $\mathcal{F}(D)$ , constraint (17) would disappear and so would  $\chi_{hk}$ , leaving a model with the equality constraint  $\sum_{\{i,j\} \in E} c_{ij} \eta_{ij,hk} = rc_{hk}$  and an objective function  $\sum_{z=1}^H a_z \lambda_z^{hk}$  weighing the tight constraints of  $D$  that define the ‘‘critical’’ traffic demand  $d$  for an edge  $\{h,k\}$ , i.e., the demand that makes edge  $\{h,k\}$  most loaded. Restricting to  $\mathcal{F}(D)$  adds a penalization term  $\chi_{hk}$  that is nonzero if the critical demand is at the boundary of the (polyhedral) set of routable demands.

**Proposition 3.1.** *For the polyhedral traffic uncertainty model where  $D = \{d = (d_{st})_{(s,t) \in Q} : Ad \leq a, d \geq 0\}$ , constraint (9) for each  $\{h,k\} \in E$  can be replaced with the equivalent inequality system (22)-(27) and the inequality*

$$-\chi_{hk} - \sum_{z=1}^H a_z \lambda_z^{hk} \geq 0. \quad (28)$$

*Proof.* Suppose  $D$  is subject to polyhedral uncertainty. For each link  $\{h, k\} \in E$  consider the following LP problem:

$$\{\min 0 : (22), (23), (24), (25), (26), (27), (28)\}. \quad (29)$$

Then the proof follows directly from a simple application of the strong duality theorem with  $g_{ij}^{st}$ ,  $d_{st}$ , and  $\omega$  as the dual variables for the constraints (22)-(24), respectively.  $\square$

Let  $S_{hk} = \{(r, f, \Pi_{hk}, \eta_{hk}, \chi_{hk}, \lambda^{hk}) : (22) - (28)\}$  where  $\Pi_{hk} = (\Pi_{i,hk}^{st})_{i \in V, (s,t) \in Q}$ ,  $\eta_{hk} = (\eta_{ij,hk})_{\{i,j\} \in E}$ , and  $\lambda^{hk} = (\lambda_z^{hk})_{1 \leq z \leq H}$ .

**Corollary 3.1.** *Assuming that the traffic demand set  $D$  is subject to polyhedral uncertainty, solving the following LP yields the optimal oblivious unconstrained routing on  $G = (V, E)$ :*

$$\min r \quad (30)$$

$$s.t. \quad f \in \Lambda \quad (31)$$

$$(r, f, \Pi_{hk}, \eta_{hk}, \chi_{hk}, \lambda^{hk}) \in S_{hk} \quad \forall \{h, k\} \in E. \quad (32)$$

Consequently, the optimal oblivious ratio for MPLS routing under general traffic uncertainty can be computed in polynomial time by solving the LP problem (30)-(32), which extends the models of Applegate and Cohen [2] and Belotti and Pinar [4].

There is no condition on the structure of the routes that carry the traffic matrices on  $G = (V, E)$  in the above model. Moreover,  $f \in \Lambda$  implies that the flow from  $s$  to  $t$  can be split in any fraction among the paths defined between them. However, this latter issue is not applicable with the current traffic engineering technology. The current approach is either to use a unique path for each demand  $d_{st}$  or to split it *equally* among multiple paths. Besides, the OSPF protocol is compatible with the current applications. Hence the inclusion of OSPF routing with the aim of “fair” traffic allocation is also important for the sake of applicability.

## 4 Modeling OSPF routing

Open Shortest Path First (OSPF) routing protocols route flow between node pairs along the corresponding shortest paths defined with respect to some metric. This metric is usually fixed in advance for each link, for instance as the inverse of its capacity, and the shortest paths are computed accordingly. The more recent efforts are on managing the OSPF metric to optimize a given design criterion. We also adopt this latter approach since we believe in the necessity of good weight management to improve the OSPF performance. Therefore, we consider OSPF routing with the goal of minimizing the oblivious ratio  $r$  defined in (7).

Naturally, there can be more than one shortest path between a pair of nodes. One could consider unsplitable routing such that each demand  $d_{st}$  can be routed on a unique path. However, using multiple paths would mostly improve the fairness of work load distribution. To further clarify this issue, consider the simple example given in Figure 1. The numbers on each link are its weight and capacity, respectively. For example, the link  $\{A, B\}$  is assigned a unit weight and 12 units of capacity, which is available for the traffic in both directions.

Suppose that we have a fixed traffic matrix  $d$  where nodes  $A$  and  $C$  exchange some traffic with  $d_{AC} = 8$  and  $d_{CA} = 4$ . In this case, we can define 3 shortest paths between  $A$  and  $C$  in both directions. With unsplitable routing, we would have the situation shown in Figure 1a where the link  $\{A, C\}$  is used to its full capacity although the rest of the links are left idle. If we allow splittable routing, then we could have the case in Figure 1b where the utilization of each link is around 50%. The latter routing is more fair since all links would use almost equal fractions of their capacities. Hence rather than unsplitable routing, it would be better to apply ECMP routing in which the demand  $d_{st}$  accumulated at some node  $i$  is splitted evenly among all shortest paths between  $i$  and  $t$ .

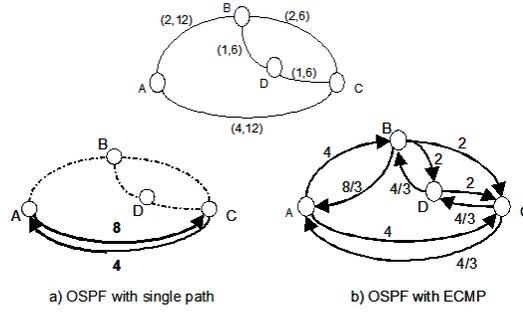


Figure 1: Example for splittable vs. unsplittable routing.

## 4.1 Variables and parameters

Below we present two models for OSPF routing. The OSPF protocol is implemented by a technology where link weights are *integer* and vary between 1 and  $\Theta_{\max}$ , which equals 65535 and is an input of the problem<sup>1</sup>. Integer variables  $\theta_{ij}$  define the weight used at each arc  $(i, j)$  and range between 1 and  $\Theta_{\max}$ . Although we only need the optimal values of  $f$  and  $\theta$ , some auxiliary classes of variables are needed. We define  $\rho_i^t$  as the shortest path distance from  $i$  to  $t$  according to the metric defined by the  $\theta_{ij}$  variables. We model ECMP constraints with variables  $\varphi_i^{st}$  giving the fraction of flow that, after entering node  $i$ , is split among different outgoing arcs due to the ECMP rule. For example, in Figure 1b, for node B and the demand from A to C we have  $\varphi_B^{AC} = 0.25$  because the portion of flow from A to B is equally split on the two shortest paths from B to C. Similarly,  $\varphi_C^{CA} = 1/3$  since there are three shortest paths from C to A.

## 4.2 Flow formulation

To model OSPF routing, we must ensure that all demands are routed on the corresponding shortest paths. We use binary variables  $y_{ij}^t$  to indicate if the arc  $(i, j)$  is on a shortest path from  $i$  to  $t$ , i.e., if it is a *Shortest Path arc* (*SP arc*) for  $t$ . Constraints

$$f_{ij}^{st} \leq y_{ij}^t \quad \forall (i, j) \in A, (s, t) \in Q \quad (33)$$

ensure that flow only occurs on shortest path arcs, whereas constraints

$$y_{ij}^t + \rho_j^t - \rho_i^t + \theta_{ij} \geq 1 \quad \forall (i, j) \in A, t \in V \quad (34)$$

$$-y_{ij}^t - \frac{\rho_j^t - \rho_i^t + \theta_{ij}}{2\Theta_{\max}} \geq -1 \quad \forall (i, j) \in A, t \in V \quad (35)$$

model OSPF routing. The Bellman conditions  $\rho_j^t - \rho_i^t + \theta_{ij} \geq 0$ , imposing non-negative reduced cost of arc  $(i, j)$  for the set of shortest paths destined at  $t$ , are dominated by constraints (34), and therefore are not included. If  $y_{ij}^t = 1$ , then  $(i, j)$  is an *SP arc* for all demands destined at  $t$  and hence the Bellman condition must be satisfied with equality, as imposed by (34) and (35). If an arc  $(i, j)$  is not an *SP arc* with destination  $t$  according to weights  $\theta$ , then its reduced cost must be at least 1 since we require  $\theta_{ij} \geq 1$ . Since for each link  $\{i, j\} \in E$  we have  $(i, j) \in A$  and  $(j, i) \in A$ , we use  $2\Theta_{\max}$  in the denominator as mentioned in Holmberg and Yuan [13]. For the Bellman condition on arc  $(j, i)$ , we have  $\rho_i^t \geq \rho_j^t - \theta_{ji}$ , hence for arc  $(i, j)$  we have  $\rho_j^t - \rho_i^t + \theta_{ij} \leq \theta_{ji} + \theta_{ij} \leq 2\Theta_{\max}$ . Finally, the ECMP rule is implemented as follows:

$$f_{ij}^{st} \leq \varphi_i^{st} \quad \forall (i, j) \in A, (s, t) \in Q \quad (36)$$

$$1 + f_{ij}^{st} - \varphi_i^{st} \geq y_{ij}^t \quad \forall (i, j) \in A, (s, t) \in Q \quad (37)$$

<sup>1</sup> Network nodes in OSPF communicate through packets where the link weight is stored in a field of two bytes, or 16 bits; hence it is integer and ranges between one and  $65535 = 2^{16} - 1$ . The value of  $\Theta_{\max}$  depends on the protocol: for instance, in two versions of the IS-IS protocol it is 63 and  $2^{24} - 1$ , respectively.

with the variable bounds

$$1 \leq \theta_{ij} \leq \Theta_{\max} \quad \text{integer} \quad \forall (i, j) \in A \quad (38)$$

$$y_{ij}^t \in \{0, 1\} \quad \forall (i, j) \in A, t \in V \quad (39)$$

$$0 \leq \varphi_i^{st} \leq 1 \quad \forall i \in V, (s, t) \in Q. \quad (40)$$

Constraints (36) and (37) impose that if demand  $d_{st}$  is routed via some node  $i$ , then all arcs originating at  $i$  and contained in some shortest path to  $t$  should share the total flow accumulated at  $i$  equally. Now let  $F_{OSPF} = \{(f, \rho, \theta, y, \varphi) : (33) - (40)\}$ .

**Corollary 4.1.** *The solution of the following linear MIP is the optimal oblivious OSPF routing on  $G = (V, E)$  with equal load sharing under polyhedral demand uncertainty:*

$$\min r \quad (41)$$

$$s.t. \quad f \in \Lambda \quad (42)$$

$$(r, f, \Pi_{hk}, \eta_{hk}, \chi_{hk}, \lambda^{hk}) \in S_{hk} \quad \forall \{h, k\} \in E \quad (43)$$

$$(f, \rho, \theta, y, \varphi) \in F_{OSPF}. \quad (44)$$

The flow formulation (41)-(44) that models OSPF with ECMP needs  $2|E|(|V| + 1) + |V|^3$  additional variables, of which  $2|E||V|$  are binary and  $2|E|$  are integer. Even for medium sized networks, our formulations can get very large and require an excessive solution time on an MILP solver. An ad-hoc formulation and solution method are therefore needed.

### 4.3 Alternative formulation

In this section we adopt an alternative approach where we use tree rather than flow variables to model OSPF routing. This model uses a set of tree variables each corresponding to an *SP tree*, which is a widely used structure in OSPF and IS-IS routing protocols.

A *Shortest Paths Tree (SP tree)* is an acyclic directed subgraph of  $G$ , rooted at a node  $t$ , such that, for at least one metric  $\theta$ , all and only directed paths from any node  $i$  to  $t$  within the *SP tree* are the shortest ones on  $G$  according to  $\theta$ . Note that  $\theta$  uniquely identifies an *SP tree* rooted at a given node  $t$ . If there are multiple shortest paths from a node  $s \in V \setminus \{t\}$  to  $t$ , then  $T$  includes all of them. An *SP tree*  $T$  is, in general, not a tree as it is the union of a set of paths.

In our formulation, an *SP tree*  $T$  shows how all the traffic towards its destination node  $t$  should be routed on the arcs of the backbone graph. Since each *SP tree*  $T$  defines a routing configuration for its root node, we want only one *SP tree* to be used for each  $t \in V$ . We model this requirement via binary  $\tau_T^t$  variables, which indicate whether the implicitly defined *SP tree*  $T$  is used to route all traffic flow ending at  $t$  or not. We define  $\Omega_t$  as the set of *SP trees* with destination  $t$  and  $\Omega_{ij}$  as the set of *SP trees* containing arc  $(i, j)$ . We ensure that a single *SP tree* is used for each destination by the constraint

$$\sum_{T \in \Omega_t} \tau_T^t = 1 \quad \forall t \in V. \quad (45)$$

Moreover, the inequality

$$f_{ij}^{st} \leq \sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t \quad \forall (i, j) \in A, (s, t) \in Q \quad (46)$$

relates the  $\tau$  variables to flow variables. Constraint (46) is analogous to (33) of the flow formulation – note that we replace each  $y_{ij}^t$  variable with  $\sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t$ . If some flow from  $s$  to  $t$  is routed on arc  $(i, j)$ , then it should be an *SP arc* for  $t$  in any *SP tree*  $T$  that will be used for it. Hence the sum on the right-hand side of (46) must be 1, which ensures that the *SP tree* for  $t$

contains  $(i, j)$ . The OSPF constraints change as follows:

$$\sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t + \rho_j^t - \rho_i^t + \theta_{ij} \geq 1 \quad \forall t \in V, (i, j) \in A \quad (47)$$

$$- \sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t - \frac{\rho_j^t - \rho_i^t + \theta_{ij}}{2\Theta_{\max}} \geq -1 \quad \forall t \in V, (i, j) \in A, \quad (48)$$

and are analogous to (34) and (35), respectively. The summations in (47) and (48) are equal to one only for the *SP arcs* ensuring that their reduced costs are zero. The final set of constraints are the following ECMP constraints

$$f_{ij}^{st} \leq \varphi_i^{st} \quad \forall (i, j) \in A, (s, t) \in Q \quad (49)$$

$$1 + f_{ij}^{st} - \varphi_i^{st} \geq \sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t \quad \forall (i, j) \in A, (s, t) \in Q \quad (50)$$

with the variable bounds

$$1 \leq \theta_{ij} \leq \Theta_{\max} \quad \text{integer} \quad \forall (i, j) \in A \quad (51)$$

$$\tau_T^t \in \{0, 1\} \quad \forall t \in V, T \in \Omega_t \quad (52)$$

$$0 \leq \varphi_i^{st} \leq 1 \quad \forall i \in V, (s, t) \in Q. \quad (53)$$

For the clarity of notation, let  $T(\Omega) = \{(f, \rho, \theta, \tau, \varphi) : (45) - (53)\}$  where  $\Omega = \cup_{t \in V} \Omega_t = \cup_{(i,j) \in A} \Omega_{ij}$ . Flow and tree formulations are analogous, and the difference is how one tries to solve them. Since *SP trees* are defined by the weight metric  $\theta$ , which is also a variable of our model, we know neither the number nor the structure of *SP trees* explicitly in advance. Hence the sets  $\Omega_t$  and  $\Omega_{ij}$  are implicitly defined.

We combine the oblivious routing model discussed in Corollary 3.1 with the flow and the tree OSPF models to find the OSPF routing under ECMP rule with minimum oblivious ratio.

## 5 A branch-and-price algorithm for exact solution

The number of paths in  $G = (V, A)$  depends on the structure of the graph, and it can grow exponentially with  $|V|$ . So do the number of variables in the tree formulation. Hence we have developed a branch-and-price (B&P) algorithm. This method, introduced by Barnhart et al. [3], is an efficient approach to solve problems with a large number of variables. Based on the branch-and-bound (B&B) scheme, it starts with a restricted LP relaxation ( $RLP_0$ ) with fewer variables than the original problem and applies column generation to solve the problem at each node of the B&B tree. The subproblem in a B&P node ( $RLP_{curr}$ ) is optimal when no new columns are added to the problem, and branching occurs if the integrality conditions are not satisfied by the current solution. An application of the B&P algorithm for the design of Virtual Private Networks (VPN) can be found in Altın et al. [1].

We summarize the main steps of our B&P in Algorithm 1. The details of the application are addressed in the rest of this section, where we use the terms *SP tree  $T$  destined at  $t$*  and  $\tau_T^t$  *variable* interchangeably.

### 5.1 Initialization

Our B&P algorithm begins with a restricted formulation  $RLP_0$ , defined with a subset of all possible *SP trees*. An optimal solution for  $RLP_0$  is feasible but in general not optimal for the

---

**Algorithm 1** B&P algorithm

---

**Require:** undirected graph  $G = (V, E)$ , traffic polytope  $D$ , link capacity vector  $c$ ;

**Ensure:** optimal oblivious ratio for OSPF routing and  $(G, D, c)$ ;

*Initialize:*

Find an initial set  $\Omega_0$  of *SP trees*;

$\tilde{\Omega} \leftarrow \Omega_0$ ; //  $\tilde{\Omega}$ : current set of *SP trees*;

$S \leftarrow \{\text{root}\}$ ; //  $S$ : the current set of unevaluated B&P nodes  
// *root*: root node of the B&P tree;

$UB \leftarrow \infty$ ;

**while**  $S \neq \emptyset$  **do**

$n_b \in \operatorname{argmin}_{n \in S} LB(n)$ ;

$S \leftarrow S \setminus \{n_b\}$ ;

**repeat**

*Optimize:* Get  $z^*(n_b, \tilde{\Omega})$ ; // optimal value of  $RLP_{curr}$

*Price:*

**for all**  $t \in V$  **do**

Search for a new *SP tree*  $\hat{T}$  destined at  $t$ ;

**if**  $\tau_{\hat{T}}^t$  has a promising reduced cost **then**

$\tilde{\Omega} = \tilde{\Omega} \cup \hat{T}$ ; // update the current set of *SP trees*

Update  $RLP_{curr}$ ;

**until** no new  $\hat{T}$  can be found

**if** current LP is feasible **then**

Let  $z_{ub}^*(n_b)$  be the upper bound obtained by approximation;

**if**  $z_{ub}^*(n_b) < UB$  **then**

$UB \leftarrow z_{ub}^*(n_b)$ ;

**if** the current optimal solution is not integral **then**

*Branch:*

Select a fractional  $\bar{\tau}_{\hat{T}}^t$  variable and branch;

Create two child nodes  $\{n_r, n_l\}$  and let  $S = S \cup \{n_r, n_l\}$ ;

*Extract* B&P nodes that are fathomed by bound or infeasibility from  $S$ .

---

original problem:

$$(RLP_0) \quad \min r \quad (54)$$

$$\text{s.t.} \quad f \in \Lambda \quad (55)$$

$$(r, f, \Pi_{hk}, \eta_{hk}, \chi_{hk}, \lambda^{hk}) \in S_{hk} \quad \forall \{h, k\} \in E \quad (56)$$

$$(f, \rho, \theta, \tau, \varphi) \in T(\Omega_0) \quad (57)$$

where  $\Omega_0 = \cup_{t \in V} \Omega_t^0 = \cup_{(i,j) \in A} \Omega_{ij}^0$  is the initial set of *SP trees*.  $RLP_0$  is feasible if there exists a metric for which at least one variable  $\tau$  in  $RLP_0$  corresponds to a *SP tree*, for every node  $t \in V$  in  $\Omega_0$ . This ensures that each  $t$  is reachable from every other node of  $V$ . Therefore, we need to construct  $\Omega_0$  using an initial metric, where each arc could be assigned a unit weight or a value proportional to the physical distance between its two endpoints. We use an *inverse capacity* weight setting in which the weight of each arc is equal to the inverse of its capacity (this has been used for instance in some network operated by Cisco). Finally,  $|\Omega_0| = |V|$  and we have one  $\tau_T^t$  variable for each  $t$  in  $RLP_0$ . Hence we start with  $|V|$  binary variables, which is much less than  $2|E||V|$  of the flow formulation.

## 5.2 Pricing

In each B&P node, we have a restricted formulation  $RLP_{curr}$  derived from  $RLP_0$  by adding branching rules and generating new  $\tau$  variables. The B&P node is solved by applying column generation to  $RLP_{curr}$ : at each iteration, after solving  $RLP_{curr}$ , a *pricing* procedure is used to find a set of new  $\tau$  variables whose reduced costs are negative and thus correspond to promising routing strategies for improving the current configuration.

Let  $\zeta_t$ ,  $\nu_{ij}^{st}$ ,  $v_{ij}^t$ ,  $\varsigma_{ij}^t$ , and  $\kappa_{ij}^{st}$  be the dual variables of the constraints (45), (46), (47), (48), and (50), respectively. In a B&P node ( $n_b$ ) other than the root node, we also have to take into account the set  $B$  of branching rules applied at all B&P nodes on the path from ( $n_b$ ) to the root node. We do not describe such branching rules here: a detailed description is in Subsection 5.4. Suffice it for now to denote as  $B_t^{ij}$  the subset of branching rules involving  $\tau$  variables corresponding to *SP trees* containing  $(i, j)$  and destined at  $t$ . Let us also denote with  $\mu_k$  the dual variable of a branching rule  $k \in B = \bigcup_{t \in V, \{i,j\} \in E} B_t^{ij}$ . The reduced cost of each  $\tau_T^t$  variable is

$$red_T^t = -\zeta_t - \sum_{(i,j) \in T} \left[ v_{ij}^t - \varsigma_{ij}^t + \sum_{s \in V \setminus \{t\}} (\nu_{ij}^{st} - \kappa_{ij}^{st}) + \sum_{k \in B_t^{ij}} \mu_k \right]. \quad (58)$$

The B&P algorithm has an initial set  $\Omega_0$  of *SP trees*. As we generate new  $\tau$  variables, we include the corresponding *SP trees* in our model. While  $red_T^t$  is nonnegative for all *SP trees* within the current formulation, if a new  $\tau_T^t$  with a negative reduced cost is found, the current solution may improve if all demands destined at  $t$  are routed on  $\hat{T}$ .

To determine such *SP trees* we solve a shortest path problem for each destination node  $t \in V$  with arc metric  $\alpha$  on an auxiliary graph  $G_{aux}(t, \alpha)$ , where

$$\alpha_{ij} = -\bar{v}_{ij}^t + \bar{\varsigma}_{ij}^t - \sum_{s \in V \setminus \{t\}} (\bar{\nu}_{ij}^{st} - \bar{\kappa}_{ij}^{st}) - \sum_{k \in B_t^{ij}} \mu_k \quad \forall (i, j) \in A.$$

Two important issues should be handled with care at this stage. First, the solution of the pricing problem must comply with the definition of an *SP tree*, i.e., ECMP routing and integer arc weights must be ensured. Second, there is no guarantee that  $\alpha$  is nonnegative and that  $G_{aux}(t, \alpha)$  has no negative cycles. Therefore we cannot use the well known shortest path algorithms of Dijkstra or Bellman-Ford to solve the pricing problem. As a result, for each destination node  $t$ , we solve the pricing problem, which is a single-node OSPF routing problem,

to determine promising *SP trees* using the following MIP model  $PR_t$

$$(PR_t) \quad z_t^* = \min \sum_{(i,j) \in A} \alpha_{ij} y_{ij} \quad (59)$$

$$\text{s.t.} \quad f \in \Lambda \quad (60)$$

$$f_{ij}^s \leq \varphi_i^s \quad \forall (i,j) \in A, s \in V \setminus \{t\} \quad (61)$$

$$1 + f_{ij}^s - \varphi_i^s \geq y_{ij} \quad \forall (i,j) \in A, s \in V \setminus \{t\} \quad (62)$$

$$-y_{ij} - \left( \frac{\rho_j - \rho_i + \theta_{ij}}{2\Theta_{\max}} \right) \geq -1 \quad \forall (i,j) \in A \quad (63)$$

$$y_{ij} + \rho_j - \rho_i + \theta_{ij} \geq 1 \quad \forall (i,j) \in A \quad (64)$$

$$0 \leq \varphi_i^s \leq 1 \quad \forall i \in V, s \in V \quad (65)$$

$$1 \leq \theta_{ij} \leq \Theta_{\max} \quad \text{integer } \forall (i,j) \in A \quad (66)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \quad (67)$$

$$\rho_i \geq 0 \quad \forall i \in V \quad (68)$$

where the binary variable  $y_{ij}$  indicates if  $(i, j)$  is an *SP arc* for  $t$  whereas  $f$ ,  $\varphi$ ,  $\rho$ , and  $\theta$  retain their definitions made in the original master problem. Since  $PR_t$  contains the OSPF and the ECMP constraints, its solution is an *SP tree*  $\hat{T}$  defined w.r.t. some metric  $\theta$  and its cost is  $z_t^* = \sum_{(i,j) \in \hat{T}} \alpha_{ij}$ . If  $z_t^* < \bar{c}_t$ , we have a new routing configuration whose inclusion could improve the current solution of the original problem. Hence we update the current set of *SP trees* by including  $\hat{T} = \{(i, j) \in A : y_{ij}^* = 1\}$  with destination  $t$ . We solve the pricing problem for all nodes  $t \in V$  at each call of the *Price* routine in Algorithm 1.

### 5.3 Generating an upper bound

At each node  $n_b$  of the B&P tree, we price  $\tau$  variables and reoptimize the updated  $RLP_{curr}$  problem until we cannot identify new *SP trees*. At this point, a lower bound  $LB(n_b)$  on the optimal oblivious ratio  $r(n_b)$  is given. A feasible solution of the original master problem would give an upper bound (*UB*) on the optimal oblivious ratio  $r^*$  that helps cut off a part of the B&P tree. We have implemented a simple rounding method, that starts from an optimal solution of  $RLP_{curr}$  and optimal values  $\bar{\tau}_T^t$  of the *SP tree* variables. For each  $t \in V$ , we pick the *SP tree*  $T^*$  with maximum  $\bar{\tau}_{T^*}^t$  among the *SP trees* destined at  $t$ . Then we round these  $\bar{\tau}_{T^*}^t$  variables to 1 and solve the original master problem. If this routing strategy is viable, i.e., if the corresponding LP is feasible, we have an upper bound  $z_{ub}(n_b)$  on the optimal oblivious ratio  $r^*$ .

### 5.4 Branching

We use a branching rule that exploits the problem structure to partition the solution space without complicating the pricing problem. As we have mentioned in Algorithm 1, we use  $\bar{\tau}_T^t$  variables to determine the restrictions we impose in each branching step. However, we do not base our branching rule on the dichotomy of these variables. Such an approach would not be efficient since the algorithm might get stuck to the same set of *SP trees* and loop. Suppose that we use a branching rule such that  $\tau_T^t = 0$  in one branch and  $\tau_T^t = 1$  in the other. The former condition means that the *SP tree*  $T$  cannot be used for the destination node  $t$ . Nevertheless, it is possible that  $PR_t$  finds an *SP tree*  $\tilde{T}$  with exactly the same set of arcs of  $T$ , i.e.,  $\tilde{T} \equiv T$ . We create two subdivisions of the current problem based on an arc  $(i^*, j^*)$  being or not being an *SP arc* for the demand  $d_{s^*t^*}$  of the pair  $(s^*, t^*)$ . The procedure for selecting the quadruple  $(i^*, j^*, s^*, t^*)$  is explained in Procedure 2.

After branch selection, we use the following rule to partition the solution space by creating two new nodes such that either of the following conditions holds:

---

**Procedure 2** Branch selection.

**Require:**  $\bar{\tau}_T^t$  values in the solution of  $RLP_{curr}$ ;

**Ensure:** The quadruple  $(h^*, k^*, s^*, t^*)$ ; //  $(s^*, t^*) \in Q$ ,  $(h^*, k^*) \in A$

Take the most fractional  $\bar{\tau}_{T^*}^t$ , let  $t^* \leftarrow t$  and  $T_1 \leftarrow T^*$ ;

Find the second most fractional  $\bar{\tau}_{T^*}^{t^*}$ , let  $T_2 \leftarrow T_*$ ;

$found \leftarrow FALSE$ ;

**for all**  $(h, k) \in A$  **do**

**if**  $(h, k) \in T^* \cup T_*$  and  $(h, k) \notin T^* \cap T_*$  **then**

**if**  $f_{hk}^{st} > 0$  and  $(h, k, s, t)$  is not used in upper branches **then**

**if**  $deg(h) > 1$  **then**

$(h^*, k^*, s^*, t^*) \leftarrow (h, k, s, t)$ ;

$found \leftarrow TRUE$ ;

**BREAK**;

**if**  $found = FALSE$  **then**

**for all**  $(h, k) \in A$  **do**

**for all**  $(s, t) \in Q$  **do**

**if**  $f_{hk}^{st} > 0$  and  $(h, k, s, t)$  is not used in upper branches **then**

$(h^*, k^*, s^*, t^*) \leftarrow (h, k, s, t)$ ;

$found \leftarrow TRUE$ ;

**BREAK**;

**if**  $found = FALSE$  **then**

**STOP**; //fathom the current B&P node

---

- $(i^*, j^*)$  is not an *SP arc* for the pair  $(s^*, t^*)$ , i.e.,

$$f_{i^*j^*}^{s^*t^*} = 0 \quad (69)$$

- $(i^*, j^*)$  is an *SP arc* for the pair  $(s^*, t^*)$ , i.e.,

$$f_{i^*j^*}^{s^*t^*} \geq \begin{cases} \frac{1}{deg(s^*)} & \text{if } i^* = s^* \\ \frac{\sum_{(k,i^*) \in A} f_{ki^*}^{st}}{deg(i^*)-1} & \text{if } i^* \in V \setminus \{s^*, t^*\} \end{cases} \quad (70)$$

where  $deg(i^*)$  is the number of arcs incident to  $i^*$ .

The right hand side of (70) in both cases is the ratio of total outflow for node  $i^*$  to the maximum number of outgoing *SP arcs* that can be incident to it in any routing  $f \in \Lambda$ . Namely, for  $s^*$  all outgoing arcs incident to it can be *SP arcs* whereas for any other node  $i^* \in V \setminus \{s^*, t^*\}$ , in order for  $(i^*, j^*)$  to be an *SP arc*, we must have at least one incoming arc and at most  $deg(i^*) - 1$  outgoing arcs. Hence in the most splitted case all arcs departing from node  $i^*$  would be *SP arcs* and the total flow accumulated in  $i^*$  will be splitted evenly among them according to the ECMP routing rule.

Given the current B&P node  $n_b$  and its associated relaxation  $RLP_{curr}$ , we create two new nodes  $n_r$  and  $n_l$  by adding the constraints (69) and (70) to the current restricted problem as well as the corresponding pricing problems  $PR_{t^*}$ . Additionally, we impose either of the constraints

- do not use *SP trees* containing arc  $(i^*, j^*)$  for  $t^*$ , i.e.,

$$\sum_{T \in \tilde{\Omega}_{i^*} \cap \tilde{\Omega}_{i^*j^*}} \tau_T^{t^*} = 0 \quad (71)$$

- do not use *SP trees* not containing arc  $(i^*, j^*)$  for  $t^*$ , i.e.,

$$\sum_{T \in \tilde{\Omega} \setminus (\tilde{\Omega}_{t^*} \cap \tilde{\Omega}_{i^*j^*})} \tau_T^{t^*} = 0 \quad (72)$$

to create  $n_r$  and  $n_l$ , respectively. The dual variables of these constraints are considered in the pricing problem discussed above.

**Proposition 5.1.** *Suppose that  $(i, j)$  is an SP arc for the pair  $(s, t)$ . Then the fraction of  $d_{st}$  routed on  $(i, j)$  satisfies the condition*

$$f_{ij}^{st} \geq \frac{1}{deg(s) * \prod_{l \in V \setminus \{s, t\}: deg(l) \geq 2} (deg(l) - 1)}. \quad (73)$$

*Proof.* Let  $SP_{st} = \{(s, j), (j, k), (k, h), \dots, (l, t)\}$  be a shortest path from  $s$  to  $t$  and  $o(i)$  be the number of arcs directed away from node  $i$  and contained in some shortest path for the pair  $(s, t)$ . Clearly, for any  $f \in \Lambda$  and each  $i \in V \setminus \{s, t\}$  with some traffic inflow, we must have  $deg(i) \geq 2$  and  $1 \leq o(i) \leq deg(i) - 1$ . Moreover,  $deg(s) \geq 1$ ,  $deg(t) \geq 1$ , and  $deg(s) \geq o(s) \geq 1$ .

Initially, for the source node  $s$ , ECMP rule imposes that  $f_{sj}^{st} = 1/o(s) \geq 1/deg(s) \forall f \in \Lambda$ . Next, consider node  $j$  for which  $f_{sj}^{st} > 0$ . Since  $f \in \Lambda$  and  $j \in V \setminus \{s, t\}$ , we know that  $\sum_{(j,i) \in A} f_{ji}^{st} = \sum_{(i,j) \in A} f_{ij}^{st}$ . Then due to the ECMP rule

$$f_{jk}^{st} = \frac{f_{sj}^{st} + \sum_{(i,j) \in A: i \neq s} f_{ij}^{st}}{o(j)} \geq \frac{1}{o(j)deg(s)} + \frac{\sum_{(i,j) \in A: i \neq s} f_{ij}^{st}}{o(j)} \geq \frac{1}{(deg(j) - 1)deg(s)}$$

since  $(j, k) \in SP_{st}$ . Similarly, consider the SP arc  $(k, h) \in SP_{st}$ . Then, for  $f \in \Lambda$  and node  $k$ , the ECMP rule ensures that

$$\begin{aligned} f_{kh}^{st} &= \frac{f_{jk}^{st} + \sum_{(i,k) \in A: i \neq \{s, j\}} f_{ik}^{st}}{o(k)} \\ &\geq \frac{1}{o(k)(deg(j) - 1)deg(s)} + \frac{\sum_{(i,k) \in A: i \neq \{s, j\}} f_{ik}^{st}}{o(k)} \\ &\geq \frac{1}{(deg(k) - 1)(deg(j) - 1)deg(s)}. \end{aligned}$$

Finally, for the arc  $(l, t) \in SP_{st}$ , we will have

$$f_{lt}^{st} \geq \frac{1}{deg(s) \prod_{i \in \mathcal{N}(SP_{st})} (deg(i) - 1)}$$

where  $\mathcal{N}(SP_{st})$  is the set of nodes incident to some SP arcs in  $SP_{st}$ . In the worst case,  $d_{st}$  may visit all nodes in  $V$  before it reaches its destination node  $t$ . This can hold true for any pair of nodes  $i$  and  $j$  such that  $(i, j)$  is an SP arc, which leads to the desired conclusion.  $\square$

In our computational experiments, we have used (73) rather than (70). This is mainly because unlike (70), the inequalities (73) ensure that the flow on an SP arc  $(i, j)$  is positive, and fewer nonzeros are introduced in the constraint matrix. We have observed an improvement in the performance of the B&P algorithm for the set of instances we have worked on. However, using (70) and (73) together would be useful especially for more dense or larger instances since neither of them dominates the other one all the time.

## 6 Computational experiments

In order to test our models as well as the B&P algorithm, we have considered two well known demand uncertainty definitions. The common property of these approaches is that we do not make any assumption about the distribution of the traffic demands or how pairwise demands are correlated with each other. For the rest of this section we let  $W \subseteq V$  be the set of demand and/or supply nodes, which we call *terminal* nodes. Moreover,  $Q = \{(s, t) : s, t \in W, s \neq t\}$  is the set of directed demand pairs with flow demands  $d_{st}$ .

## 6.1 Hose model

This uncertainty model has been introduced by Duffield et al. [11] within the context of Virtual Private Network (VPN) design. Here, the focus is on the outgoing and incoming demands of *terminal* nodes: the set of possible demands is defined by bounds on the total flow each terminal node can exchange with the other terminals:

$$D = \{d \in \mathbb{R}_+^{|Q|} : \sum_{t \in W \setminus \{s\}} d_{st} \leq b_s^+, \sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s^-, \forall s \in V\} \quad (74)$$

where  $b_s^-$  and  $b_s^+$  are the ingress and egress capacities of the terminal node  $s \in W$ , respectively. This is more known as the *asymmetric* Hose model, and there is a symmetric version where an upper bound is given on the sum of all traffic demands originating or ending in  $s$ .

## 6.2 Bertsimas-Sim (BS) uncertainty model

Consider the case where lower and upper bounds are given for the pairwise demands. In several Network Design problems under uncertainty, considering only this bounding box allows for a very conservative solution, which assumes that all demands can get their peak levels simultaneously. To overcome this problem, a positive integer  $\Gamma$  is used to scale the trade off between the robustness of the model and the conservatism level of the solution. This is the robust optimization approach discussed by Bertsimas and Sim [6, 7]. For our problem,  $\Gamma$  is the maximum number of pairs whose demands would change simultaneously within their uncertainty limits so as to affect the solution adversely. Let us assume that demand  $d_{st}$  ranges between  $d'_{st}$  and  $d'_{st} + \hat{d}_{st}$  (where  $\hat{d}_{st} > 0$ ) and that not more than  $\Gamma$  demands may differ from their nominal value  $d'_{st}$  simultaneously. We can define each demand as  $d_{st} = d'_{st} + \beta_{st}\hat{d}_{st}$ , where  $\beta_{st}$  is a binary variable, and impose that  $\sum_{(s,t) \in Q} \beta_{st} \leq \Gamma$ . Since  $\beta_{st} = \frac{d_{st} - d'_{st}}{\hat{d}_{st}}$ , if we relax integrality of  $\beta$ , the BS uncertainty model defines the polyhedral set of possible demands as follows:

$$D = \{d \in \mathbb{R}^{|Q|} : d'_{st} \leq d_{st} \leq d'_{st} + \hat{d}_{st} \quad \forall (s,t) \in Q; \sum_{(s,t) \in Q} \frac{d_{st} - d'_{st}}{\hat{d}_{st}} \leq \Gamma\}. \quad (75)$$

## 6.3 Numerical results

We have performed numerical experiments on instances of various sizes with two purposes: to assess the quality of our formulations and of the B&P algorithm and to compare OSPF and MPLS routing mechanisms. The MPLS oblivious performance ratio under general demand uncertainty is found by solving the linear program (30)-(32). Denote with  $z_{mpls}$  and  $z_{ospf}$  the oblivious performance ratios for MPLS routing and OSPF routing with ECMP, respectively. As MPLS is not restricted to route demands on shortest paths,  $z_{mpls} \leq z_{ospf}$  for a given polyhedron of demands. Fortz and Thorup [12] compare the optimal OSPF routing with the optimal MPLS routing for a *fixed* traffic demand and state that their performances almost match in this case. We provide below the results of experimental campaign that generalizes the comparison to a scenario of demand uncertainty.

We have collected the information (network topology,  $V$  and  $E$ ) of some instances from the IEEE literature (*bhvac*, *pacbell*, *eon*, *metro*, and *arpanet*), and from the Rocketfuel project [19] (*Exodus* (Europe), *Abovenet* (US), *VNSL* (India), *Telstra* (Australia)). For the latter, the current link weights ( $w$ ) and the number of data packets entering and leaving each node are available. For these instances we have assumed that the weight metric  $w$  obeys the inverse capacity weight setting where the weight of each link is inversely proportional to its capacity, i.e.,  $c_{ij} = 1/w_{ij} \quad \forall \{i,j\} \in E$ . Due to the scarce availability of traffic demand data, we have used the Gravity model (Applegate and Cohen [2]) to generate the demand polyhedra  $D$  for all instances. We assume that a demand  $d_{st}$  is proportional to the product of a *repulsion* term  $R_s$  and an *attraction* term  $A_t$ , associated with the source and the destination, respectively. These parameters may correspond to the total observed outgoing and incoming traffic for each node,

respectively, or the population of the city each node refers to. The uncertainty polyhedron is constructed around a *base demand*  $\bar{d} = (\bar{d}_{st})_{(s,t) \in Q}$  where  $\bar{d}_{st} = \beta R_s A_t$ , and  $\beta$  is computed such that  $\bar{d}$  is routable and to choose how close  $\bar{d}$  is to the boundary of the feasibility region. We define a parameter  $\varsigma \in [0, 1]$  such that  $\beta = \varsigma v^*$  with

$$v^* = \max v \quad (76)$$

$$\text{s.t.} \quad \sum_{j: \{s,j\} \in E} (g_{sj}^{st} - g_{js}^{st}) = v R_s A_t \quad \forall (s, t) \in Q \quad (77)$$

$$\sum_{j: \{i,j\} \in E} (g_{ij}^{st} - g_{ji}^{st}) = 0 \quad \forall i \in V \setminus \{s, t\}, (s, t) \in Q \quad (78)$$

$$\sum_{(s,t) \in Q} (g_{ij}^{st} + g_{ji}^{st}) \leq c_{ij} \quad \forall \{i, j\} \in E \quad (79)$$

$$g_{ij}^{st} \geq 0 \quad \forall (i, j) \in A, (s, t) \in Q. \quad (80)$$

This is equivalent to fixing a *direction* (the half-line  $\bar{d}_{st} = \beta R_s A_t$ ,  $\beta \geq 0$ ) on which  $\bar{d}$  must lie, and solving the LP above to find the most critical demand value, which is on the boundary of the feasibility region. Then,  $\varsigma$  scales this value so that  $\bar{d}$  is an interior point of the demand polyhedron if  $\varsigma < 1$ . As a result,  $(d_{st})_{(s,t) \in Q}$  is a feasible traffic matrix for the current topology such that the maximum congestion is no more than  $\varsigma$ .

For the Hose and BS uncertainty models, we have determined the set of terminal nodes  $W$  among the busiest nodes, i.e., those with large  $R_i$  and  $A_i$  parameters. Our instances are *dense* since in all but two cases we have  $|W|/|V| \geq 0.33$ . We have created four variants of each instance using a different value for the uncertainty parameter  $p \in \{1.1, 2, 5, 20\}$  for the BS model. We set  $d'_{st} = \bar{d}_{st}/p$  and  $\hat{d}_{st} = (p - \frac{1}{p})\bar{d}_{st}$ . We refer to each BS instance using the label  $(name, p)$ , e.g.,  $(nsf, 2)$  is the *nsf* instance with  $p = 2$ . Larger  $p$  values imply higher variation in demand estimates, and hence an expectedly larger oblivious ratio. We have chosen  $\Gamma$  such that  $\Gamma/|Q| \in \{0.07, 0.15\}$  in all but one instance. We have randomly picked a subset  $S$  of  $W$  such that  $|S| = \lceil |W|/2 \rceil$ . Then we have used  $b_s^+ = (\sum_{(s,t) \in Q} \bar{d}_{st})/1.1 \forall s \in S$ ,  $b_s^+ = 1.1(\sum_{(s,t) \in Q} \bar{d}_{st}) \forall s \in W \setminus S$ ,  $b_s^- = 1.1(\sum_{(s,t) \in Q} \bar{d}_{st}) \forall s \in S$ , and  $b_s^- = (\sum_{(s,t) \in Q} \bar{d}_{st})/1.1 \forall s \in W \setminus S$  as the maximum out- and inflow capacities of the terminal nodes in the Hose model. It is worth noting that the uncertainty set is asymmetric in this case, which is believed to complicate the problem based on the VPN design literature (Altm et al. [1]).

The numbers of variables and constraints are at most  $O(|V|^5)$  and  $O(|V|^6)$ , respectively, in both the flow and the initial tree formulations. For instance, with BS uncertainty there are 635, 654 constraints and 177, 843 variables (74 general integer and 1110 binary) in the flow model of the *eon* instance whereas we have 485, 760 constraints and 127, 601 variables (100 general integer and 1000 binary) for the *arpanet* case. We provide some statistics on our Branch-and-Price algorithm in Table 1. In the table,  $p$  is the uncertainty level parameter for the BS case (an entry 1.1 – 20 in this column means the B&P statistics are the same for all values of  $p$ );  $|\Omega|$  is the number of  $\tau$  variables, i.e., *SP trees* generated; *depth* is the maximum depth of the B&P tree (0 meaning the problem was solved at the root node); *max-rows* is the maximum number of rows in an LP solved in any B&P node, and *#LPs* is the total number of LP problems solved.

We have used AMPL to model the flow formulation and the MPLS routing and solved them with the Cplex 9.1 MIP solver. The B&P algorithm is implemented in C using MINTO [18] and Cplex 9.1 as LP solver. We have set a two hours time limit both for AMPL and MINTO. Our test results are summarized in Table 2 and Table 3 with:

- the instance characteristics, i.e., the name of the instance, the numbers of nodes, arcs, and terminals, as well as the value of  $p$  for the BS case;
- the solution  $z_{tree}$  and CPU time  $t_{tree}$  of the B&P algorithm;
- the solution  $z_{flow}$  and CPU time  $t_{flow}$  of the flow formulation;

Instance	p	$ \Omega $	depth	max-rows	#LPs
BS model					
Exodus	1.1 - 20	7	0	5353	1
nsf	1.1	8351	70	5955	8101
	2	6308	60	5945	5921
	5	9312	46	5931	4676
	20	3882	27	5912	2630
VNSL	1.1	3	0	2191	1
	2	579	20	2211	694
	5	359	20	2211	308
	20	622	25	2216	477
metro	1.1	20	1	78,010	7
Telstra	1.1	7	0	179,351	1
	2	7	0	179,351	1
	5	381	19	179,370	302
	20	2288	34	179,385	2129
Pacbell	1.1	61	1	44,780	14
	2	51	2	44,781	12
	5	48	0	44,779	7
	20	46	0	44,779	6
bhvac	1.1-20	11	0	127,017	1
Hose model					
Exodus		7	0	5353	1
VNSL		35	22	2213	93
example		1265	65	7239	2358
bhvac		33	0	127,017	7

Table 1: Sample B&P statistics.

- the solution  $z_{mpls}$  and CPU time  $t_{mpls}$  for the MPLS routing.

All run times are given in seconds.

The OSPF routing problem we focus on is clearly different from the regular OSPF routing with fixed link metric. Applegate and Cohen [2] call this more complicated routing effort as *best OSPF style* routing and mention that it is highly non-trivial. Therefore, some instances could not be solved to optimality at the end of 2 hours time limit. In those cases for which we could find a feasible but not an optimal solution of the corresponding problem, we put a \* next to this upper bound. If no feasible solution is available, then the best lower bound obtained by solving the associated LP relaxation is given in parentheses. A “NoI” means that we do not even have a feasible solution for the LP relaxation. Finally, we label with *MA* the *Telstra* instance in Table 3, which MINTO could not solve due to excessive memory requirements.

The  $z_{tree}$ ,  $z_{flow}$ , and  $z_{mpls}$  columns provide relative performance measures for the corresponding routings. They indicate how much each routing deviates from the optimal oblivious routing for the corresponding  $D$ . Hence, as specified in our mathematical models these values are at least 1 where larger numbers imply larger deviation from the best possible routing tailored for that instance. A value of 1 means that by using our optimization tools we find a perfectly oblivious routing, which is the best tailored for any feasible traffic matrix in  $D$ .

Table 2 shows the results for the BS case for 11 instances of 4 different uncertainty levels. As expected, oblivious ratios never get smaller as the variability increases. MINTO and Cplex could solve 19 and 17 of these 44 instances to optimality in 2 hours, respectively. In the remaining cases neither MINTO nor Cplex is clearly superior to the other. Cplex outperforms

MINTO for the *nsf* instances. For example, Cplex solves the flow formulation of (*nsf*,5) in 94 seconds whereas the B&P algorithm stops with a B&P tree of 46 maximum depth yielding a loose upper bound after solving 4676 LP problems. On the other hand, our B&P method finds the perfectly oblivious OSPF routing for (*example*,1.1), (*bhvac*,1.1), (*bhvac*,2), (*bhvac*,5), (*Abovenet*,2), (*Abovenet*,5), and (*Abovenet*,20) in around one minute. Cplex could only find very loose upper bounds for the *Abovenet* instances and just lower bounds for the remaining four. A specialized algorithm as our B&P is hence necessary for this type of problem, although this problem has considerable memory requirements with mid-to-large network instances.

A comparison of the OSPF and MPLS routings based on our test results should be made in two stages. In the first part, we focus on the 24 instances for which we could find an optimal solution and compare the gap for the oblivious ratios. In 15 of them we could find the perfectly oblivious routings with both routing protocols. For the remaining 9, the oblivious ratio of our OSPF routing is 5.4% to 47% larger than that of the oblivious MPLS routing. An important observation here is that the gap between two alternatives does not improve with  $p$  and the deviation for OSPF at uncertainty level  $p$  is almost never less than the one for a smaller  $p$  for any network. For example, consider the *nsf* instance for which the oblivious MPLS routing performs strictly better in all of the four uncertainty levels. A comparison of the three routing technologies, namely our *best OSPF style* routing, MPLS routing, and OSPF under inverse capacity weight setting with ECMP, is provided in Figure 2.

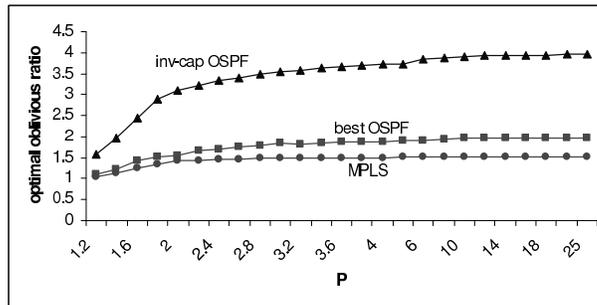


Figure 2: The change in optimal solutions of the *best OSPF style*, MPLS, and inverse capacity OSPF routings for the *nsf* network with different values of  $p$ .

First, notice the significant difference between the *best OSPF style* routing and the OSPF in inverse capacity weight environment. This is a very good example to depict the benefit of using weight management. As is clear from Figure 2, weight management resulted in an improvement in the OSPF performance. A more concrete comparison of the three routing alternatives is given in Figure 3, which shows the gaps between the optimal performance ratios. Inverse capacity OSPF routing is almost 100% worse than *best OSPF* in all higher uncertainty levels for the *nsf* network, while the gap between *best OSPF* and MPLS increases with  $p$  from 8% to 30%. Finally, due to the increasing demand uncertainty, the performances of MPLS, *best OSPF*, and inverse capacity OSPF routings degrade by 32%, 43.6%, and 60.4%, respectively. The degradation in oblivious ratio with uncertainty is already expected. Additionally, these observations certify that the effect is more significant for both OSPF routing strategies.

Finally, we compare the best upper bounds we obtain for the OSPF routing with the optimal solutions for MPLS. The gaps are more variable for those instances and range from 3.7% to 335.7%. Just like the previous comment, the deviation is larger for more uncertain as well as more difficult instances.

Table 3 shows the test results with the Hose model. It is apparent from the running times that these optimization problems are more difficult than those with the BS uncertainty model. For the instances *eon* and *arpanet* we could not even solve the root relaxations in two hours.

The solution times for the tree and flow formulations are comparable for relatively smaller

Instance	N	E	W	p	$z_{tree}$	$t_{tree}$	$z_{flow}$	$t_{flow}$	$z_{mpls}$	$t_{mpls}$
Exodus	7	12	7	1.1	1	0.06	1	0.05	1	0.05
				2	1	0.05	1	0.04	1	0.05
				5	1	0.05	1	0.04	1	0.04
				20	1	0.04	1	0.04	1	0.04
nsf	8	20	5	1.1	1.168*	2 hrs	1.05*	2 hrs	1.013	0.37
				2	2.045*	2 hrs	1.556	3821.53	1.44	0.75
				5	3.808*	2 hrs	1.904	94.33	1.423	0.98
				20	3.936*	2 hrs	1.976	241.10	1.462	1.05
VNSL	9	22	3	1.1	1.066	4.02	1.066	0.19	1	0.02
				2	1.066	23.56	1.066	0.14	1	0.02
				5	1.066	14.67	1.066	0.22	1	0.02
				20	1.066	9.24	1.066	0.30	1	0.02
example	10	30	4	1.1	1	0.11	(1)	2 hrs	1	0.28
				2	1	0.15	1	1900.19	1	0.41
				5	2.25*	2 hrs	1.82*	2 hrs	1.034	0.55
				20	2.575*	2 hrs	3.269*	2 hrs	1.079	0.78
metro	11	84	5	1.1	4.357*	2 hrs	(1)	2 hrs	1	92.97
				2	(1.211)	2 hrs	(1.211)	2 hrs	1.210	450.96
				5	(2.192)	2 hrs	(1.299)	2 hrs	1.299	4642.34
				20	(1.648)	2 hrs	(1.306)	2 hrs	1.302	3577.76
bhvac	19	44	11	1.1	1	109.63	(1)	2 hrs	1	81.18
				2	1	120.03	(1.001)	2 hrs	1	23
				5	1	41.32	(1)	2 hrs	1	44.23
				20	(1.706)	2 hrs	(1.001)	2 hrs	1.443	1130.53
Abovenet	19	68	5	1.1	1	12.78	1	60.78	1	12.48
				2	1	13.58	2.243*	2 hrs	1	35.95
				5	1	13.92	2.687*	2 hrs	1	54.06
				20	1	16.31	5.357*	2 hrs	1	46.35
Telstra	44	88	7	1.1	1	1.75	1	0.50	1	0.16
				2	1	1.79	1	0.41	1	0.16
				5	2.075*	2 hrs	1.054	2.56	1	0.16
				20	2.081*	2 hrs	1.886	2.39	1.283	0.18
pacbell	15	42	7	1.1	1.667*	2 hrs	1.283*	2 hrs	1.014	70.93
				2	1.868*	2 hrs	(1.249)	2 hrs	1.249	134
				5	(1.521)	2 hrs	(1.489)	2 hrs	1.488	174.29
				20	(1.565)	2 hrs	(1.541)	2 hrs	1.54	159.54
eon	19	74	15	1.1	(1)	2 hrs	NoI	2 hrs	NoI	2 hrs
				2	(1)	2 hrs	NoI	2 hrs	4.433*	2 hrs
				5	(4.718)	2 hrs	NoI	2 hrs	NoI	2 hrs
				20	(6.411)	2 hrs	NoI	2 hrs	6.87*	2 hrs
arpanet	24	100	10	1.1	(1.313)	2 hrs	NoI	2 hrs	1.017	492.85
				2	(1.922)	2 hrs	NoI	2 hrs	4.4*	2 hrs
				5	(4.993)	2 hrs	NoI	2 hrs	NoI	2 hrs
				20	(5.799)	2 hrs	NoI	2 hrs	NoI	2 hrs

Table 2: Results for the BS uncertainty model.

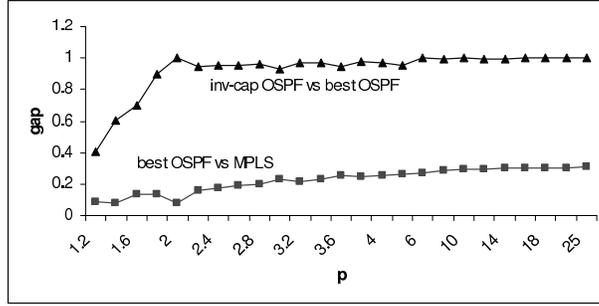


Figure 3: Comparison of the *best OSPF style* routing with MPLS and OSPF under inverse capacity weight setting for the *nsf* network for different values of  $p$ .

instances like *Exodus* and *VNSL* where the optimal oblivious ratios are found. *Exodus* is solved to optimality at the root node of the B&P tree using the initial set of *SP trees* whereas for *VNSL*, depth of the B&P tree is 22 and the optimal solution is obtained after solving 93 LP problems and generating 26 new *SP trees*. The B&P algorithm had to stop due to excessive memory requirements for *Telstra* and because of time limit for *nsf* and *example* providing upper bounds on the optimal oblivious ratios of our *best OSPF style* routing. These bounds are worse than the bounds provided by the flow formulation under the same settings, although the tree formulation gives better lower bounds at the end of 2 hours for the remaining 6 instances.

The difference between the OSPF and MPLS routings is more evident for the Hose model. For *Exodus* we could find the perfectly oblivious routing with both protocols. However, a comparison between the optimal solutions of the instances *nsf*, *VNSL*, *example*, and *Telstra* shows that the difference between the two alternatives are 31.8%, 6.6%, 85.4%, and 50%, respectively. In brief, the average gap between the optimal solutions of the two routing schemes is 34.8% for the Hose model and 6.5% for the BS model. The Hose model relies on the estimates for the total inflow and outflow capacities of the routers whereas for the BS case we need an estimate for the lower and upper bounds on the individual demands. Thus we can say that the definition of the traffic polyhedra  $D$  is looser in the former<sup>2</sup>. Therefore, we believe that these average deviations between the two protocols support our remark that degradation of the network performance due to increased uncertainty is higher for OSPF routing.

Instance	N	E	W	$z_{tree}$	$t_{tree}$	$z_{flow}$	$t_{flow}$	$z_{mpls}$	$t_{mpls}$
Exodus	7	12	7	1	0.04	1	0.05	1	0.03
nsf	8	20	5	4*	2 hrs	2	2730.38	1.517	0.40
VNSL	9	22	3	1.066	8.77	1.066	0.30	1	0.16
example	10	30	4	2.7*	2 hrs	2.2*	2 hrs	1.079	0.42
metro	11	84	5	(1.437)	2 hrs	(1.302)	2 hrs	1.302	1657.83
bhvac	19	44	11	(2.853)	2 hrs	(1.515)	2 hrs	(1.515)	2 hrs
Abovenet	19	68	5	(1.116)	2 hrs	(1.116)	2 hrs	1.045	326.13
Telstra	44	88	7	2.081*	MA	1.925	1.22	1.283	0.08
pacbell	15	42	7	(1.544)	2 hrs	(1.543)	2 hrs	1.543	59.13
eon	19	74	15	(6.857)	2 hrs	NoI	2 hrs	NoI	2 hrs
arpanet	24	100	10	(5.85)	2 hrs	NoI	2 hrs	NoI	2 hrs

Table 3: Results for the hose uncertainty model.

Our final comment is about the benefit of considering a polyhedra of demands rather than a

<sup>2</sup>Based on how we have determined  $b_s^+$  and  $b_s^-$  as well as  $d'_{st}$  and  $\hat{d}_{st}$  for the Hose and BS instances respectively given the same average pairwise demand estimates  $\bar{d}_{st}$ .

single traffic matrix  $\bar{d}$  of average demands. To make such a comparison we use  $\frac{MaxU_{\bar{d}}^{f^*}}{BEST_{\bar{d}}}$ , where  $f^*$  is the optimal oblivious OSPF routing in a given instance and  $BEST_{\bar{d}}$  is the maximum link utilization of the most fair routing, say  $f_{\bar{d}}$ , for the average demand  $\bar{d}$ . First, such a comparison does not provide additional information in those instances where we could find the perfectly oblivious routing. We already know that the most fair routing for *any* traffic matrix in  $D$  is attained in such cases. Hence we focus on the remaining examples and we have observed that it is not possible to make a conclusion that is valid for all cases. For example, in the *VNSL* instances the optimal routing for  $\bar{d}$  is different than  $f^*$ , while the opposite happens for *nsf*. This means that if we optimize just for the mean demand and the current demand turns out to be a different one, then we might have  $f_{\bar{d}}$  perform significantly worse than  $f^*$ . Thus, optimizing just for the mean demands does not ensure the fair allocation of work load in all cases.

## 7 Conclusion

Current traffic engineering efforts are mostly based on the efficient use of network resources so as to route a given traffic matrix. In practice the demands are not likely to be known exactly. We have proposed two mixed-integer models obtained by a duality-based reformulation for our problem. The first is a compact formulation based on flow variables. Because this model gets large very rapidly even for medium sized problems, we have proposed an alternative tree formulation based on special structured subgraphs of the original network, and solved it with a branch&price (B&P) algorithm supported by cutting planes.

From the tests performed with our models and the B&P algorithm on two traffic uncertainty definitions, the Hose and the BS model, we have observed that it pays to create a specialized B&P algorithm especially for the BS uncertainty case. These tests also prove that an efficient optimization of OSPF weights leads to a substantial improvement in the network performance in terms of link utilization. Although the shortest-path constraints limit such a performance behind that of more advanced routing technologies such as MPLS, the gap between the two routing protocols can be decreased even for scenarios where traffic demands are very uncertain.

## References

- [1] A. Altın, E. Amaldi, P. Belotti, and M. Ç. Pınar, Provisioning virtual private networks under traffic uncertainty. *Networks* 20 (1), 2007, pp. 100-115.
- [2] D. Applegate and E. Cohen, Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In proceedings of *SIGCOMM '03*, pp. 313-324.
- [3] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance, Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46, 1998, pp. 316-329.
- [4] P. Belotti and M.Ç. Pınar, Optimal oblivious routing under linear and ellipsoidal uncertainty. *Optimization and Engineering* 9 (3), 2008, pp. 257-271.
- [5] W. Ben-Ameur and H. Kerivin, Routing of uncertain traffic demands. *Optimization and Engineering*, 3, 2005, pp. 283-313.
- [6] D. Bertsimas and M. Sim, Robust discrete optimization and network flows. *Mathematical Programming*, Ser. B 98, 2003, pp. 43-71.
- [7] D. Bertsimas and M. Sim, The price of robustness. *Operations Research*, 52, 2004, pp. 35-53.

- [8] A. Bley and T. Koch, Integer programming approaches to access and backbone IP-network planning. Tech. Report TR 02-41, ZIB institute, Berlin, 2002.
- [9] P. Broström and K. Holmberg, Design of IP/OSPF networks using a Lagrangean heuristic on an in-graph based model. In proceedings of *INOC 2005*, B3, pp.702.
- [10] L. De Giovanni, B. Fortz, and M. Labbé, A lower bound for the internet protocol network design problem. In proceedings of *INOC 2005*, B2, pp. 401-408.
- [11] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J.E. van der Merive, A flexible model for resource management in virtual private networks. In proceedings of *ACM SIGCOMM*, 1999, pp. 95-108.
- [12] B. Fortz and M. Thorup, Internet traffic engineering by optimizing OSPF weights. In proceedings of *IEEE INFOCOM 2000*, pp. 519-528.
- [13] K. Holmberg and D. Yuan, Optimization of internet protocol network design and routing. *Networks* 43(1), 2004, pp. 39-53.
- [14] F. Y. Lin and J. L. Wang, Minimax open shortest path first routing algorithms in networks supporting SMDS service. In proceedings of *IEEE Int. Conf. Communications (ICC)*, 2, 1993, pp. 666-670.
- [15] G. Oriolo, Domination in traffic matrices. *Mathematics of Operations Research* 33 (1), pp. 91-96, 2008.
- [16] A. Parmar, S. Ahmed, and J. Sokol, An integer programming approach to the OSPF weight setting problem. Available for download at <http://www2.isye.gatech.edu/~sahmed/publications.html>.
- [17] M. Pióro, Á. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, and S. Kozdrowski, On open shortest path first related network optimization problems. *Performance Evaluation*, 48, 2002, pp. 201-223.
- [18] M.W.P. Savelsbergh, G.C. Sigismondi, and G.L. Nemhauser. A functional description of MINTO, a Mixed INTEger Optimizer. *OR Letters* 15, 1994, pp. 47-58.
- [19] N. Springs, R. Mahajan, D. Wetherall, and T. Anderson, Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking* 12 (1), 2004, pp. 2-16.
- [20] A. Sridharan, R. Guérin, and C. Diot, Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks. *IEEE INFOCOM 2003*, San Francisco, CA.
- [21] A. Tomaszewski, M. Pióro, M. Dzida, and M. Zagożdżon, Optimization of administrative weights in IP networks using the branch-and-cut approach. In proceedings of *INOC 2005*, B2, pp. 393-400.
- [22] J. Wang, Y. Yang, L. Xiao, and K. Nahrstedt, Edge-based traffic engineering for OSPF networks. *Computer Networks*, in press.
- [23] Y. Wang, Z. Wang, and L. Zhang, Internet traffic engineering without full mesh overlaying. In proceedings of *IEEE INFOCOM 2001*, pp. 565-571.