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

Ayşegül 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@andrew.cmu.edu

Abstract

We consider the *best OSPF style* routing problem in telecommunication networks, where weight management is employed to get the optimal routing configuration with the minimum oblivious ratio. We incorporate polyhedral demand uncertainty into the problem so that the performance of each routing is assessed on its worst case congestion ratio for any feasible traffic matrix in the polyhedron of demands. The problem is an accurate reflection of real world IP networks not just 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., OSPF with equal load sharing. As the OSPF routing problem with equal split is NP-hard even for a fixed demand matrix, the problem considered in the paper is also very difficult. First, we prove that the optimal oblivious MPLS routing under polyhedral traffic uncertainty can be obtained in polynomial time using a duality-based reformulation. Then we consider the OSPF routing with ECMP in case of general traffic uncertainty, and present a compact mixed-integer linear programming formulation based on flow variables. We propose an alternative tree formulation along with a specialized Branch-and-Price algorithm as an exact solution tool. Finally, we report and discuss test results for several network instances.

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

1 Introduction

The importance of effective traffic engineering for today's highly information dependent economy should not be underestimated. Hence, the configuration of an 'effective' routing strategy to achieve a high customer satisfaction and the efficient use of network resources is crucial. Different routing protocols like Multi-Protocol Label Switching (MPLS), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP) etc. can be used to tell routers the best paths to use whereas miscellaneous criteria can be used to determine these paths. We will consider the OSPF routing protocol with a 'fairness' criterion, which is about the optimization of network utilization through a 'fair' allocation of the traffic load among the links of the available shortest paths. Particularly, we look for a general OSPF routing strategy which is fair for a set of traffic demands, i.e., an optimal oblivious OSPF routing scheme.

Open Shortest Path First (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. Routers send the traffic between all nodes in an internetwork along the

*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.

corresponding shortest paths composed of *available* links of the underlying network. These shortest paths are determined based on a metric established prior to network operations. 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 (Giovanni et al. [17]). On the other hand, the management of link weights so as to optimize a design and routing criterion is the focus of the most recent references (Parmar et al. [3], Tomaszewski et al. [5], Fortz and Thorup [6], Holmberg and Yuan [16], Pióro et al. [18], and Wang et al. [24]).

The 'fairness' of a routing can be measured by the utilization (i.e., the fraction of capacity used by data flow) of the most congested link. If some 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. On the other hand each routing is assessed irrespective of a specific traffic demand when there is a set of feasible demands rather than a single one. Such a routing is called *oblivious* since the path between every node pair is chosen independent of the current demand matrix. To sum up, the goal of oblivious routing is to find a set of fair routing paths for all source-sink pairs regardless of the demand matrix.

Since it is not likely for traffic engineers to estimate the traffic demands with certainty in advance, considering some level of uncertainty in the definition of demand matrices would strengthen the traffic engineering efforts. Applegate and Cohen [8] study oblivious routing in the case of very limited information of traffic demands. More recently Belotti and Pınar [22] consider box and ellipsoidal uncertainty representations. They focus on the case where traffic demand is assumed to have some lower and upper bounds as well as the case where the mean-covariance information of random demand is available. In the present paper, we consider the case of polyhedral demand uncertainty where the possible traffic matrices are assumed to lie in a polyhedron, defined by a set of linear inequalities.

The problem considered in this paper is a *best OSPF style* routing problem. We incorporate weight management into our analysis based on the common belief that OSPF might lead to unsatisfactory network performance without a good traffic engineering. Naturally, the freedom of defining a clever weight setting to optimize any design criterion would not deteriorate the effectiveness of OSPF. Moreover, we have used a general definition of the set of feasible traffic matrices in deference to the difficulty of having an exact estimate of the demands in real life. Finally, we apply the Equal Cost Multi-Path Protocol (ECMP) rule, which complies with the current forwarding technology. It is worth mentioning that these specifics of our problem make our models practically feasible. Moreover, the added flexibility via weight management and general demand definition improves the effectiveness of the OSPF routing.

There has been a lot of research to accomplish effective traffic engineering. Different routing strategies as well as various ways to manage them have been proposed. However, given the difficulty of the problem, some simplifications had to be made. The most common one is the assumption of a given demand matrix. Then again there is agreement among researchers that weight management is crucial to improve the performance of OSPF routing, and hence weight metric is not supposed to be given. Unfortunately, it is not trivial to determine a metric consistent with the capabilities of today's traffic forwarding technology and thus various strategies for controlling the weight metric are proposed. Such a problem can be thought of as a particular *inverse shortest path problem* (Zhang et al [13], Burton et al. [14]).

Weight management under ECMP is NP-hard (Wang et al. [24], Pióro et al. [18], Fortz and Thorup[6]) and the current technology does not support arbitrary load sharing. In order to tackle this difficulty either 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 [2], Tomaszewski et al. [5], and Lin and Wang [11], as the examples for unsplit routing while we refer to Parmar et al. [3], Sridharan et al. [4] and Wang et al. [15] for the latter case. References Parmar et al. [3], Tomaszewski et al. [5], Giovanni et al. [17], Pióro et al. [18], and Broström and Holmberg [23] also show Mixed-Integer modeling examples for incorporating the ECMP rule. In Bley and Koch [2], Pióro et al. [18], and Broström and Holmberg [23] a two-stage algorithm is used where the authors initially find an optimal routing scheme. Then, in the second step they look for a metric that is compatible with the paths found in the first step, namely a metric according to which these paths are shortest paths. The drawback of these approaches is that not all configurations are guaranteed to be realized as shortest paths. Although Wang et al. [24] shows that a class of routes with some property can be converted to shortest-paths, still no complete description of admissible routing schemes is available. Alternatively, Parmar et al. [3], Fortz and Thorup [6], Lin and Wang [11], Giovanni et al. [17], and Wang et al. [24] prefer to consider the optimization of a design criterion and the link metric, simultaneously.

To the best of our knowledge, there is no other work that combines general traffic uncertainty with the oblivious routing problem. We use duality-based reformulations to convert our originally semi-infinite models to their linear counterparts. Hence, we provide a compact linear mixed-integer formulation based on flow variables for the best oblivious OSPF routing problem under weight management. Furthermore, we present an alternative tree formulation using destination-based multiple shortest paths as well as a solution tool based on a specialized Branch-and-Price algorithm, that is strengthened by the inclusion of cutting planes. Also, a relaxation of our flow formulation, which is an extension of the models of Applegate and Cohen [8] and Belotti and Pınar [22], can be used to model the MPLS routing under general demand uncertainty. Hence we show that optimal oblivious MPLS routing can be found in polynomial time. As a result, we can discuss the relative performances of the oblivious OSPF routing and the oblivious MPLS routing under a very general setting where any polyhedral definition of traffic demands can be used. Therefore, we provide a concrete perspective to the discussions on the feasibility and effectiveness of these routing alternatives.

In summary, the present paper makes an important contribution in terms of modeling and applicability of the results. We make no assumption of arbitrary split or known traffic demand, and hence sacrifice neither the practicality nor the generality of the model. Moreover, we avoid two-stage approaches, which do not guarantee to find a shortest path configuration. Furthermore, we focus on the efficient use of network resources so as to improve customer satisfaction by allocating the traffic demand "fairly" among the network links. Above all, we use compact MIP formulations to model this difficult problem and propose a specialized Branch-and-Price algorithm as an exact solution tool.

The rest of the paper is organized as follows. In Section 2 we make some basic definitions and explain the performance measure we will use in our models to assess the goodness of different routings. Then in Section 3 we present our integer programming models for the oblivious routing with general demand uncertainty. Consequently, we show how we incorporate OSPF routing into our models in Section 4. Section 5 discusses our Branch-and-Price algorithm while numerical results are provided in Section 6. Finally we offer conclusions in Section 7.

2 Basic definitions and measures of performance

Consider the undirected graph $G = (V, E)$. All 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. Moreover, we suppose that 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} where 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 source-sink pairs. Although d is defined as a vector, the term *traffic matrix* is obiquitous 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^{st}_{hk}. Then the matrix $f = (f^{st}_{hk})_{(h,k)\in A,(s,t)\in Q}$ defines a *routing* if it satisfies the following conditions:

$$\sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q \tag{1}$$

$$0 \le f^{st}_{hk} \le 1 \qquad \forall (h,k) \in A, (s,t) \in Q \tag{2}$$

and we denote the set of all possible routings on G as Λ. Consequently, the traffic load assigned by $f \in \Lambda$ to the undirected link $\{h,k\} \in E$ for the traffic matrix d is $L^f_d(hk) = \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh})$ whereas its utilization is $U^f_d(hk) = L^f_d(hk)/c_{hk}$. The fairness of a routing, i.e., the measure of how balanced the distribution of a traffic demand d is, can be measured by the maximum link utilization of f ($MaxU^f_d$), that is:

$$MaxU^f_d = \max_{\{h,k\}\in E} U^f_d(hk).$$

Then, the problem of finding the routing with the minimum $MaxU^f_d$ for a fixed TM d is

$$\min_{f\in\Lambda} \{MaxU^f_d\}$$

and it can be modeled as follows:

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

$$\text{s.t.} \quad \sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, \forall (s,t) \in Q \tag{4}$$

$$r \ge \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh})/c_{hk} \qquad \forall \{h,k\} \in E \tag{5}$$

$$\sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) \le c_{hk} \qquad \forall \{h,k\} \in E \tag{6}$$

$$0 \le f^{st}_{hk} \le 1 \qquad \forall (h,k) \in A, (s,t) \in Q \tag{7}$$

where (4) ensures that f is a routing and (5)-(6) imply the existence of a flow, which routes the traffic matrix d respecting the capacity limitations. Notice that (5) and (6) together with the objective of minimizing r imply that $r \le 1$, i.e., the traffic load of each link must be less than its capacity. Therefore, (6) imposes that no link be overloaded.

3 Oblivious routing under polyhedral demand uncertainty

The optimal oblivious routing problem consists in finding a routing for each source-sink pair $(s,t) \in Q$ independent of the traffic matrix d such that the maximum edge utilization is minimized. In this case we have a set of traffic matrices D and the best routing is required to support any feasible traffic matrix $d \in D$ in the most balanced way. Thus, oblivious routing yields a conservative strategy with a worst case approach when the demand is uncertain. As a result, the 'goodness' of a routing is assessed based on a set of matrices where the maximum link utilization of a routing f is the highest ratio it achieves over D, i.e., $\max_{d\in D} MaxU^f_d$. However, a more common approach is to use a measure of how close each f is to optimality for any traffic matrix $d \in D$ (Applegate and Cohen [8], Belotti and Pınar [22]). Then the *oblivious ratio* of f on the set D is

$$OR^f_D = \max_{d\in D} \frac{MaxU^f_d}{BEST_d}$$

where $BEST_d$ is the smallest maximum link utilization ratio for d and is equal to the optimal solution of the linear problem (3)-(7). As a result, the problem of finding the routing with the smallest maximum link utilization for the set D of traffic demands becomes

$$\min_{f\in\Lambda}\max_{d\in D}\frac{\max_{\{h,k\}\in E} U_d^f(hk)}{BEST_d}. \tag{8}$$

Notice that $BEST_d$ does not depend on $\{h,k\}$ and hence $\frac{\max_{\{h,k\}\in E} U_d^f(hk)}{BEST_d}$ 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 (8) to have the equivalent expression

$$\min_{f\in\Lambda}\max_{\{h,k\}\in E}\max_{d\in D}\frac{U_d^f(hk)}{BEST_d}. \tag{9}$$

In the sequel, we can model (9) as the following mathematical model:

$$\min r \tag{10}$$

$$\text{s.t.} \quad r \geq \max_{d\in D}\frac{\sum_{(s,t)\in Q} d_{st}(f_{hk}^{st}+f_{kh}^{st})/c_{hk}}{BEST_d} \qquad \forall\{h,k\}\in E \tag{11}$$

$$(1)-(2) \tag{12}$$

where (12) ensures that f is a routing. Constraint (11) implies that for each link $\{h,k\}\in E$ and routing $f\in\Lambda$, we have a maximization problem over D. Hence the definition of D is important in modeling and solving (10)-(12).

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 (10)-(12). It is instead a variable of the inner optimization model on the right-hand side of constraint (11). Due to the max operator in constraint (11), the model (10)-(12) is equivalent to a semi-infinite optimization model with one constraint (11) for each $d\in D$.

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 [12]), which are defined as those that suffice to describe the entire uncertainty set, or in other words, such that 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 $d'\leq d$. A necessary and sufficient condition for dominance between traffic demands has been given by Oriolo [12]. However, the same does not apply here because the objective function of the inner optimization problem is not linear w.r.t. d, hence for two demands d and d' such that $d'\leq d$ we cannot prove that $\frac{MaxU_{d'}^f}{BEST_{d'}}\leq\frac{MaxU_d^f}{BEST_d}$.

Bearing in mind that the demand uncertainty can be modeled in various ways, we will consider the case of polyhedral uncertainty: traffic demand matrices are not known but are supposed to belong to a polyhedron defined by some linear inequalities specifying the capacity of routers or bounds on the traffic flow between some node pairs etc. Consequently, we consider the general traffic uncertainty model

$$D=\{d=(d_{st})_{(s,t)\in Q}: Ad\leq a,\ d\geq 0,\ d\neq 0\} \tag{13}$$

where $A\in\mathbb{R}^{H\times|Q|}$ and $a\in\mathbb{R}^H$ with H being the number of linear inequalities that define D. We prove that the above semi-infinite optimization model can be reduced to its equivalent linear counterpart by using LP duality. Firstly, notice that we can write (11) as

$$\max_{d\in D}\left\{\sum_{(s,t)\in Q} d_{st}(f_{hk}^{st}+f_{kh}^{st}) - rc_{hk}BEST_d\right\}\leq 0 \quad \forall\{h,k\}\in E. \tag{14}$$

Then the left-hand side of (14) is a maximization problem and we have the following model P_{hk} for each $\{h, k\} \in E$:

$$(P_{hk}) \quad \max \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) - r\omega c_{hk} \tag{15}$$

$$\text{s.t.} \quad \sum_{j:\{s,j\}\in E} (g^{st}_{sj} - g^{st}_{js}) = d_{st} \qquad \forall (s,t) \in Q \tag{16}$$

$$\sum_{j:\{i,j\}\in E} (g^{st}_{ij} - g^{st}_{ji}) = 0 \qquad \forall i \in V \setminus \{s,t\}, (s,t) \in Q \tag{17}$$

$$\sum_{(s,t)\in Q} (g^{st}_{ij} + g^{st}_{ji}) \leq \omega c_{ij} \qquad \forall \{i,j\} \in E \tag{18}$$

$$\omega \leq 1 \tag{19}$$

$$\sum_{(s,t)\in Q} a^{st}_z d_{st} \leq a_z \qquad \forall z = 1, \ldots, H \tag{20}$$

$$g^{st}_{ij} \geq 0 \qquad \forall (i,j) \in A, (s,t) \in Q \tag{21}$$

$$d_{st} \geq 0 \qquad \forall (s,t) \in Q \tag{22}$$

$$\omega \geq 0 \tag{23}$$

where $\omega = BEST_d$ and the traffic polytope D is defined by H linear inequalities of the form (20). Applegate and Cohen [8] assume that, at the optimum of the inner optimization problem (12), $BEST_d = 1$ and hence one of the arcs is assumed to be used to its full capacity in the worst case. However, as Belotti and Pınar [22] show, this is not a valid assumption all the time. They give an example of the case where $D = (d_{st})_{(s,t)\in Q}$ is such that $d_{st} \leq \alpha \frac{\min_{\{h,k\}\in E} c_{hk}}{|Q|} \;\; \forall (s,t) \in Q$ 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. Hence, we avoid such an assumption and use (15), (18), and (19) to model this feature of the problem. Moreover, constraints (16)-(19) ensure that there is a feasible flow g on $G = (V, E)$ that routes demand d without violating the link capacities.

For a given r and routing f, P_{hk} is a linear programming problem, and hence we can employ duality to get the dual problem (DP_{hk}) for each link $\{h, k\} \in E$. Consider the dual variables π^{st}_{hk}, $\sigma^{st}_{i,hk}$, $\eta_{ij,hk}$, χ_{hk}, and λ^{hk}_z of the constraints (16) - (20). If we let

$$\Pi^{st}_{i,hk} = \begin{cases} \pi^{st}_{hk} & \text{if } i = s \\ 0 & \text{if } i = t \\ \sigma^{st}_{i,hk} & \text{otherwise} \end{cases} \qquad \forall i \in V, (s,t) \in Q,$$

then we have:

$$(DP_{hk}) \quad \min \chi_{hk} + \sum_{z=1}^{H} a_z \lambda^{hk}_z \tag{24}$$

$$\text{s.t.} \quad \Pi^{st}_{i,hk} - \Pi^{st}_{j,hk} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A, \; (s,t) \in Q \tag{25}$$

$$-\pi^{st}_{hk} + \sum_{z=1}^{H} a^{st}_z \lambda^{hk}_z \geq f^{st}_{hk} + f^{st}_{kh} \qquad \forall (s,t) \in Q \tag{26}$$

$$-\sum_{\{i,j\}\in E} c_{ij}\eta_{ij,hk} + \chi_{hk} \geq -r c_{hk} \tag{27}$$

$$\eta_{ij,hk} \geq 0 \qquad \forall \{i,j\} \in E \tag{28}$$

$$\chi_{hk} \geq 0 \tag{29}$$

$$\lambda^{hk}_z \geq 0 \qquad \forall z = 1, .., H \tag{30}$$

We use DP_{hk} and the duality theorems to reduce (11) to an equivalent set of linear inequalities.

Proposition 3.1. *For the polyhedral traffic uncertainty model where* $D = \{d = (d_{st})_{(s,t)\in Q} : Ad \leq a,\ d \geq 0,\ d \neq 0\}$ *the right-hand side of the constraint (11) for each* $\{h,k\} \in E$ *can be replaced with the equivalent inequality system (25)-(30) and the inequality*

$$-\chi_{hk} - \sum_{z=1}^{H} a_z \lambda_z^{hk} \geq 0. \tag{31}$$

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

$$\{\min 0 : (25), (26), (27), (28), (29), (30), (31)\}. \tag{32}$$

Let g_{ij}^{st}, d_{st}, ω, and ψ_{hk} be the dual variables associated with constraints (25)-(27), and (31), respectively. Consequently, the corresponding dual problem (FD_{hk}) is

$$\max\{-\omega r c_{hk} + \sum_{(s,t)\in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st})\} \tag{33}$$

$$\text{s.t.} \quad \sum_{j:\{s,j\}\in E} (g_{sj}^{st} - g_{js}^{st}) = d_{st} \qquad \forall (s,t) \in Q \tag{34}$$

$$\sum_{j:\{i,j\}\in E} (g_{ij}^{st} - g_{ji}^{st}) = 0 \qquad \forall i \in V \backslash \{s,t\}, (s,t) \in Q \tag{35}$$

$$\sum_{(s,t)\in Q} (g_{ij}^{st} + g_{ji}^{st}) \leq \omega c_{ij} \qquad \forall \{i,j\} \in E \tag{36}$$

$$\omega \leq \psi_{hk} \tag{37}$$

$$\sum_{(s,t)\in Q} d_{st} a_z^{st} \leq \psi_{hk} a_z \qquad \forall z = 1, .., H \tag{38}$$

$$g_{ij}^{st} \geq 0 \qquad \forall (i,j) \in A, (s,t) \in Q \tag{39}$$

$$d_{st} \geq 0 \qquad \forall (s,t) \in Q \tag{40}$$

$$\omega \geq 0 \tag{41}$$

$$\psi_{hk} \geq 0 \qquad \forall \{h,k\} \in E. \tag{42}$$

Without loss of generality, we can assume that $\psi_{hk} > 0$ since we would have the trivial solution otherwise. Hence, if we scale each variable by ψ_{hk} such that $\tilde{g}_{hk}^{st} = g_{hk}^{st}/\psi_{hk}$, $\tilde{d}_{st} = d_{st}/\psi_{hk}$, and $\tilde{\omega} = \omega/\psi_{hk}$, then (33)-(42) reduces to (15)-(23) as we wanted to show. □

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

$$\min r \tag{43}$$

$$s.t. \quad \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & otherwise \end{cases} \qquad \forall h \in V, (s,t) \in Q \tag{44}$$

$$\chi_{hk} + \sum_{z=1}^{H} a_z \lambda_z^{hk} \leq 0 \qquad \forall \{h,k\} \in E \tag{45}$$

$$\Pi_{i,hk}^{st} - \Pi_{j,hk}^{st} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A,\ (s,t) \in Q,\ \{h,k\} \in E \tag{46}$$

$$-\pi_{hk}^{st} + \sum_{z=1}^{H} a_z^{st} \lambda_z^{hk} \geq f_{hk}^{st} + f_{kh}^{st} \qquad \forall (s,t) \in Q,\ \{h,k\} \in E \tag{47}$$

$$-\sum_{\{i,j\}\in E} c_{ij}\eta_{ij,hk} + \chi_{hk} \geq -rc_{hk} \qquad \forall \{h,k\} \in E \tag{48}$$

$$0 \leq f_{hk}^{st} \leq 1 \qquad \forall (h,k) \in A, (s,t) \in Q \tag{49}$$

$$\eta_{ij,hk} \geq 0 \qquad \forall \{i,j\}, \{h,k\} \in E \tag{50}$$

$$\chi_{hk} \geq 0 \qquad \forall \{h,k\} \in E \tag{51}$$

$$\lambda_z^{hk} \geq 0 \qquad \forall z = 1, \ldots, H,\ \{h,k\} \in E. \tag{52}$$

An important issue that we need to highlight here is that the model (43)-(52) is a linear programming problem. Hence we proved that the optimal oblivious ratio for MPLS routing under general traffic uncertainty can be computed in polynomial time by solving 43)-(52).

Before discussing how the OSPF routing protocol could be incorporated into the oblivious routing problem, it would be useful to mention that there is no condition on how the routes would be chosen to carry the feasible traffic matrices on $G = (V, E)$ in the above model. Moreover, constraints (49) imply 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, 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. This is discussed in the following section.

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. The traditional approach is to fix this metric in advance and determine the shortest paths a priori. The more recent approach is to manage the OSPF metric to optimize a given design criteria. 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 (9), where $BEST_d$ is the oblivious ratio of the best non-OSPF routing, since we want to compare the performance of OSPF with that of the best non-constrained routing.

Naturally, there can be more than one shortest path between a pair of nodes. In terms of routing design, one option is to consider unsplittable 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.

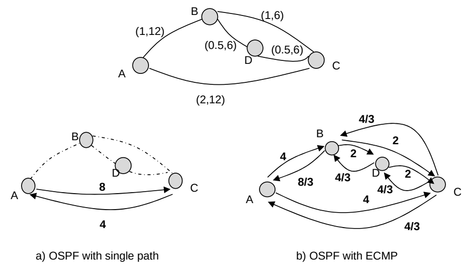

Figure 1: Example for splittable vs unsplittable routing

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 unsplittable routing we would route each demand along a single path. Then 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. On the other hand, if we allow splittable routing, then we would have the case in Figure 1b where the utilization of all links are around 50%. The latter routing is more fair since all links would use almost equal fractions of their capacities. Hence rather than unsplittable routing, it would be better to apply Equal Cost Multi-Path (ECMP) routing in which the demand d_{st} accumulated at some node i is split evenly among all shortest paths between i and t. As a result, we model OSPF routing with ECMP. Two formulations, namely the flow formulation and the tree formulation, will be presented in the rest of this section.

4.1 Variables and parameters

Below we present two formulations to model OSPF routing. Integer variables θ_{ij} define the metric used at each arc (i, j) and range between 1 and Θ_{max}, which is a parameter of value 65535[1]. Although we only need the optimal values of the f and θ variables, some auxiliary classes of variables are needed. We define ρ_i^t as the shortest path distance between i and t according to the metric defined by the θ_{ij} variables. In order to impose ECMP constraints, we use a variable φ_i^{st} which gives the fraction of flow that, after entering node i, is split among different outgoing arcs due to the ECMP rules. 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 the demands are routed on the corresponding shortest paths. We can do so via a set of linear inequalities where the binary variable y_{ij}^t indicates if the arc (i, j) is on some shortest path destined to node t, i.e., if it is a shortest path arc for t. Note that OSPF is a source invariant routing scheme, and this is the reason why we do not need an index for the source node s in y variables. Consequently, we use the constraints

$$f_{ij}^{st} \leq y_{ij}^t \quad \forall (i, j) \in A, (s, t) \in Q \tag{53}$$

to relate y variables to flow variables. Moreover we include

$$y_{ij}^t + \rho_j^t - \rho_i^t + \theta_{ij} \geq 1 \qquad \forall (i, j) \in A, t \in V \tag{54}$$

$$-y_{ij}^t - \frac{\rho_j^t - \rho_i^t + \theta_{ij}}{2\Theta_{max}} \geq -1 \quad \forall (i, j) \in A, t \in V \tag{55}$$

[1]This is the common constant used in the literature when integer link weights are required.

to 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 to t, are dominated by constraints (54), and therefore are not included. If $y_{ij}^t = 1$, then (i,j) is a shortest path arc for all demands destined to t and hence the Bellman condition must be satisfied with equality, as imposed by (54) and (55).

On the other hand, if some arc (i,j) is not a shortest path arc to t according to weights θ, then its reduced cost must be at least 1 since we require $\theta_{ij} \geq 1$. Lastly, any constant larger than $2\Theta_{max}$ can be used in (55). Since for each link $\{i,j\} \in E$ we have $(i,j) \in A$ and $(j,i) \in A$, we use $2\Theta_{max}$ as mentioned in Holmberg and Yuan [16]. Although they explain the reason for such a choice, we restate it here for the sake of completeness. Firstly, for some arc (j,i) we know $\rho_i^t \geq \rho_j^t - \theta_{ji}$ by the Bellman conditions. As a result, for arc (i,j) we have $\rho_j^t - \rho_i^t + \theta_{ij} \leq \rho_j^t - \rho_j^t + \theta_{ji} + \theta_{ij} \leq 2\Theta_{max}$. Finally, we need the following set of constraints since we want to apply the ECMP rule to implement splittable routing:

$$f_{ij}^{st} \leq \varphi_i^{st} \qquad \forall (i,j) \in A, (s,t) \in Q \tag{56}$$

$$1 + f_{ij}^{st} - \varphi_i^{st} \geq y_{ij}^t \quad \forall (i,j) \in A, (s,t) \in Q \tag{57}$$

with the variable bounds

$$1 \leq \theta_{ij} \leq \Theta_{max} \quad \text{integer } \forall (i,j) \in A \tag{58}$$

$$y_{ij}^t \in \{0,1\} \qquad \forall (i,j) \in A, t \in V \tag{59}$$

$$0 \leq \varphi_i^{st} \leq 1 \qquad \forall i \in V, (s,t) \in Q. \tag{60}$$

Constraints (56) and (57) 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 in i equally.

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:*

$$\min r \tag{61}$$

$$s.t. \quad \sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & otherwise \end{cases} \qquad \forall h \in V, (s,t) \in Q \tag{62}$$

$$\chi_{hk} + \sum_{z=1}^{H} a_z \lambda^{hk}_z \leq 0 \qquad \forall \{h,k\} \in E \tag{63}$$

$$\Pi^{st}_{i,hk} - \Pi^{st}_{j,hk} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A,\ (s,t) \in Q,\ \{h,k\} \in E \tag{64}$$

$$-\pi^{st}_{hk} + \sum_{z=1}^{H} a^{st}_z \lambda^{hk}_z \geq f^{st}_{hk} + f^{st}_{kh} \qquad \forall (s,t) \in Q,\ \{h,k\} \in E \tag{65}$$

$$-\sum_{\{i,j\}\in E} c_{ij}\eta_{ij,hk} + \chi_{hk} \geq -rc_{hk} \qquad \forall \{h,k\} \in E \tag{66}$$

$$f^{st}_{ij} \leq y^t_{ij} \qquad \forall (i,j) \in A, (s,t) \in Q \tag{67}$$

$$y^t_{ij} + \rho^t_j - \rho^t_i + \theta_{ij} \geq 1 \qquad \forall (i,j) \in A, t \in V \tag{68}$$

$$-y^t_{ij} - \frac{\rho^t_j - \rho^t_i + \theta_{ij}}{2\Theta_{max}} \geq -1 \qquad \forall (i,j) \in A, t \in V \tag{69}$$

$$f^{st}_{ij} \leq \varphi^{st}_i \qquad \forall (i,j) \in A, (s,t) \in Q \tag{70}$$

$$1 + f^{st}_{ij} - \varphi^{st}_i \geq y^t_{ij} \qquad \forall (i,j) \in A, (s,t) \in Q \tag{71}$$

$$0 \leq f^{st}_{hk} \leq 1 \qquad \forall (h,k) \in A, (s,t) \in Q \tag{72}$$

$$\eta_{ij,hk} \geq 0 \qquad \forall \{i,j\}, \{h,k\} \in E \tag{73}$$

$$\chi_{hk} \geq 0 \qquad \forall \{h,k\} \in E \tag{74}$$

$$\lambda^{hk}_z \geq 0 \qquad \forall z = 1,\dots,H,\ \{h,k\} \in E \tag{75}$$

$$1 \leq \theta_{ij} \leq \Theta_{max} \qquad integer\ \forall (i,j) \in A \tag{76}$$

$$y^t_{ij} \in \{0,1\} \qquad \forall (i,j) \in A, t \in V \tag{77}$$

$$0 \leq \varphi^{st}_i \leq 1 \qquad \forall i \in V, (s,t) \in Q. \tag{78}$$

Notice that if we use the flow formulation (61)-(78) to model OSPF with ECMP we need $2|E|(|V|+1) + |V|^3$ additional variables, $2|E||V|$ of which are binary. Even for medium sized networks, the size of our formulation can get very large and the related solution time would be too long. Hence the time required to solve these problems using MIP solvers is quite long. Therefore, we propose an alternative tree formulation, whose linear relaxation can be solved by column generation, in the next section.

4.3 Alternative formulation

In this section we adopt an alternative approach where we use tree- rather than flow variables to model OSPF routing. In this model each tree variable corresponds to an *SP tree*, which is a widely used structure in OSPF and IS-IS routing protocols. We give a brief explanation of these special structures in the following lines and later proceed with a discussion of our tree formulation in the rest of this section.

4.3.1 Shortest Paths Trees

A Shortest Paths Tree (*SP tree*) is an acyclic graph such that, for at least one metric, all and only paths within the *SP tree* are the shortest ones. In other words, an *SP tree* T with respect to some node t (destination node) of the backbone graph $G = (V, E)$ contains all shortest paths from all other nodes of G to t for a given vector of link weights. Notice that if there are multiple shortest paths from some node $s \in V \setminus \{t\}$ to t, then all of them are included in T. Hence the structure of an *SP tree*, i.e., the set of arcs it contains, is very much affected by the link metric. Therefore we need to underline that an *SP tree* T does not have to be a tree literally since it is the union of some paths. As a result it is important to mention here that an *SP tree* is some acyclic subgraph of G, but not a tree in general.

4.3.2 Tree Formulation

In our problem formulation we use destination based *SP trees* and hence T shows how all the traffic towards its destination node t should be routed on the arcs of the backbone graph. In other words, each *SP tree* T defines a routing configuration for its root node. Thus, we want only one *SP tree* to be used for each $t \in V$. We model this requirement via binary τ_T^t variables, which indicate whether the implicitly defined *SP tree* T is used to route all traffic flow ending at t or not. Bearing in mind that the number of paths in a graph can be exponential in number we define Ω_t as the set of *SP* trees with destination t and Ω_{ij} as the set of *SP* trees containing arc (i, j). Consequently, we can ensure that a single *SP tree* would be used for each destination by the constraint

$$\sum_{T \in \Omega_t} \tau_T^t = 1 \quad \forall t \in V. \tag{79}$$

Moreover, we use the inequality

$$f_{ij}^{st} \leq \sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t \quad \forall (i,j) \in A, (s,t) \in Q \tag{80}$$

to relate the τ variables to flow variables. Notice that (80) is analogous to (53) of the flow formulation. If some flow originated at s and destined to t is routed on arc (i, j), then this arc should be a shortest path arc for t in any *SP tree* T that will be used for it. Hence the sum on the right-hand side of (80) must be 1, which ensures that the *SP tree* for t contains (i, j).

Consequently, we include the OSPF constraints

$$\sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t + \rho_j^t - \rho_i^t + \theta_{ij} \geq 1 \qquad \forall t \in V, (i,j) \in A \tag{81}$$

$$-\sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t - \frac{\rho_j^t - \rho_i^t + \theta_{ij}}{2\Theta_{max}} \geq -1 \quad \forall t \in V, (i,j) \in A, \tag{82}$$

which are analogous to (54) and (55), respectively. Note that the summations in (81) and (82) would be equal to one only for the shortest path arcs ensuring that their reduced costs are zero. The final set of constraints are the following ECMP constraints

$$f_{ij}^{st} \leq \varphi_i^{st} \qquad \forall (i,j) \in A, (s,t) \in Q \tag{83}$$

$$1 + f_{ij}^{st} - \varphi_i^{st} \geq \sum_{T \in \Omega_t \cap \Omega_{ij}} \tau_T^t \quad \forall (i,j) \in A, (s,t) \in Q \tag{84}$$

with the variable bounds

$$1 \leq \theta_{ij} \leq \Theta_{max} \quad \text{integer} \ \ \forall (i,j) \in A \tag{85}$$

$$\tau_T^t \in \{0,1\} \qquad \forall t \in V, T \in \Omega_t \tag{86}$$

$$0 \leq \varphi_i^{st} \leq 1 \qquad \forall i \in V, (s,t) \in Q. \tag{87}$$

The flow and tree formulations are analogous to each other, and the difference is how one tries to solve them. Before discussing our solution approach for the tree formulation, we should make a remark here. As we have mentioned before, the *SP trees* are defined by the weight metric θ, which is also a variable of our model. Hence, we know neither the number nor the structure of *SP trees* explicitly in advance, and we can say the sets Ω_t and Ω_{ij} are implicitly defined.

Consequently, the oblivious routing model discussed in Proposition 3.1 can be combined with one of the flow or path OSPF models to find the optimal OSPF routing under ECMP rule such that the oblivious ratio is minimum.

5 A Branch-and-Price algorithm for exact solution

The number of paths in a graph depends on the structure of the graph, and it can be huge. So can be the number of variables in the tree formulation, hence we have decided to develop a branch-and-price (B&P) algorithm, which is a column generation integrated branch-and-bound technique. This method was initially discussed in Barnhart et al. [7] and it is an efficient approach to cope with those models with a large number of variables. Basically, it is a modified branch-and-bound (B&B) algorithm, which starts with a restricted LP relaxation (RLP_0) with fewer variables than the original problem and applies column generation at each node of the B&B tree. The subproblem in a B&B node (RLP_{curr}) is optimal when no new columns can be 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 in a VPN design problem can be found in Altın et al. [1].

In our problem, we consider destination based *SP trees* comprising shortest paths to each node $t \in V$ from all other nodes of the graph $G = (V, E)$. Just like the number of paths in a graph, the number of *SP trees* can also be very large. Therefore it is wise to use *Branch-and-Price* to solve the tree formulation. We summarize the main steps of our B&P algorithm in Figure 2. The details of the application are addressed in the rest of this section.

As a final remark, note that we use the terms *SP tree T destined at t* and τ_T^t *variable* interchangeably throughout this section.

5.1 Initialization

We start our B&P algorithm with a relaxed formulation RLP_0, whose solution is feasible but not necessarily optimal for the original problem. For the sake of completeness, we define RLP_0 as follows:

```
algorithm B&P;
input: an undirected graph $G = (V, E)$, a traffic polytope $D$, a link capacity vector $\vec{c}$ ;
output: the optimal oblivious ratio in the OSPF routing environment for the given input;

begin
    Initialize:
        Find an initial set $\Omega_0$ of SP trees, i.e., $\tau_T^t$ variables;
        Let $\tilde{\Omega} = \Omega_0$; // $\tilde{\Omega}$ is the current set of SP trees
        Let $S = \{root\}$ ; // $S$ is the set of unevaluated B&B nodes, root is the root node
        Let $UB = \infty$; // $UB$ is the best oblivious ratio obtained so far
    while $S \neq \emptyset$ begin
        Select $n_b \in S$ such that $LB(n_b) \leq LB(n) \; \forall n \in S$
        Let $S = S \backslash \{n_b\}$;
        repeat
            Optimize: Get $z^*(n_b, \tilde{\Omega})$;//optimal value of $RLP_{curr}$
            Price:
                For each $t \in V$ begin
                    Search for a new $\tau_{\hat{T}}^t$ variable, i.e., an SP tree $\hat{T}$ destined to $t$;
                    If $\tau_{\hat{T}}^t$ has a promising reduced cost then begin
                        Add $\hat{T}$ to the current set of SP trees, i.e., $\tilde{\Omega} = \tilde{\Omega} \cup \hat{T}$;
                        Update $RLP_{curr}$;
                    end
                end
        until no new $\hat{T}$ can be found
        If current LP is feasible then begin
            Let $z_{ub}^*(n_b)$ be the upper bound obtained by approximation
            If $z_{ub}^*(n_b) < UB$ then begin
                $UB = z_{ub}^*(n_b)$
            end
            If the current optimal solution is not integral then begin
                Branch:
                    Select a fractional $\bar{\tau}_T^t$ variable and branch;
                    Create two child nodes $\{n_r, n_l\}$ and let $S = S \cup \{n_r, n_l\}$
            end
        end
        Extract B&B nodes that are fathomed by bound or infeasibility from $S$
    end
end
```

Figure 2: Summary of the B&P Algorithm for the Tree Formulation

$$\min r \tag{88}$$

$$\text{s.t.} \quad \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q \tag{89}$$

$$\chi_{hk} + \sum_{z=1}^{H} a_z \lambda_z^{hk} \leq 0 \qquad \forall \{h,k\} \in E \tag{90}$$

$$\Pi_{i,hk}^{st} - \Pi_{j,hk}^{st} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A,\ (s,t) \in Q,\ \{h,k\} \in E \tag{91}$$

$$-\pi_{hk}^{st} + \sum_{z=1}^{H} a_z^{st} \lambda_z^{hk} \geq f_{hk}^{st} + f_{kh}^{st} \qquad \forall (s,t) \in Q,\ \{h,k\} \in E \tag{92}$$

$$-\sum_{\{i,j\}\in E} c_{ij}\eta_{ij,hk} + \chi_{hk} \geq -rc_{hk} \qquad \forall \{h,k\} \in E \tag{93}$$

$$\sum_{T\in\Omega_t^0} \tau_T^t = 1 \qquad \forall t \in V \tag{94}$$

$$f_{ij}^{st} \leq \sum_{T\in\Omega_t^0\cap\Omega_{ij}^0} \tau_T^t \qquad \forall (i,j) \in A, (s,t) \in Q \tag{95}$$

$$\sum_{T\in\Omega_t^0\cap\Omega_{ij}^0} \tau_T^t + \rho_j^t - \rho_i^t + \theta_{ij} \geq 1 \qquad \forall t \in V, (i,j) \in A \tag{96}$$

$$-\sum_{T\in\Omega_t^0\cap\Omega_{ij}^0} \tau_T^t - \frac{\rho_j^t - \rho_i^t + \theta_{ij}}{2\Theta_{max}} \geq -1 \qquad \forall t \in V, (i,j) \in A \tag{97}$$

$$f_{ij}^{st} \leq \varphi_i^{st} \qquad \forall (i,j) \in A, (s,t) \in Q \tag{98}$$

$$1 + f_{ij}^{st} - \varphi_i^{st} \geq \sum_{T\in\Omega_t^0\cap\Omega_{ij}^0} \tau_T^t \qquad \forall (i,j) \in A, (s,t) \in Q \tag{99}$$

$$1 \leq \theta_{ij} \leq \Theta_{max} \qquad \text{integer } \forall (i,j) \in A \tag{100}$$

$$\tau_T^t \in \{0,1\} \qquad \forall t \in V, T \in \Omega_t^0 \tag{101}$$

$$0 \leq \varphi_i^{st} \leq 1 \qquad \forall i \in V, (s,t) \in Q \tag{102}$$

where $\Omega_t^0 = \Omega_0 \cap \Omega_t$ and $\Omega_{ij}^0 = \Omega_0 \cap \Omega_{ij}$ with the inital set of *SP trees* Ω_0. Given constraints (94), we must have at least one *SP tree* for every node $t \in V$ in Ω_0 to ensure that each t is reachable from every other node of $G = (V, E)$. Therefore, we need to construct Ω_0 using some metric. This metric can be such that each arc is assigned a unit weight or a value proportional to the physical distance between its two endpoints. We prefer to 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). This choice implies that Ω_0 will be formed by considering the link capacities to some extent. Finally, notice that $|\Omega_0| = |V|$ and we have one τ_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 node of the B&B tree, the linear programming problem is solved by generating the necessary τ variables dynamically. Given a solution of RLP_{curr} which uses a subset of τ variables, a *pricing* procedure is used to find a set of new τ variables whose reduced cost is negative and which may therefore improve the current routing. In other words, we look for some more promising routing strategies. Notice that the reduced cost of each τ_T^t variable (red_T^t)

is

$$-\zeta_t - \sum_{(i,j)\in T} \left[\upsilon_{ij}^t - \varsigma_{ij}^t + \sum_{s\in V\setminus\{t\}} (\nu_{ij}^{st} - \kappa_{ij}^{st}) + \sum_{c\in cut(n_b)} B_c \o_{ij}^c \right], \tag{103}$$

where ζ_t, ν_{ij}^{st}, υ_{ij}^t, ς_{ij}^t, and κ_{ij}^{st} are the dual variables of the constraints (79),(80), (81), (82), and (84), respectively. Moreover, we take care of the dual variables for the branching rules by including $B_c \o_{ij}^c$ in (103). In brief, suppose we are in the B&B node n_b with $cut(n_b)$ being the set of cutting planes added for all ancestor nodes of n_b. Then consider the *SP tree* T. If the arc (i,j) is contained in the *SP tree* T, then we would have $\o_{ij}^c = 1\ \forall c \in cut(n_b)$ provided that T appears in the cutting plane c. As a result, the dual variables B_c of the corresponding branching rules will be included in the reduced cost of τ_T^t.

As we have expressed in Section 5.1, we start the B&P algorithm with an initial set Ω_0 of *SP trees*. Then, as we generate new τ variables, we include the corresponding *SP trees* in our model, and update the set of currently available *SP trees* ($\tilde{\Omega}$) accordingly. While red_T^t is nonnegative for all *SP trees*, which are enumerated so far, i.e., $\forall T \in \tilde{\Omega}$, if we can find a new $\tau_{\hat{T}}^t$ with a negative reduced cost, then we can improve the current solution by simply routing all demands destined to t on $\hat{T}$.

To determine such *SP trees* we solve a shortest path problem for each destination node $t \in V$ with arc metric α on an auxiliary graph $G_{aux}(t,\alpha)$. 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, we can guarantee to have neither a nonnegative α nor an acyclic $G_{aux}(t,\alpha)$. Actually, it is very likely that $G_{aux}(t,\alpha)$ has negative cycles. Hence we cannot use the well known shortest path algorithms like Djikstra or Bellman-Ford algorithms to solve the pricing problem. Therefore, for each destination node t, we solve the pricing problem to determine promising *SP trees* using the following MIP model PR_t

$$z_t^* = \min \sum_{(i,j)\in A} \alpha_{ij} y_{ij} \tag{104}$$

$$\text{s.t.} \quad \sum_{j:\{i,j\}\in E} (f_{ij}^s - f_{ji}^s) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V, s \in V\setminus\{t\} \tag{105}$$

$$f_{ij}^s \le \varphi_i^s \qquad \forall (i,j) \in A, s \in V\setminus\{t\} \tag{106}$$

$$1 + f_{ij}^s - \varphi_i^s \ge y_{ij} \qquad \forall (i,j) \in A, s \in V\setminus\{t\} \tag{107}$$

$$-y_{ij} - \left(\frac{\rho_j - \rho_i + \theta_{ij}}{2\Theta_{max}}\right) \ge -1 \qquad \forall (i,j) \in A \tag{108}$$

$$y_{ij} + \rho_j - \rho_i + \theta_{ij} \ge 1 \qquad \forall (i,j) \in A \tag{109}$$

$$0 \le f_{ij}^s \le 1 \qquad \forall (i,j) \in A, s \in V \tag{110}$$

$$0 \le \varphi_i^s \le 1 \qquad \forall i \in V, s \in V \tag{111}$$

$$1 \le \theta_{ij} \le \Theta_{max} \qquad \text{integer } \forall (i,j) \in A \tag{112}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{113}$$

$$\rho_i \ge 0 \qquad \forall i \in V \tag{114}$$

where the binary variable y_{ij} indicates if (i,j) is an *SP arc* for t whereas f, φ, ρ, and θ retain their definitions made in the original master problem. Moreover, we set

$$\alpha_{ij} = -\bar{\upsilon}_{ij}^t + \bar{\varsigma}_{ij}^t - \sum_{s\in V\setminus\{t\}} (\bar{\nu}_{ij}^{st} - \bar{\kappa}_{ij}^{st}) - \sum_{c\in cut(n_b)} B_c \o_{ij}^{n_b} \quad \forall (i,j) \in A$$

in the objective function. Consequently, since PR_t contains the OSPF and the ECMP constraints as we have discussed before, its solution is an *SP tree* $\hat{T}$ defined with respect to some metric θ and its total length is $z_t^* = \sum_{(i,j)\in T^*} \alpha_{ij}$. Now, if $z_t^* < \bar{\zeta}_t$ then we have a new routing configuration whose inclusion could improve the current solution of the original problem. Hence we add the *SP tree* $\hat{T} = \{(i,j) \in A : y_{ij}^* = 1\}$ destined at t to $\tilde{\Omega}$. Note that we solve the pricing problem for all nodes $t \in V$ at each call of the *Price* routine in Figure 2.

5.3 Upper bound approximation

At each node n_b of the B&B tree, we keep on pricing τ variables and reoptimizing the updated RLP_{curr} problem until we cannot identify new *SP trees*. When we are done at n_b we have a lower bound $LB(n_b)$ on the optimal oblivious ratio $r(n_b)$ we could achieve under the same set of constraints defining n_b. On the other hand, if we can find a feasible solution of the original master problem, then this will be an upper bound (UB) on the optimal oblivious ratio r^*. Such an information would be useful especially for those large instances that are difficult to solve to optimality in reasonable time. As a result, we implement a simple method where we fix an *SP tree* for each destination node $t \in V$ and solve the original problem with this specific routing plan. In brief, given the optimal solution of RLP_{curr} we have the optimal values for the $\bar{\tau}_T^t$ variables. So, for each $t \in V$ we pick the *SP tree* T^* destined at t such that $\bar{\tau}_{T^*}^t \geq \bar{\tau}_T^t \;\; \forall T \in \tilde{\Omega}_t$, where $\tilde{\Omega}_t$ is the set of currently known *SP trees* destined at t. Then we fix these $\bar{\tau}_{T^*}^t$ variables to 1 and solve the original master problem. If this routing strategy is viable, then we have an oblivious ratio $z_{ub}(n_b)$, which is an upper bound UB on the optimal oblivious ratio r^*. An overview of this method is provided in Figure 3.

5.4 Branching

The efficiency of the B&P algorithm is highly dependent on the effectiveness of the branching rule. Moreover, the structure of the pricing problem should not be destroyed for the B&P method to be applicable. Hence, 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 Figure 2, we use fractional $\bar{\tau}_T^t$ variables to determine the restrictions we impose in each branching step. This does not mean that we 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 have used 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. However, it is possible that PR_t finds an *SP Tree* $\tilde{T}$ with exactly the same set of arcs of T, i.e., $\tilde{T} \equiv T$. Consequently, we have decided to 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 Figure 4.

When we are done with 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:

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

$$f_{i^*j^*}^{s^*t^*} = 0 \tag{115}$$

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

$$f_{i^*j^*}^{s^*t^*} \geq \frac{\sum_{(k,i^*)\in A} f_{ki^*}^{st}}{deg(i^*) - 1} \tag{116}$$

Notice that the summation on the right hand side of the inequality (116) is the total inflow for node i^*. Moreover, suppose $deg(i^*)$ arcs are incident to i^*. Then in order (i^*, j^*) to be an *SP arc*, we must have at least one incoming arc and at most $deg(i^*) - 1$ outgoing arcs for node

Procedure Upper Bound Approximation
input: Optimal values of the τ variables for RLP_{curr}, i.e.,$\bar{\tau}$
output: Upper bound UB on the optimal oblivious ratio r

begin

For each $t \in V$ **begin**

Pick the largest $\bar{\tau}^t_{T^*}$ variable such that $\tau^{\bar{t}}_{T^*} \geq \tau^t_T \ \ \forall T \in \tilde{\Omega}_t$

Let $\bar{A}_t$ be the set of *SP arcs* contained in T^*

Get the fraction of demand routed on each arc $(i,j) \in \bar{A}_t$ by solving LP_t:

$$\min 0$$

$$\text{s.t.} \ \sum_{(i,j)\in \bar{A}_t} f^s_{ij} - \sum_{(j,i)\in \bar{A}_t} f^s_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V, s \in V \backslash \{t\}$$

$$-f^s_{ij} + \varphi^s_i = 0 \qquad \forall (i,j) \in \bar{A}_t, s \in V \backslash \{t\}$$

$$0 \leq f^s_{ij} \leq 1 \qquad \forall (i,j) \in \bar{A}_t, s \in V \backslash \{t\}$$

$$f^s_{ij} = 0 \qquad \forall (i,j) \in A \backslash \bar{A}_t, s \in V \backslash \{t\}$$

$$0 \leq \varphi^s_i \leq 1 \qquad \forall i \in V, s \in V \backslash \{t\}$$

Let $\bar{\sigma}^{st}_{ij} = \bar{f}^s_{ij} + \bar{f}^s_{ji} \quad \forall \{i,j\} \in E, (s,t) \in Q$

end

Solve the following problem P_{UB} to get an upper bound UB on r^*

$$z^*_{ub}(n_b) = \min r$$

$$\text{s.t.} \ \chi_{hk} + \sum_{z=1}^{H} a_z \lambda^{hk}_z \leq 0 \qquad \forall \{h,k\} \in E$$

$$\Pi^{st}_{i,hk} - \Pi^{st}_{j,hk} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A; (s,t) \in Q; \{h,k\} \in E$$

$$-\pi^{st}_{hk} + \sum_{z=1}^{H} a^{st}_z \lambda^{hk}_z \geq \bar{\sigma}^{st}_{ij} \qquad \forall (s,t) \in Q; \{h,k\} \in E$$

$$\rho^t_j - \rho^t_i + \theta_{ij} = 0 \qquad \forall (i,j) \in \bar{A}_t, t \in V$$

$$\rho^t_j - \rho^t_i + \theta_{ij} \geq 0 \qquad \forall (i,j) \in A \backslash \bar{A}_t, t \in V$$

$$\chi_{hk} \geq 0 \qquad \forall \{h,k\} \in E$$

$$\lambda^{hk}_z \geq 0 \qquad \forall z = 1,..,H, \ \{h,k\} \in E$$

$$\eta_{ij,hk} \geq 0 \qquad \forall \{i,j\} \in E, \ \{h,k\} \in E$$

$$\rho^t_i \geq 0 \qquad \forall i \in V, \ t \in V$$

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

$$r \geq 1$$

If $z^*_{ub}(n_b) < UB$ **then begin**

Let $UB = z^*_{ub}(n_b)$

end

end

Figure 3: Summary of the Upper Bound Approximation Method

```
procedure: Select Branch
input: τ̄_T^t values in the solution of RLP_curr
output: The quadruple (i*, j*, s*, t*) // (s*, t*) ∈ Q; (i*, j*) ∈ A

begin
    Take the most fractional τ̄_{T*}^t; let t* = t and T_1 = T*
    Find the second most fractional τ̄_{T_*}^{t*}; let T_2 = T_*
    found = FALSE; //no quadruple is chosen
    For each arc (i, j) ∈ A begin
        if ((i, j) ∈ T* ∪ T_* and (i, j) ∉ T* ∩ T_*) then begin
          if (f̄_ij^st > 0 and (i, j, s, t) is not used in upper branches) then begin
            if deg(i) > 1 then begin
              Let (i*, j*, s*, t*) = (i, j, s, t);
              found = TRUE;
            end
          end
        end
    end
    If (found = FALSE) then begin
      For each arc (i, j) ∈ A begin
          For each pair (s, t) ∈ Q begin
              If (f_ij^st > 0 and (i, j, s, t) is not used in upper branches) then begin
                Let (i*, j*, s*, t*) = (i, j, s, t);
                found = TRUE;
              end
          end
      end
    end
    If found = FALSE then begin
      STOP //no further branching at the current B&B node
    end
end
```

Figure 4: Steps of Branch Selection

i^*. 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. This is why we have this constant in the denominator of (116).

Given the current B&B node n_b and its associated relaxation RLP_{curr}, we create the two new nodes n_r and n_l by adding the constraints (115) and (116) to the current restricted problem as well as the corresponding pricing problems PR_{t^*}. Additionally, we also impose the constraints

- Do not use *SP trees* containing arc (i^*, j^*), i.e.,

$$\sum_{T \in \tilde{\Omega}_{t^*} \cap \tilde{\Omega}_{i^* j^*}} \tau_T^{t^*} = 0 \tag{117}$$

- Do not use *SP trees* not containing arc (i^*, j^*), i.e.,

$$\sum_{T \in \tilde{\Omega} \setminus (\tilde{\Omega}_{t^*} \cap \tilde{\Omega}_{i^* j^*})} \tau_T^{t^*} = 0 \tag{118}$$

to create n_r and n_l, respectively. For both branches we just need to modify the upper bounds of the corresponding τ variables. Similarly, for n_r we need to modify the upper bound of the flow variable whereas for n_l we add a new constraint. Alternatively, together with the cutting plane in (118), the result of the following proposition can also be used to define n_r.

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*

$$f_{ij}^{st} \geq \frac{1}{deg(s) * \prod_{l \in V \setminus \{s,t\} : deg(l) > 1} (deg(l) - 1)} \tag{119}$$

Proof. Suppose that arc (i,j) is an *SP arc* for the demand pair (s,t). In the worst case the demand d_{st} originated at the source node s would visit all nodes in the graph $G = (V, E)$ before it ceases at the destination node t and all arcs of G would be *SP arcs*. Then, for the source node s we would have

$$f_{sj}^{st} = \frac{1}{deg(s)} \quad \forall (s,j) \in A$$

whereas

$$f_{hk}^{st} = \frac{\sum_{(l,h) \in A} f_{lh}^{st}}{(deg(h) - 1)} \quad \forall (h,k) \in A, \;\; h \in V \setminus \{s,t\}$$

for the rest of the graph. For example, suppose that $SP_{st} = \{(s,i), (i,j), (j,m), ..., (k,t)\}$ is a shortest path from s to t and all arcs incident to all nodes on this path are *SP arcs*. Then we have

$$\begin{aligned}
f_{si}^{st} &\geq \frac{1}{deg(s)} \\
f_{ij}^{st} &\geq \frac{1}{deg(s) * (deg(i) - 1)} \\
f_{jm}^{st} &\geq \frac{1}{deg(s) * (deg(i) - 1) * (deg(j) - 1)} \\
&\vdots \\
f_{kt}^{st} &\geq \frac{1}{deg(s) * \prod_{l \in PATH_{sk}} (deg(l) - 1)}
\end{aligned}$$

and hence

$$f_{kt}^{st} \geq \frac{1}{deg(s) * \prod_{l \in V \setminus \{s,t\}} (deg(l) - 1)}$$

where $PATH_{sk}$ is the set of nodes on the shortest path from s to k. The latter inequality is based on the assumption that in the worst case d_{st} would visit all nodes before it reaches its destination node. □

In our computational experiments, we have used (119) rather than (116). This is mainly because our models are already difficult to solve and we do not want the increase the size of current problem as we go down the B&B tree. Moreover, unlike (116), the inequalities (119) ensure that the flow on an *SP arc* (i, j) is positive. We have observed that this difference has improved the performance of the B&P algorithm for the set of instances we have worked on. On the other hand, using (119) and (116) 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. [20] within the context of Virtual Private Network (VPN) design. In this approach the focus is on the outflow and inflow capacities of some special nodes, which are called VPN terminals, rather than the individual demands. Namely, the set of feasible demands is defined by some bounds on the total flow each terminal node can send to and accept from the rest of the VPN terminals. Then the set of feasible demand matrices with the Hose model is

$$D = \{d \in \mathbb{R}^{|Q|} : \sum_{t \in W \setminus \{s\}} d_{st} \leq b_s^+; \quad \sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s^-; \quad d_{st} \geq 0 \quad \forall (s,t) \in Q\} \qquad (120)$$

where b_s^- and b_s^+ are the ingress and egress capacities of the terminal node $s \in W$, respectively. Notice that this is more known as the *asymmetric* version of the Hose model, and that 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 we have box constraints to define the lower and upper bounds on the pairwise flow demands. Since our models provide worst case guarantees we would get a very conservative solution, which assumes that all demands can get their peak levels simultaneously. To overcome this problem we can use a positive integer parameter Γ to scale the trade off between the robustness of the model and the conservatism level of the solution. This is the approach discussed within the context of robust optimization by Bertsimas and Sim in [9] and [10]. In our problem, Γ 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 demands d_{st} range between d'_{st} and $d'_{st} + \hat{d}_{st}$ (where $\hat{d}_{st} > 0$) and that not more than Γ 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 β_{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 β the BS uncertainty model defines the polyhedral set of feasible demands as follows:

$$D = \{d \in \mathbb{R}^{|Q|} : d'_{st} \leq d_{st} \leq d'_{st} + \hat{d}_{st} \quad \forall (s,t) \in Q; \;\; \sum_{(s,t)\in Q} \frac{d_{st}-d'_{st}}{\hat{d}_{st}} \leq \Gamma\}. \tag{121}$$

6.3 Numerical results

We have performed numerical experiments on instances of various sizes to assess the performance of our formulations and the B&P algorithm. We have also included MPLS routing in our estimations to compare it to the OSPF routing with ECMP condition under weight management. Note that the MPLS oblivious performance ratio under general demand uncertainty is found by solving the linear program (43)-(52), where we restrict neither the routing pattern nor how each demand d_{st} is shared among multiple paths between s and t. Therefore, MPLS routing does not perform worse than OSPF routing with ECMP. Nevertheless, it would be a good benchmark for us to comment on the oblivious ratios under OSPF environment since we have $z_{mpls} \leq z_{ospf}$ where z_{mpls} and z_{ospf} are the oblivious performance ratios for MPLS routing and OSPF routing with ECMP, respectively. Furthermore, Fortz and Thorup [6] compare the performance of optimal OSPF routing with the optimal MPLS routing for a *fixed* TM and state that their performances almost match in this case. However, we deal with oblivious routing where there is a *set* of feasible demands. To the best of our knowledge, there is no other reference comparing oblivious MPLS routing with oblivious OSPF routing with weight management for such a general definition of feasible traffic matrices. Therefore, we believe it is important to extend this comparison to the case of a *set* of feasible demands rather than a single TM as Applegate and Cohen [8] also mention.

The instances *bhvac, pacbell, eon, metro*, and *arpanet* are well known instances studied in the IEEE literature. On the other hand, *Exodus (Europe), Abovenet (US), VNSL (India)*, and *Telstra (Australia)* are from the Rocketfuel project [21] for which we have the data for the topology ($|V|$ and $|E|$), the link weights (w), and the number of data packets entering and leaving each node. For these instances we have assumed that the weight metric w obey the inverse capacity weight setting where the weight of each link is inversely proportional to its capacity, i.e., $c_{ij} = 1/w_{ij} \;\; \forall \{i,j\} \in E$. Moreover, since the information on real demand matrices is not made publicly available, we have used the Gravity model mentioned by Applegate and Cohen [8] to generate the demand polyhedra D matching each instance. This approach is based on the assumption that a demand d_{st} is proportional to the product of a *repulsion* term R_s associated with the source, and an *attraction* term A_t associated with the destination, which, for instance, can be set as the total observed outgoing and incoming traffic, respectively. A *base demand* $\bar{d}$ is defined and the uncertainty polyhedron is constructed around $\bar{d}$: we have the data on the number of data packets incoming and outgoing for each node i, i.e., the repulsion (R_i) and attraction (A_i) parameters. Then the base demand for pair (s,t) is estimated using the relation $\bar{d}_{st} = \beta R_s A_t$, where β is computed in order for $\bar{d}$ to be feasible (i.e., to admit at least one routing) and to choose how close $\bar{d}$ is to the boundary of the feasibility region. Let us define $\varsigma \in [0,1]$ such that $\beta = \varsigma \upsilon^*$ with

$$\upsilon^* = \max \upsilon \tag{122}$$

$$\text{s.t.} \quad \sum_{j:\{s,j\}\in E} (g^{st}_{sj} - g^{st}_{js}) = \upsilon R_s A_t \quad \forall (s,t) \in Q \tag{123}$$

$$\sum_{j:\{i,j\}\in E} (g^{st}_{ij} - g^{st}_{ji}) = 0 \quad \forall i \in V \setminus \{s,t\}, (s,t) \in Q \tag{124}$$

$$\sum_{(s,t)\in Q} (g^{st}_{ij} + g^{st}_{ji}) \leq c_{ij} \quad \forall \{i,j\} \in E \tag{125}$$

$$g^{st}_{ij} \geq 0 \quad \forall (i,j) \in A,\ (s,t) \in Q. \tag{126}$$

We fix a *direction* (the half-line $\bar{d}_s t = \beta R_s A_t$) on which $\bar{d}$ must lie, and solve the LP above to find the most critical demand value, which is on the boundary of the feasibility region. Then, ς scales this value so that $\bar{d}$ is an inner 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 ς.

For the Hose and BS uncertainty models, we have determined the set of terminal nodes W among the busiest nodes, i.e., the ones with large R_i and A_i parameters. It should be mentioned that our instances are *dense* instances in the sense that in all but two cases we have $|Q|/|V| \geq 0.33$. Moreover, we have created 4 variants of each instance using different uncertainty parameters p with values $\{1.1,\ 2,\ 5,\ 20\}$ for the BS model. We will refer to each BS instance using the label *(name,p)*, i.e., *(nsf,2)* is the nsf instance with uncertainty level $p = 2$. Larger p values imply higher variation in demand estimates. Hence the optimal oblivious ratio is also expected to be larger for such cases. On the other hand, we have randomly picked a subset S of W such that $|S| = |W|/2$. Then we have used $b^+_s = (\sum_{(s,t)\in Q} \bar{d}_{st})/1.1\ \forall s \in S$, $b^+_s = 1.1(\sum_{(s,t)\in Q} \bar{d}_{st})\ \forall s \in W \backslash S$, $b^-_s = 1.1(\sum_{(s,t)\in Q} \bar{d}_{st})\ \forall s \in S$, and $b^-_s = (\sum_{(s,t)\in Q} \bar{d}_{st})/1.1\ \forall s \in W \backslash S$ as the outflow and inflow capacities of the terminal nodes in the Hose model. It is worth noting that the uncertainty set is asymmetric in this case. This feature is believed to complicate the problem based on the VPN design literature (Altın et al. [1]).

We have used AMPL to model the flow formulation as well as the MPLS routing and Cplex 9.1 MIP solver to solve them. The B&P algorithm is implemented in C using MINTO (Mixed INTeger Optimizer) [19] and Cplex 9.1 as LP solver. We have set a two hours time limit both for AMPL and MINTO. Our test results for two uncertainty models discussed above are summarized in Table 1 and Table 2 with:

- the instance characteristics, i.e., the name of the instance as well as the numbers of nodes, arcs, and terminals,
- the measure of the demand uncertainty p that we use in the creation of the test instances for the BS model. After getting an estimate of the average traffic demand ($\bar{d}_{st}$) for a pair (s,t), we set the corresponding $d'_{st} = \bar{d}_{st}/p$ and $\hat{d}_{st} = (p - \frac{1}{p})\bar{d}_{st}$.
- the solution z_{tree} and total CPU time t_{tree} of the B&P algorithm;
- the solution z_{flow} and CPU time t_{flow} of the flow formulation;
- 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 [8] call this more complicated routing effort as *best OSPF style* routing and mention that it is highly non-trivial. Therefore, it is not surprising that 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 the optimal solution of the corresponding problem we put a * next to this upper bound. On the other hand, if no feasible solution is available, then the best lower bound obtained by solving the associated LP relaxation is given in brackets. Furthermore, *NoI* means that we do not even have a feasible solution for the LP relaxation, i.e., the Phase I problem could not be solved in 2 hours. Finally, MINTO could not solve some instances due to excessive memory requirements. We label such cases with *MA* under the t_{tree} column.

Note that the z_{tree}, z_{flow}, and z_{mpls} columns provide a relative performance measure 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 can be at least 1 where larger numbers imply larger deviation from the best possible routing tailored for that instance. Moreover, a value of 1 means that the perfectly oblivious routing is found by solving the corresponding model. In other words, by using our optimization tools we find a routing, which is the best tailored for any traffic matrix in the feasible set D.

Table 1 shows the results for the BS uncertainty model for 11 instances of 4 different levels of uncertainty. As expected, the 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. Moreover, in those cases where the tree formulation provides a worse upper bound than the flow formulation, B&P algorithm run less than 2 hours and had to stop due to memory allocation problems. Finally, 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. Hence, we can say that it is worth implementing a specialized B&P algorithm for the BS uncertainty model. However, this problem has considerable memory requirements. Therefore, it is not likely to get very promising results for large instances within reasonable time limits neither with the tree nor with the flow formulation. As a result, the performance of optimal oblivious OSPF routing with weight management is not expected to be comparable with the performance of optimal oblivious MPLS routing for large cases.

A comparison of the OSPF and MPLS routings based on our test results should be made in two stages. In the first step, we focus on the 24 instances for which we could find the optimal solutions 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. In other words for any network the deviation for OSPF at uncertainty level p is almost never less than the one for a smaller p. 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, for the *nsf* network is provided in Figure 5.

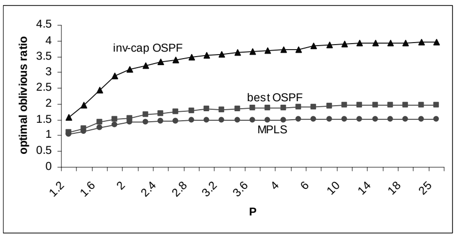

Figure 5: The change in the optimal solutions of the *best OSPF style*, MPLS, and inverse capacity weight routings for the network *nsf* for different values of p.

Firstly, 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 5, weight management resulted in an improvement in the OSPF performance. A more concrete comparison of the three alternative routing alternatives is given in Figure 6, which shows the gaps between the optimal performance ratios. We can say that inverse capacity OSPF routing is almost 100% worse than *best OSPF* in all higher uncertainty levels for the *nsf* network. On the other hand 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. However, we can say that weight management has also helped to reduce the impact of demand uncertainty on oblivious ratio to some extent.

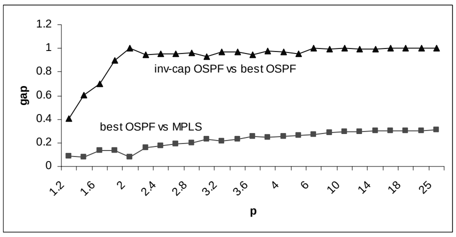

Figure 6: Comparison of the *best OSPF style* routing with MPLS and OSPF under inverse capacity weight setting for the instance *nsf* for different values of p.

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[2] instances.

The second traffic uncertainty model we focus on is the Hose model for which the test results are shown in Table 2. The most obvious comment we can make is that the management of the Hose uncertainty model is more difficult than the BS model for both the OSPF and MPLS routings. We can make such a comment based on the computation times. Moreover, for the instances *eon* and *arpanet* we could not get even a feasible solution with neither the flow nor the MPLS formulations. Hence, we believe it will be fair to focus on the other instances of the Hose model while interpreting the numerical results.

The performances of the tree and flow formulations in terms of computation times are comparable for relatively smaller instances like *Exodus* and *VNSL* where the optimal oblivious ratios are found. Nonetheless, the B&P algorithm had to stop due to excessive memory requirements for *nsf*, *example*, and *Telstra* 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. On the other hand, the tree formulation is superior with respect to the lower bounds found at the end of 2 hours.

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, the 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. Note that 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

[2]We consider large and dense topologies as difficult instances. t_{mpls} values are also indicators of the difficulty level.

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	0.048	1	0.052
				2	1	0.05	1	0.044	1	0.048
				5	1	0.05	1	0.044	1	0.036
				20	1	0.04	1	0.036	1	0.036
nsf	8	20	5	1.1	1.381*	*MA*	1.05*	2 hrs	1.013	0.368
				2	2.299*	*MA*	1.556	3821.53	1.44	0.752
				5	3.808*	*MA*	1.904	94.33	1.423	0.984
				20	3.936*	*MA*	1.976	241.1	1.462	1.054
VNSL	9	22	3	1.1	1.066	39.75	1.066	0.19	1	0.016
				2	1.066	3.61	1.066	0.14	1	0.024
				5	1.066	24.77	1.066	0.22	1	0.02
				20	1.066	9.24	1.066	0.296	1	0.02
example	10	30	4	1.1	1	0.11	(1)	2 hrs	1	0.275
				2	1	0.15	1	1900.19	1	0.406
				5	2.25*	*MA*	1.82*	2 hrs	1.034	0.547
				20	2.575*	*MA*	3.269*	2 hrs	1.079	0.775
metro	11	84	5	1.1	4.357*	2 hrs	(1)	2 hrs	1	92.969
				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	81.177
				2	1	120.03	(1.0004)	2 hrs	1	23
				5	1	41.32	(1)	2 hrs	1	44.234
				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	12.482
				2	1	13.58	2.24284*	2 hrs	1	35.95
				5	1	13.92	2.68684*	2 hrs	1	54.06
				20	1	16.31	5.3568*	2 hrs	1	46.35
Telstra	44	88	7	1.1	1	1.75	1	0.504	1	0.156
				2	1	1.79	1	0.414	1	0.158
				5	2.075*	MA	1.054	2.56	1	0.159
				20	2.081*	MA	1.886	2.39	1.283	0.181
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)	4403 sec	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	(1.3133)	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 1: Results for the BS uncertainty model

say that the definition of the traffic polyhedra D is looser in the former[3]. Therefore, we believe that these average deviations between the two protocols support our remark that degredation 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	0.052	1	0.031
nsf	8	20	5	4*	*MA*	2	2730.38	1.517	0.403
VNSL	9	22	3	1.0655	8.77	1.0655	0.296	1	0.16
example	10	30	4	2.7*	MA	2	2 hrs	1.079	0.424
metro	11	84	5	(1.437)	2 hrs	(1.302)	2 hrs	1.302	1657.832
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	326.125
Telstra	44	88	7	2.081*	MA	1.925	1.224	1.283	0.084
pacbell	15	42	7	(1.544)	2 hrs	(1.543)	2 hrs	1.543	59.131
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 2: Results for the hose uncertainty model

Our final comment is about the benefit of considering a polyhedra of demands rather than a 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, note that 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^*. 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^*. On the other hand, for the nsf instances we have observed that $f_{\bar{d}} \equiv f^*$. As a result, we believe that optimizing just for the mean demands does not suffice to 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. This is the main motivation of our work and we consider the case where the polyhedra of feasible demands is defined by some system specific constraints. We incorporate this general uncertainty into the OSPF style routing problem. To comply with the current forwarding technology, we also include the equal load sharing condition (ECMP) in our analysis. Furthermore, we employ weight management to improve the network performance of OSPF. Given all these specifics of the problem we focus on the minimization of the maximum link congestion via a fair allocation of traffic among the network links. To our knowledge, our paper is the first work on such a general and practically defensible *best OSPF style* routing.

We have proposed two mixed integer models obtained by a duality-based reformulation for our problem. The first one 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

[3] 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 $\bar{d}_{st}$.

tree formulation based on special structured subgraphs of the backbone graph, i.e., *SP trees*. Moreover, we have proposed a B&P algorithm supported by cutting planes to solve this model.

We have tested our models and the B&P algorithm on two traffic uncertainty definitions, namely the Hose model and the BS model. We have presented a comparison of the two formulations in terms of the solution quality and computation times. We have observed that it pays to create a specialized B&P algorithm especially for the BS uncertainty case. Unfortunately, due to excessive memory requirements of the algorithm, it had to stop before two hours time limit in some instances. Additionally, we have compared the OSPF style routing and the MPLS style routing for these two traffic polyhedra. First, we have realized that for the BS case the optimal oblivious ratios for both routing styles increase as the level of demand variability increases. Another important observation is that the performance of OSPF routing degrades more than the MPLS routing as the demand uncertainty increases. To sum up, we believe that a polyhedral definition of the feasible set of demand matrices, which is accurate as far as possible, could make the OSPF performance get closer to the MPLS performance.

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] A. Bley and T. Koch, Integer programming approaches to access and backbone IP-network planning. Tech. Report TR 02-41, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2002.

[3] A. Parmar, S. Ahmed, and J. Sokol, An integer programming approach to the OSPF weight setting problem. *submitted for publication*, 2006.

[4] 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, 2003.

[5] A. Tomaszewski, M. Pióro, M. Dzida, and M. Zagożdżon, Optimization of administrative weights in IP networks using the branch-and-cut approach. Proceedings of *INOC 2005*, B2, pp. 393-400.

[6] B. Fortz and M. Thorup, Internet traffic engineering by optimizing OSPF weights. Proceedings of *IEEE INFOCOM 2000*, pp. 519-528.

[7] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance, Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46, 1998, pp. 316-329.

[8] D. Applegate and E. Cohen, Making intra-domain routing robust to changing and uncertain traffic demands:Understanding fundamental tradeoffs. Proceedings of *SIGCOMM '03*, Karlsruhe, Germany, pp. 313-324.

[9] D. Bertsimas and M. Sim, Robust discrete optimization and network flows. *Mathematical Programming*, Ser. B 98, 2003, pp. 43-71.

[10] D. Bertsimas and M. Sim, The price of robustness. *Operations Research*, 52, 2004, pp. 35-53.

[11] F. Y. Lin and J. L. Wang, Minimax open shortest path first routing algorithms in networks supporting SMDS service. *Proc. IEEE Int. Conf. Communications(ICC)*, 2, 1993, pp. 666-670.

[12] G. Oriolo, Domination in traffic matrices. Available for download at http://www.optimization-online.org/DB_FILE/2004/09/952.pdf

[13] J. Zhang, Z. Ma, C. Yang, A column generation method for inverse shortest path problems. Mathematical Methods of Operations Research (ZOR) 41 (3), 1995, pp. 347-358.

[14] B. Burton, W. R. Pulleyblank, and Ph. L. Toint. The inverse shortest path problem with upper bounds on shortest path costs, In: P. Pardalos, D. W. Hearn, and W. H. Hager, eds. Network Optimization, Springer Lecture Notes in Economics and Mathematical Systems, 450 (1997), 156–171.

[15] J. Wang, Y. Yang, L. Xiao, K. Nahrstedt, Edge-based traffic engineering for OSPF networks. *Computer Networks*, in press.

[16] K. Holmberg and D. Yuan, Optimization of internet protocol network design and routing. *NETWORKS*, 43(1), 2004, pp. 39-53.

[17] L. De Giovanni, B. Fortz, and M. Labbé, A lower bound for the internet protocol network design problem. Proceedings of *INOC 2005*, B2, pp. 401-408.

[18] 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.

[19] M.W.P. Savelsbergh, G.C. Sigismondi, G.L. Nemhauser. A functional description of MINTO, a Mixed INTeger Optimizer. OR Letters 15 (1994), pp. 47-58.

[20] 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. *Proceedings of ACM SIGCOMM*, 1999, pp. 95-108.

[21] N. Springs, R. Mahajan, D. Wetherall, Measuring ISP topologies with Rocketfuel. Proceedings of *IEEE/ACM Transactions on Networking* 12-1 (2004), pp. 2-16.

[22] P. Belotti and M.Ç. Pınar, Optimal oblivious routing under linear and nonlinear uncertainty, 2005, submitted to *Optimization and Engineering*. Available for download from http://www.andrew.cmu.edu/user/belotti/papers/obl-stat.pdf

[23] P. Broström and K. Holmberg, Design of IP/OSPF networks using a Lagrangean heuristic on an in-graph based model. Proceedings of *INOC 2005*, B3, pp.702.

[24] Y. Wang, Z. Wang, and L. Zhang, Internet traffic engineering without full mesh overlaying. Proceedings of *IEEE INFOCOM 2001*, pp. 565-571.