Oblivious OSPF Routing with Weight Optimization under Polyhedral Demand Uncertainty

Ayşegül Altın†, Bernard Fortz†‡ and Hakan Ümit∗‡
†Département d’Informatique, Faculté des Sciences, Université Libre de Bruxelles
∗IAG - Louvain School of Management and †CORE, Université catholique de Louvain
{aaltin, bfortz}@ulb.ac.be, hakan.umit@uclouvain.be

October 23, 2008

Abstract

The desire for configuring well-managed OSPF routes to handle the communication needs in the contemporary business world with larger networks and changing service requirements has opened the way to use traffic engineering tools with the OSPF protocol. Moreover, anticipating possible shifts in expected traffic demands while using network resources efficiently has started to gain more attention. We take these two crucial issues into consideration and study a weight-managed OSPF routing problem for polyhedral demands. Our motivation is to optimize the link weight metric such that the minimum cost routing uses shortest paths with Equal Cost Multi-Path (ECMP) splitting, and the routing decisions are robust to possible fluctuations in demands. In addition to a compact mixed integer programming model, we provide an algorithmic approach with two variations to tackle the problem. We present several test results for these two strategies and discuss whether we could make our weight-managed OSPF comparable to unconstrained routing under polyhedral demand uncertainty.

Keywords: OSPF, polyhedral demand, hose model, oblivious routing, tabu search.

1 Introduction

Open Shortest Path First (OSPF) was developed as an Interior Gateway Protocol in mid-1980s by the Internet Engineering Task Force to provide a better service for large heterogeneous internetworks. It is designed to work within an hierarchical structure where the largest entity is called an Autonomous System (AS) or a domain, interchangeably. Each AS is a collection of networks and routers having a common routing strategy. OSPF is an intra-AS routing protocol where a Shortest Path First (SPF) algorithm is used to determine the routing paths between routers in the same AS. These paths are defined uniquely by link weights (also called link metric), which can be set in proportion to several measures such as physical distances or inversely proportional to link capacities. Nevertheless fixing link weights a priori gives no chance to traffic engineering with OSPF and leaves it behind in the competition with new routing protocols like MPLS. Consequently, Fortz and Thorup [20, 22] had launched the idea of managing link weights so as to make an efficient use of network resources and improve performance. Since then, determining the link metric and hence the routing paths so as to optimize some design criteria like link utilization or routing cost has been the focus of the most recent research on OSPF routing [2, 23, 31, 32, 34, 39, 41].

The effective and efficient use of network resources is important for providing a stable service at a reasonable cost. A balanced distribution of traffic among network links in proportion to their capacities would improve service uptime by making it less prone to changes in demand estimates[4]. The most common measure for the fairness of a routing is the congestion rate, i.e., the utilization rate of the most
loaded link defined as the proportion of link capacity usage. The concern is to make this rate as small as possible to avoid any link to become the bottleneck due to overloading.

For a given network, the traditional routing problem deals with selecting paths to transfer a ‘given’ set of demands from their origins to destinations. In this general definition, there is no restriction on the structure of the paths to be used, and it is assumed that the amount of traffic between all origin and destination pairs are already known. However, several restrictions are imposed on the path structure in telecommunication networks, and designing a reliable network using a single demand matrix strains credibility as the network size and the service variety increase in the contemporary business world. It is not likely to anticipate fluctuations in demand expectations without overestimations, which would lead to the waste of network resources or a high service cost. A well-known online approach to handle such shifts is to update routes adaptively as some changes are observed. However, the additional benefits of these methods are not for free since excessive modifications might ruin the consistency and dependability of network operations. At this point, off-line methods based on optimizing over a set of traffic matrices have started to win adherents [2] [3] [4] [7] [31].

The general method is to use either a discrete set and hence a scenario-based optimization or a polyhedral set defined by network characteristics. Then the motivation is to determine the routing whose worst case performance for any feasible realization in this set is the best. Such a routing is called oblivious since it is determined irrespective of a specific demand matrix. Applegate and Cohen [3] discuss the general routing problem with almost no information on traffic demands. Later Belotti and Pinar [4] incorporate box model of uncertainty as well as statistical uncertainty into the same problem. Ben-Ameur and Kerivin [7] study the minimum cost general oblivious routing problem under polyhedral demand uncertainty and use an algorithm based on iterative path and constraint generation as a solution tool. Mulyana and Killat [31] deal with the OSPF routing problem, where traffic uncertainty is described by a set of outbound constraints. Finally, Altın et al. [2] study polyhedral demand uncertainty with OSPF routing under weight management and provide a compact MIP formulation and a Branch-and-Price algorithm.

In this paper, we discuss oblivious OSPF routing with weight management and polyhedral demands. Optimizing weights would enable traffic engineering with OSPF since link metric is the only tool we can employ to manipulate routes so as to make OSPF more comparable to other flexible protocols like MPLS. Moreover, polyhedral demands make the problem more practically defensible by ensuring a design robust under a range of applicable shifts in traffic demand. In addition, the current OSPF technology is not compatible with the arbitrary split of flow among multiple paths and allows either single path routing or multi-path routing with equal load share. We implement the latter approach called Equal Cost Multi-Path (ECMP) rule, where the traffic leaving each node is split equally among the outgoing links belonging to a shortest path from that node to the destination. Although ECMP would make our problem more complicated, it will provide a larger space for improvement in the configuration process.

Except Altın et al. [2] and Mulyana and Killat [31], we are not aware of any other work on OSPF routing and demand uncertainty. Nevertheless, the deterministic problem has been studied in depth for several variants arising from the number of paths used. Interested reader can refer to Bley and Koch [9], Lin and Wang [30], and Tomaszewski et al. [38] for single path routing. We cite Broström and Holmberg [12], Fortz and Thorup [20], Giovanni et al. [15], Parmar et al. [32], Sridharan et al. [37], and Wang et al. [40] for multi-path routing of deterministic demands under ECMP, which is an NP-hard problem (Fortz and Thorup [20], Pióro et al. [33], Wang et al. [41]). Among these references, Fortz and Thorup [20], Giovanni et al. [15], Lin and Wang [30], Parmar et al. [32], and Wang et al. [41] use single stage solution methods where they include the link metric in the optimization of some performance measure whereas Bley and Koch [9], Broström and Holmberg [12], and Pióro et al. [33] implement two stage algorithms where they determine an optimal routing initially and then search for a weight metric to make this routing the shortest path routing. Unfortunately, two-stage methods do not guarantee that the second stage problem will be feasible. Except some special cases (41) there is no complete description of routing configurations which can be realized as shortest-paths, but some necessary conditions have been studied in [3] [6] [8] [10] [11] [13] [14].
Being inspired by the available studies on oblivious routing, we aim to provide a new expansion of the problem. The current work differs from what is available in the literature in several dimensions. Firstly, Ben-Ameur and Kerivin [7] use a polyhedral definition of demands for the general routing problem, where there is no restriction on the structure of the paths or how the flow is split among multiple paths. However, we study OSPF routing with ECMP for polyhedral demands. Although Mulyana and Killat [31] discuss oblivious OSPF routing problem, they consider a rather restricted case where there can only be outbound or inbound constraints, but not both. On the other hand, we consider the general case such that our models and methods can easily be used to handle any polyhedral definition of demand uncertainty. Moreover, we provide an alternative approach to the problem of Altın et al. [2], where they use a quotient min-max regret performance measure based on link utilization and employ pure mathematical programming tools to model and solve the problem. In the current work, we use the cost function of Fortz and Thorup [20] and extend their tabu search algorithm to handle polyhedral demands. Since the cost of routing on each link is an increasing function of link utilization, the final routing configuration would also be fair in terms of the work load distribution. Moreover, Altın et al. [2] assume that any traffic matrix in the demand polyhedron can be routed without violating the arc capacities. In this study, we allow capacity violations but punish such overuses in the objective function.

To put in a nutshell, we discuss OSPF by optimizing link weights to design a routing configuration that is able to handle ‘applicable’ changes in demand estimates in the least costly and most fair way.

The rest of the paper is organized as follows. We outline the problem and present a compact MIP model in Section 2. We discuss our tabu search based algorithm in Section 3 and continue with some test results in Section 4. We conclude the paper with a summary of our study in Section 5.

2 Problem definition and model

Let $G = (V, A, c)$ be a capacitated backbone graph with node set $V$, directed arc set $A$, and a capacity $c_{ij} > 0$ for each arc $(i, j) \in A$. Let $Q \subseteq \{(s, t) : s, t \in V, t \neq s\}$ be the nonempty set of commodities where each commodity $(s, t) \in Q$ is defined with its source $s$ and destination $t$. A traffic matrix (TM) $d \in \mathbb{R}^{|Q|}$ keeps the demand information for each commodity in $Q$. The motivation of the general routing problem is to choose paths on $G$ so as to route the demand for each commodity from its source to its destination by operating the network as efficiently as possible. Several criteria can be used to measure the effectiveness of a routing configuration. In this study, we will adopt cost minimization, where the routing cost for each arc is an increasing convex function of its utilization. More precisely, with each arc $(i, j) \in A$, we associate a cost function $\Phi_{(i,j)}(l_{ij})$ of the load $l_{ij}$, depending on how close the load is to the capacity $c_{ij}$. We assume in the following that $\Phi_{ij}$ is an increasing piecewise linear function (as in [20, 22]).

Our routing problem is quite different than the general problem since we impose several restrictions on the structure of the paths that we use and consider a polyhedron of feasible TMs rather than a single one. Let us lay these restrictions aside for the time being and start with the general case. Suppose that we are given a single TM $d$ and we can use arbitrary paths. Then, the mathematical model of the
corresponding routing problem is

\[
\begin{align*}
\min & \sum_{(i,j) \in A} \phi_{ij} \\
\text{s.t.} & \sum_{j:(i,j) \in A} f_{ij}^t - \sum_{j:(j,i) \in A} f_{ji}^t = \begin{cases} 
1 & i = s \\
-1 & i = t \\
0 & \text{otherwise} 
\end{cases} \quad i \in V, (s,t) \in Q
\end{align*}
\]

where \( f_{ij}^t \) is the fraction of the traffic \( d_{st} \) from origin \( s \) to destination \( t \) routed on the arc \((i,j) \in A\), \( l_{ij} \) is the total traffic on \((i,j) \in A\), and \( \phi_{ij} \) is the routing cost for the arc \((i,j) \in A\). Constraints (2) are the flow conservation constraints. Inequalities (3) define the load on each arc and inequalities (4) define the cost on each arc, where \( Z \) is the set of break points of the piecewise linear function and \( u_z \) and \( v_z \) are the coefficients of the corresponding segment. Obviously, (1)-(6) is an LP and we can solve it in polynomial time. However, the problem gets more complicated as we move to the oblivious OSPF routing problem.

Let us first relax the assumption of a fixed \( TM d \) and consider a polyhedron \( D \) of feasible demand realizations. Now, the concern is to determine paths such that any \( d \in D \) can be accommodated efficiently. This means that our ‘optimal’ routing will have the best worst case performance for \( D \) independent of a specific \( TM \). In terms of the mathematical model, the main impact of such a shift will be on the link capacity constraint (3), since the traffic load on each link \((i,j) \in A\) is defined as a function of \( d \), which can be any vector in \( D \). Consequently, for the polyhedral case, (3) changes as

\[
l_{ij} \geq \sum_{(s,t) \in Q} d_{st} f_{ij}^t \quad d \in D, (i,j) \in A
\]

where \( D \) is an arbitrary polyhedron. However, this change leads to a semi-infinite optimization problem. We eliminate this difficulty in Proposition 1 by using a duality transformation, that is similar to the one mentioned in Soyster [35] and also used in Altun et al. [1].

Proposition 1. Let \( D = \{d \in \mathbb{R}^{|Q|} : Ad \leq \alpha, d \geq 0\}, A \in \mathbb{R}^{|K||Q|}, \text{ and } \alpha \in \mathbb{R}^K \). The general routing problem with polyhedral demands can be modeled as (GRD)

\[
\begin{align*}
\min & \sum_{(i,j) \in A} \phi_{ij} \\
\text{s.t.} & \quad l_{ij} \geq \sum_{k=1}^K \alpha_k \lambda_k^{ij} \quad (i,j) \in A \\
& \quad \sum_{k=1}^K \alpha_k \lambda_k^{ij} \geq f_{ij}^t \quad (i,j) \in A, (s,t) \in Q \\
& \quad (2), (4) - (6) \\
& \quad \lambda_k^{ij} \geq 0 \quad (i,j) \in A, k = 1, \ldots, K
\end{align*}
\]

where \( \lambda_k^{ij} \) are the dual variables used in the transformation.

Proof. Let \( D = \{d \in \mathbb{R}^{|Q|} : Ad \leq \alpha, d \geq 0\} \) be the polytope of feasible TMs with \( A \in \mathbb{R}^{|K||Q|} \) and \( \alpha \in \mathbb{R}^K \). Then (7) implies that

\[
l_{ij} \geq \max_{d \in \mathcal{D}} \sum_{(s,t) \in Q} d_{st} f_{ij}^t \quad (i,j) \in A.
\]
The maximization problem on the right hand side is always feasible and bounded. Thus, for each arc \((i, j) \in A\), we can apply a duality transformation similar to the one by Soyster [35] to get

\[
 l_{ij} \geq \min \sum_{k=1}^{K} \alpha_k \lambda_k^{ij} \quad (13)
\]

\[
 \sum_{k=1}^{K} \alpha_k \lambda_k^{ij} \geq f_{ij}^t \quad (s, t) \in Q \quad (14)
\]

\[
 \lambda_k^{ij} \geq 0 \quad k = 1, ..., K, \quad (15)
\]

where \(\lambda\) is the corresponding vector of dual variables. After replacing \((7)\) in \(GR_D\) with \((13)-(15)\), we can also remove the \(\min\) in \((13)\) since \((1)\) minimizes the sum of \(\Phi_{ij}\), which are piecewise linear increasing functions of \(l_{ij}\) \((20)\).

Proposition 1 reveals that the oblivious unconstrained routing can be determined in polynomial time by solving the compact linear programming problem \(GR_D\). However, \(GR_D\) has no restriction on the structure of the paths or how flow should be split among multiple paths. Per contra, OSPF stipulates the demand for each commodity to be routed on the shortest paths between its origin and destination nodes in accordance with technical restrictions on traffic splitting. Moreover, we want to choose the link metric, which makes the shortest paths with equal load sharing the optimal routes. Hence, we need additional constraints in our final model.

**Proposition 2.** Let \(W = \{s \in V : \exists (s, t) \in Q, t \in V \setminus \{s\}\}\). The oblivious OSPF routing problem under polyhedral demand uncertainty with equal load sharing can be modeled by appending the following set of constraints in \(GR_D\):

\[
f_{ij}^t \leq y_{ij}^t \quad (i, j) \in A, (s, t) \in Q \quad (16)
\]

\[
y_{ij}^t + \rho_j^t - \rho_i^t + \omega_{ij} \geq 1 \quad (i, j) \in A \quad (17)
\]

\[
-y_{ij}^t - \frac{\rho_j^t - \rho_i^t + \omega_{ij}}{2\omega_{\text{max}}} \geq 1 \quad (i, j) \in A, t \in W \quad (18)
\]

\[
f_{ij}^t \leq \phi_h^t \quad (i, j) \in A, (s, t) \in Q \quad (19)
\]

\[
1 + f_{ij}^t - \phi_h^t \geq y_{ij}^t \quad (i, j) \in A, (s, t) \in Q \quad (20)
\]

\[
1 \leq \omega_{ij} \leq \omega_{\text{max}} \quad \text{integer} \quad (i, j) \in A \quad (21)
\]

\[
y_{ij}^t \in \{0, 1\} \quad (i, j) \in A, t \in W \quad (22)
\]

\[
0 \leq \phi_h^t \leq 1 \quad i \in V, (s, t) \in Q \quad (23)
\]

where \(y_{ij}^t\) indicates if arc \((i, j)\) is used in a shortest path to destination node \(t\), \(\omega_{ij}\) is the weight of arc \((i, j)\), and \(\omega_{\text{max}} = 65,535\). Constraints \((17)\) and \((18)\) model OSPF routing whereas \((19)\) and \((20)\) ensure that flow is split equally among multiple shortest paths. Finally, \((21)\) guarantees that link metric is chosen in accordance with OSPF technology \((2)\).

Obviously, solving the compact MIP model \(OSPFP_{\Phi} := \min_{(f, l, \lambda, y, p, \omega, \varphi)} \sum_{(i, j) \in A} \phi_{ij}\) where \(\Psi = \{(f, l, \lambda, y, p, \omega, \varphi) : (2), (3) - (5), (9) - (11), (10) - (23)\}\) using available solvers is not reasonable even for small sized networks given that the numbers of variables and constraints are both of \(O(|V|^4)\), and the linear relaxations of these models are usually very weak. Hence, in the next section, we propose an efficient heuristic approach.

### 3 Heuristic approach to oblivious OSPF routing

Fortz and Thorup [20][22] proposed a tabu search algorithm for managing OSPF weights for a fixed \(TM\), where optimized weights support up to 110% more demand compared to Cisco recommended heuristic
weights, i.e. weights inversely proportional to link capacities. Following this study, Fortz and Thorup extend their algorithm in order to optimize OSPF weights for a discrete set of multiple TMs with the objective of capturing the changes in TMs \[21\]. In this regard, the tabu search heuristic is input several TMs to minimize

\[
\min \Phi(TM_1, \ldots, TM_k) = \sum_{i=1}^{k} \Phi(TM_i)
\]  

(24)

Recently, an opensource version of this tabu search approach, called IGP-WO, handling multiple demand matrices, was implemented in the TOTEM opensource toolbox \[29\] \[23\] \[39\].

In this section, we discuss our algorithmic approach to tackle polyhedral demands. It has two main steps, namely the TM enumeration and weight optimization. We use IGP-WO, the TOTEM weight optimizer, for the latter step, whereas we use mathematical programming for the first part. As the representation theorem for polytopes suggests, any TM \(d \in \mathcal{D}\) can be represented as a convex combination of the extreme points of \(\mathcal{D}\). Hence, we could equally write the link load constraint \((7)\) for each extreme point of \(\mathcal{D}\), which are in finite but exponential number. In Section 2, we use the duality transformation in Proposition 1 to overcome this difficulty and provide a compact formulation. As an alternative to that, we propose an algorithmic approach in this section.

For a given routing \(f\), the motivation in the TM generation step is to enumerate the extreme points of \(\mathcal{D}\) which correspond to the ‘most challenging’ traffic demands in terms of either the arc utilization or the routing cost. Since \(\mathcal{D}\) is a polyhedral set, the algorithm will terminate after a finite number of iterations. Besides, the greedy choice of extreme points would lead to much fewer iterations before termination. We provide the pseudo codes of two different strategies in Algorithm 1 and Algorithm 2, respectively. We will use \(a\) and \((i, j)\) to denote an arc in \(A\) interchangeably throughout this section.

The first step INITIALIZE and the final step CHALLENGE are common for the two strategies. To start, we need an initial \(d_0 \in \mathcal{D}\). To create it, in INITIALIZE, we solve maximization problem \((12)\) for an arbitrary arc \(a \in A\) by setting \(f_{st}^a = \alpha\) for all commodities \((s, t) \in Q\), where \(\alpha\) can be any positive constant. Then we create \(\tilde{\mathcal{D}}\) to hold all the TMs that we generate throughout the algorithm. On the other hand, the aim of the step CHALLENGE is to determine a ‘challenge’ case to compare the routing cost and maximum link utilization of the two routings. For this purpose, we take \(d_{\text{max}} = \argmax_{d \in \mathcal{D}} \sum_{(s,t) \in Q} d_{st}\) as our challenge TM. Notice that we choose \(d_{\text{max}}\) independent of any performance measure or any topological information. At this stage, we are only interested in the TM that requires the utmost use of network resources. Although we could use some other criteria at this stage, we believe the current choice is fair enough since we will use \(d_{\text{max}}\) for comparison. In between INITIALIZE and CHALLENGE, there is the MAIN step, where the two strategies differentiate.

The first strategy is based on the greedy search of a new feasible TM based on total routing cost. Algorithm 1 outlines this strategy, which we call CM in the rest of the paper. At iteration \(cnt\) of CM, we have an OSPF routing \(g^*\) with the minimum average routing cost \(\Phi_{\tilde{D}}\) for the TMs in \(\tilde{D}\). The question we want to answer is: Does there exist another demand \(d \in \mathcal{D} \setminus \tilde{D}\) that costs more than \(\Phi_{\tilde{D}}\), if we route
it using $g^*$? To tackle this question, we first solve the following MIP model ($P_{\text{MaxCost}}$):

$$\max \sum_{(i,j) \in A} \phi_{ij}$$

s.t. $\phi_{ij} - u_z l_{ij} \leq M(1 - y_{ij}^z) - v_z c_{ij}$ $(i, j) \in A, \ z \in Z$

$$\sum_{z \in Z} y_{ij}^z = 1 \quad (i, j) \in A$$

$$l_{ij} - \sum_{(s, t) \in Q} d_{st} g_{ij}^{st} = 0 \quad (i, j) \in A$$

$$\sum_{(s, t) \in Q} a_{st} d_{st} \leq \alpha_k \quad k = 1, \ldots, K$$ (25)

$$\phi_{ij}, l_{ij} \geq 0 \quad (i, j) \in A$$

$$y_{ij}^z \in \{0, 1\} \quad (i, j) \in A, \ z \in Z$$

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

where $y_{ij}$ variables show the segment of the objective function that each $\phi_{ij}$ lies in and (25) ensures that we obtain a feasible TM $d^{\text{new}} \in \mathcal{D}$. Notice that $g^*$ is not a variable anymore in $P_{\text{MaxCost}}$. Thus, the link load $l_{ij}$ is defined as a linear function of the demand variables $d \in \mathcal{D}$ with coefficients $g_{ij}^{st}$ obtained in the most recent TABU iteration. In consequence, the solution of $P_{\text{MaxCost}}$ will be the worst case TM $d^{\text{new}}$ leading to the highest routing cost $\sum_{a \in A} \phi_a^*$ for $g^*$.

Since $\mathcal{D}$ is nonempty, $P_{\text{MaxCost}}$ will always yield a feasible TM $d^{\text{new}}$. However, there is no guarantee that we will get a new $d^{\text{new}} \notin \hat{D}$ at each iteration since (25) ensures $d^{\text{new}} \in \mathcal{D}$ but not $d^{\text{new}} \in \mathcal{D} \setminus \hat{D}$. To shun fake updates, we keep track of all matrices in $\hat{D}$ using a hashing table. This is similar to what we use to avoid cycling in the tabu search algorithm for optimizing link weights. In brief, we use a hashing function to map each $d^{\text{new}}$ to an integer $h_{d^{\text{new}}}$ and we mark its generation in the $h_{d^{\text{new}}}$ entry of a boolean table. Each time we solve $P_{\text{MaxCost}}$, we decide whether or not we should update $\hat{D}$ using the boolean table and continue with the next iteration only if we have a new TM $d^{\text{new}} \notin \hat{D}$ for which the routing cost $\sum_{a \in A} \phi_a^*$ is higher than the current average cost $\Phi_{\hat{D}}$.

On the other hand, the second strategy is greedy in the sense of traffic load on arcs. It uses link utilization as the determining factor for new TM generation. Basically, given an OSPF routing $g^*$ optimal for $\hat{D}$, it looks for a TM $d^{\text{new}}$, which makes some arc $a \in A$ overloaded or increases the current congestion rate of the network, that is $U_{\text{max}} = \max_{a \in A, d \in \mathcal{D}} d_{a}$. We use the hashing function that we have described above to keep track of the TMs in $\hat{D}$ and avoid cycling. A framework of this strategy is provided in Algorithm 2. We will refer this strategy as LM from now on.

Both algorithms are implemented and tested on several instances. We provide our results in the next section but first we would like to make some remarks on our algorithm and two strategies. We prefer to use either congestion rate or routing cost but not both in the MAIN stage of the two approaches. This is primarily for the sake of consistency in the optimization process. However, this does not mean that we focus on just one dimension and ignore the other. For example, in CM, we pursue costly TMs. Our routing cost is an increasing piecewise linear function of the link utilization. Thus, a costly TM would increase the traffic load and hence the utilization rate for some links. On the other hand, a higher link utilization will increase the routing cost by definition. Obviously, we may use several hybrid strategies by incorporating the two measures explicitly in the decision process. But our preliminary tests show that we do not gain any significant benefit by doing so. Therefore, we prefer to continue with the two strategies.

The main difference between CM and LM is the domain of the challenge. CM generates a demand matrix $d^*$ that puts the network in a worse situation as a whole for a given routing configuration on the basis of the total routing cost. On the contrary, in LM, the new TM is at least ‘locally’ challenging, since we consider the worst case for each arc individually. In both strategies, we enumerate at most one TM at each iteration. However, we can modify Algorithm 2 easily to generate multiple TMs, namely at
Algorithm 1 Strategy 1 with Cost Maximization - CM

**Require:** directed graph $G = (V, A)$, traffic polytope $\mathcal{D}$, link capacity vector $c$;

**Ensure:** minimum cost OSPF routing $f^*$ and metric $\omega^*$ for $(G, \mathcal{D}, c)$;

**INITIALIZE:**
- Find an initial feasible TM $d_0 \in \mathcal{D}$;
- $d^{rec} \leftarrow d_0$; // $d^{rec}$: the most recently enumerated TM;
- $\tilde{\mathcal{D}} \leftarrow d_0$; // $\tilde{\mathcal{D}}$: current set of TMs enumerated so far;
- $NewTM = TRUE$;
- $cnt = 0$;

**MAIN:**
- **while** ($cnt \leq cnt\_limit$) and ($NewTM = TRUE$) **do**
  - **TABU:**
    - Find an optimized oblivious OSPF routing $g^*$ for $\tilde{\mathcal{D}}$ and the associated metric $\omega_T^*$;
    - Get $\Phi_{\tilde{\mathcal{D}}}$: the average routing cost for $\tilde{\mathcal{D}}$;
    - $NewTM = FALSE$;
    - Solve $P_{MaxCost}$ to get $\sum_{a \in A} \phi_a^*$ and $d^{new}$;
    - **if** $\sum_{a \in A} \phi_a^* > \Phi_{\tilde{\mathcal{D}}}$ and $d^{new} \notin \tilde{\mathcal{D}}$ **then**
      - $\tilde{\mathcal{D}} \leftarrow d^{new}$;
      - $NewTM = TRUE$;
    - $cnt \leftarrow cnt + 1$;
    - $f^* \leftarrow g^*$;
    - $\omega^* \leftarrow \omega_T^*$;
  - **CHALLENGE:**
    - Find the challenge TM $d^{max} = \text{argmax}_{d \in \mathcal{D}} \sum_{(s,t) \in Q} d_{st}$;
    - Get $\Phi_{d^{max}}$ // the cost of routing $d^{max}$ with $f^*$;
    - Get $U_{d^{max}}$ // the congestion rate for $d^{max}$ with $f^*$;

most one for each arc. Finally, each time the algorithm performs a tabu search, it starts with the optimal weight metric of the most recent iteration. This is useful to reduce the time spent for re-optimizing the weight metric in the **TABU** stage.

4 Computational experiments

In this section, we report our test results for our weight-managed oblivious OSPF routing problem. We use the hose model of demand uncertainty ([16],[19]), which is very widely used especially in the telecommunications network design literature.

4.1 Hose Model

Hose model was proposed by Duffield et al. [16] and Fingerhut et al. [19] independently. The motivation is to enhance the current network management efforts by eliminating the dependence of final designs on a specific estimation of average system behavior. In the hose model, each node is assigned an outgoing and incoming traffic bandwidth capacity. Then for $G = (V, A)$, a TM $d$ is feasible if it satisfies

$$
\sum_{t \in W \setminus \{s\}} d_{st} \leq b^+_s \quad s \in W
$$

(26)

$$
\sum_{t \in W \setminus \{s\}} d_{ts} \leq b^-_s \quad s \in W
$$

(27)

where $W \subseteq V$ is the set of nodes called terminals who want to exchange traffic with the rest of the nodes in $W$, whereas $b^+_s$ and $b^-_s$ are the outflow and inflow capacities of terminal $s$, respectively. This
Algorithm 2 Strategy 2 with Arc Load Maximization - LM

Require: directed graph \( G = (V, A) \), traffic polytope \( \mathcal{D} \), link capacity vector \( c \);
Ensure: minimum cost OSPF routing \( f^* \) and metric \( \omega^* \) for \( (G, \mathcal{D}, c) \);

\text{INITIALIZE} // As in Algorithm 1

\text{MAIN:}
while \((cnt \leq cnt\_limit) \) and \((\text{NewTM} = \text{TRUE}) \) do
    \text{TABU:} Find an optimized oblivious OSPF routing \( g^* \) for \( \tilde{D} \) and the associated metric \( \omega^*_T \);
    \( U_{\max} = \) maximum link utilization for \( \tilde{d}^{\text{rec}} \);
    \( \text{NewTM} = \text{FALSE} \);
    \( a = 0 \) // start with the first arc of \( G \);
    while \((a < |A|) \) and \((\text{NewTM} = \text{FALSE}) \) do
        \( d^{\text{new}} = \operatorname{argmax}_{d \in \mathcal{D}} (g^*_a d) \); // \( d^{\text{new}} \): worst case TM for \( a \) with routing \( g^* \);
        if \((g^*_a d^{\text{new}} > c_a) \) or \((\frac{2 d^{\text{new}}}{c_a} > U_{\max}) \) then
            if \( d^{\text{new}} \notin \tilde{D} \) then
                \( \tilde{d}^{\text{rec}} = d^{\text{new}} \);
                \( \tilde{D} = \tilde{d}^{\text{rec}} \);
                \( \text{NewTM} = \text{TRUE} \);
                \( cnt = cnt + 1 \);
            if \( \text{NewTM} = \text{FALSE} \) then
                \( a = a + 1 \);
            if \( \text{NewTM} = \text{TRUE} \) then
                \( f^* = g^* \);
                \( \omega^* = \omega^*_T \);
            \end{if}
        \end{if}
    \endwhile
\endwhile

\text{CHALLENGE} // As in Algorithm 1
specific case is known as the Asymmetric Hose in the literature. There are also the Symmetric and Sum-Symmetric models. In the Symmetric case, a single bandwidth \( b_s \) is defined for the total flow incident to node \( s \) such that \( \sum_{t \in W \setminus s} (d_{st} + d_{ts}) \leq b_s \), \( s \in W \). On the other hand, \( \sum_{s \in W} b^+_s = \sum_{s \in W} b^-_s \) in the Sum-Symmetric case. In our tests, we use the Asymmetric definition.

The motivation of optimizing for a set of demands rather than a single TM is to make resource management more flexible to cope with possible changes in the forecasted demand realizations. In addition to its accuracy, the amount of information used to describe the likely behaviors of the network has an impact on the practicality, efficiency, and the robustness of the final design. On this account, the hose model is a very powerful tool since it relies on cumulative bandwidth capacities, which can be more reliably and easily estimated than individual demand expectations. Given this and several other advantages, the hose model has gained significant attention especially in the telecommunications literature \[7, 17, 18, 24, 25, 26, 27, 28\].

4.2 Experimental Results

In this section, we provide our test results for the two algorithmic settings CM and LM. We perform our tests on 11 instances among which bhvac, pacbell, eon, metro, and arpanet are from the IEEE literature. On the other hand, exodus, abovenet, ensl, and telstra are from the Rocketfuel project \[36\].

We have the data for the topology \(|V|\) and \(|A|\), the arc weights \(|w|\), and the number of data packets entering and leaving each node for the Rocketfuel instances. In order to get the arc capacities, we assume that the given weight metric \( w \) obeys the Cisco setting. So we set the capacity of each arc \((i, j) \in A\) as \( c_{ij} = 1/w_{ij} \). Moreover, we use the gravity model \([33]\) to generate the outflow and inflow bandwidth capacities. The framework of this method is as follows: First, we associate a repulsion \((R_h)\) and an attraction \((A_h)\) coefficient to each node \( h \in V \) in terms of the number of packets entering and leaving \( h \) \[30\]. Then, we choose the nodes with highest repulsion or attraction coefficients as the terminals to create \( W \subseteq V \) and let \( Q = \{(s, t) : s \in W, t \in W \setminus \{s\}\} \). Next, we determine a base demand estimate \( d_{st} \) for each \((s, t)\) pair as \( d_{st} = \beta R_h A_h \) where \( \beta = \eta \gamma^* \). We control the proximity of \( \tilde{d} \) to the boundary of the feasible set of applicable TMs via \( \eta \in [0, 1] \). On the other hand, we ensure that \( \tilde{d} \) is feasibly routed by means of \( \gamma^* \), which is the optimal solution of

\[
\begin{align*}
\max_{\gamma} & \quad \gamma A_h \\
\text{s.t.} & \quad \sum_{(i,j) \in A} u^s_{ij} - \sum_{(j,i) \in A} u^t_{ji} = \begin{cases} 
\gamma R_h A_h & h = s \\
-\gamma R_h A_h & h = t \\
0 & \text{otherwise}
\end{cases} \\
& \quad \sum_{(s,t) \in Q} u^s_{ij} \leq c_{ij} \quad (i,j) \in A \\
& \quad u^t_{ij} \geq 0 \quad (i,j) \in A, (s,t) \in Q
\end{align*}
\]

Consequently, we randomly create \( S \subseteq W \) such that \(|S| = \left\lceil \frac{|W|}{2} \right\rceil \). Finally, we set the inflow and outflow bandwidth of each node as \( b^+_s = \sum_{t \in W \setminus \{s\}} d_{st} / 1.1 \) and \( b^-_s = 1.1 \sum_{t \in W \setminus \{s\}} d_{ts} / 1.1 \) for all \( s \in S \) whereas \( b^+_s = 1.1 \sum_{t \in W \setminus \{s\}} d_{st} / 1.1 \) and \( b^-_s = \sum_{t \in W \setminus \{s\}} d_{ts} / 1.1 \) for all \( s \in W \setminus S \).

We implement the algorithm in C and use Cplex 11.0 to solve the maximization problems in CM and LM. Moreover, we determine the initial TM \( d_0 \) by choosing \( \alpha = 1 \) for the arc with the smallest index in the INITIALIZE step. In the MAIN stage of both settings, we choose \( cnt\_limit \) as 50. However, for CM, we had to reduce \( cnt\_limit \) to 5 and 10 in con and arpanet to avoid excessive solution times. We provide our test results for the two strategies in Table1 and Table2 where we have the following entries:

- topology of the network, i.e., the number of nodes \(|V|\), the number of arcs \(|A|\), and the number of terminal nodes \(|W|\);

\(^{1}\)Notice that \( R_i = A_i = 0 \) for all \( i \in V \setminus W \).
• the number of TMs enumerated throughout the algorithm, i.e., $|\tilde{D}|$;
• the average routing cost for $\tilde{D}$ at termination, i.e., $\Phi_{\tilde{D}}$;
• the routing cost for the final TM, $d^{rec}$, added to $\tilde{D}$, i.e., $\Phi_F$;
• the normalized cost for $d^{rec}$, i.e., $\Phi_f^{norm} = \frac{\Phi_F}{\Phi_U}$ where $\Phi_U$ is the cost of routing $d^{rec}$ if all arcs in $A$ had a unit length and unlimited capacity;
• the maximum utilization rate at termination, i.e., $U_{\text{max}} = \max_{a \in A, d \in \tilde{D}} \frac{d_a}{c_a}$;
• time elapsed till termination, i.e., $t$.

Notice that $\Phi_{\tilde{D}}$ and $\Phi_F$ are absolute measures of how good the final routing $f^*$ performs for each instance. However, it would be more fair to use a measure, which is neutral to network topology as much as possible. Hence for the most recently enumerated TM, we also display the normalized cost $\Phi_f^{norm}$. To calculate the normalizing factor $\Phi_U$, we relax the arc capacity constraints and assume that all link weights are 1. Then $\Phi_U$ is the cost of routing the final TM along the shortest paths on this uncapacitated network. By definition, $\Phi_f^{norm}$ shows how good our weight-managed OSPF routing performs on each capacitated network when compared to the unit-weight OSPF routing without capacity restrictions. Hence, smaller values of $\Phi_f^{norm}$ indicate superior performance. Nevertheless, it takes a value of 1 when we can route the final TM such that the load on each arc is less than 1/3rd of its capacity.

| Instance | $|V|$ | $|A|$ | $|W|$ | $|D|$ | $\Phi_{\tilde{D}}$ | $\Phi_F$ | $\Phi_f^{norm}$ | $U_{\text{max}}$ | $t$ (sec) |
|----------|-----|-----|-----|-----|--------|--------|----------------|-------------|---------|
| exodus   | 7   | 12  | 7   | 2   | 844.5  | 877.16 | 30.28          | 4.53        | 1       |
| nsf      | 8   | 20  | 5   | 5   | 2961.73 | 3550.52 | 0.26           | 0.96        | 9       |
| vnsl     | 9   | 22  | 3   | 2   | 170,331.3 | 170,331.3 | 0.25          | 0.83        | 2       |
| example  | 10  | 30  | 4   | 3   | 2409.8 | 10,630.33 | 16.89         | 1.25        | 8       |
| metro    | 11  | 84  | 5   | 6   | 528.84 | 899.07   | 0.27           | 0.69        | 78      |
| bhvac    | 19  | 44  | 11  | 5   | 26,268,982.1 | 27,638,469.33 | 401.91      | 49.25      | 67      |
| abovenet | 19  | 68  | 5   | 4   | 708.84 | 725.28   | 105.67        | 2.36        | 71      |
| telstra  | 44  | 88  | 7   | 2   | 0.28   | 0.31     | 0.12           | 0.88        | 200     |
| pacbell  | 15  | 42  | 7   | 5   | 2370.83 | 2671.5   | 0.15           | 0.84        | 111     |
| econ     | 19  | 74  | 15  | 5   | 11,734,977.88* | 16,889,017.5* | 214.6*      | 6.45*       | 8135*   |
| arpanet  | 24  | 100 | 10  | 10  | 353,069.59* | 470,151.65* | 12.12         | 1.5*        | 124,074* |

Table 1: Results for CM under the hose demand uncertainty model.

Table 1 shows that the algorithm stops after enumerating much fewer TMs than the cntLimit of 50. Although the $\Phi_f^{norm}$ entries are relevant for the most recent TM, we see that large values of $\Phi_f^{norm}$ are accompanied by large $U_{\text{max}}$ values and vice versa. Hence, these two columns together provide us not only the necessary information on the relative and absolute performances of the final routings but also give a hint about the sufficiency of the current arc capacities. High entries in these columns suggest the existence of some bottleneck arcs for which the traffic engineering tools cannot help much and they would anyway be overloaded to route some feasible TMs.

Firstly, we can observe from the $\Phi_f^{norm}$ column that our traffic engineering efforts have improved the relative performance of the final routing significantly in nsf, vnsl, metro, telstra, and pacbell. On the other hand, we observe high $\Phi_f^{norm}$ and $U_{\text{max}}$ values for exodus, example, bhvac, and abovenet. We believe this is due to the need for capacity expansion rather routing poorly. Let us consider exodus, which is a simple example supporting our conclusion. In exodus, all nodes are terminal nodes with one incoming and one outgoing arc. So, no matter what the link weights are, there is a unique path for each demand and not much space for traffic engineering. In the optimal solution, the most congested link is adjacent to node X whose outflow bandwidth is 4.53 times larger than the capacity of the single
outgoing link. Hence, in any feasible TM where $\sum_{t \in W} d_{Xt} = b^+_X$, this link will become overloaded as expected. Moreover, eon and arpanet are difficult instances for CM. Since the time to solve $P_{MaxCost}$ is relatively long, we had to stop the algorithm after 5 and 10 iterations, respectively.

Finally, we display how the objective function value of $P_{MaxCost}$ has changed at each iteration in some instances in Figure 1. Except one-time increases in metro and arpanet, we observe a downward trend in all graphics. This also gives an idea about how the average cost of routing changes throughout the algorithm.

![Figure 1: Change in $\sum_{a \in A} \phi^*_a$ for CM.](image)

We present our test results for LM in Table 2. $\Phi_{form}$ shows that LM performs significantly better than the unit-weight OSPF routing on uncapacitated networks in 5 of the instances, namely nsf, vnsl, metro, telstra, and pacbell whereas as good as it in example. We also observe that $\Phi_{form}$ and $U_{max}$ follow a similar trend as in CM. Moreover, the algorithm had to stop after 50 iterations in bhvac and eon.

Table 1 and Table 2 provide us some information for making a rough comparison between CM and LM. Unsurprisingly, we had to enumerate more TMs in LM than CM on the average. This is a consequence of the difference in the domain of impact for each enumeration. As we have discussed in Section 3, LM generates at least ‘locally challenging’ TMs since it considers arcs one by one. Moreover, CM routes $\tilde{D}$
Table 2: Results for LM under the hose demand uncertainty model.

<table>
<thead>
<tr>
<th>Instance</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>(\Phi_D)</th>
<th>(\Phi_F)</th>
<th>(\Phi_{norm})</th>
<th>(U_{max})</th>
<th>t (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>exodus</td>
<td>7</td>
<td>12</td>
<td>7</td>
<td>2</td>
<td>841.07</td>
<td>837.66</td>
<td>178.09</td>
<td>4.53</td>
<td>1</td>
</tr>
<tr>
<td>nsf</td>
<td>8</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>1373.3</td>
<td>1219.33</td>
<td>0.77</td>
<td>0.96</td>
<td>3</td>
</tr>
<tr>
<td>vnsl</td>
<td>9</td>
<td>22</td>
<td>3</td>
<td>1</td>
<td>170,331.3</td>
<td>170,331.3</td>
<td>0.25</td>
<td>0.83</td>
<td>1</td>
</tr>
<tr>
<td>example</td>
<td>10</td>
<td>30</td>
<td>4</td>
<td>7</td>
<td>2236.8</td>
<td>85.33</td>
<td>1</td>
<td>1.25</td>
<td>23</td>
</tr>
<tr>
<td>metro</td>
<td>11</td>
<td>84</td>
<td>5</td>
<td>23</td>
<td>140.61</td>
<td>99.5</td>
<td>0.20</td>
<td>0.73</td>
<td>1029</td>
</tr>
<tr>
<td>bhvac</td>
<td>19</td>
<td>44</td>
<td>11</td>
<td>51</td>
<td>7,547,647.45*</td>
<td>3,385,313.33*</td>
<td>368.18*</td>
<td>50.7*</td>
<td>3084*</td>
</tr>
<tr>
<td>abovenet</td>
<td>19</td>
<td>68</td>
<td>5</td>
<td>1</td>
<td>254.43</td>
<td>106.85</td>
<td>46.8</td>
<td>3.19</td>
<td>440</td>
</tr>
<tr>
<td>telstra</td>
<td>44</td>
<td>88</td>
<td>7</td>
<td>1</td>
<td>603.72</td>
<td>635</td>
<td>0.18</td>
<td>0.84</td>
<td>485</td>
</tr>
<tr>
<td>pacbell</td>
<td>15</td>
<td>42</td>
<td>7</td>
<td>23</td>
<td>1,068,540.68*</td>
<td>1,991,960.17*</td>
<td>110.63*</td>
<td>6.80*</td>
<td>10,500*</td>
</tr>
<tr>
<td>eon</td>
<td>19</td>
<td>74</td>
<td>15</td>
<td>51</td>
<td>2,236,840.68*</td>
<td>1,991,960.17*</td>
<td>110.63*</td>
<td>6.80*</td>
<td>10,500*</td>
</tr>
<tr>
<td>arpanet</td>
<td>24</td>
<td>100</td>
<td>10</td>
<td>45</td>
<td>73,835.82</td>
<td>185,369.83</td>
<td>13.69</td>
<td>1.5</td>
<td>18,331</td>
</tr>
</tbody>
</table>

Table 3: CM versus LM in the CHALLENGE step.

<table>
<thead>
<tr>
<th>Instance</th>
<th>(\Phi_{CM})</th>
<th>(U_{max}^{CM})</th>
<th>(\Phi_{LM})</th>
<th>(U_{max}^{LM})</th>
</tr>
</thead>
<tbody>
<tr>
<td>exodus</td>
<td>844.48</td>
<td>4.53</td>
<td>844.48</td>
<td>4.53</td>
</tr>
<tr>
<td>nsf</td>
<td>2656.70</td>
<td>0.88</td>
<td>2037.73</td>
<td>0.76</td>
</tr>
<tr>
<td>vnsl</td>
<td>170,331.3</td>
<td>0.83</td>
<td>170,331.3</td>
<td>0.83</td>
</tr>
<tr>
<td>example</td>
<td>522.83</td>
<td>1.1</td>
<td>533.83</td>
<td>1.1</td>
</tr>
<tr>
<td>metro</td>
<td>464.71</td>
<td>0.57</td>
<td>455.67</td>
<td>0.57</td>
</tr>
<tr>
<td>bhvac</td>
<td>24,166,769.6</td>
<td>36.3</td>
<td>24,273,340.6</td>
<td>36.95</td>
</tr>
<tr>
<td>abovenet</td>
<td>659.53</td>
<td>2.09</td>
<td>689.55</td>
<td>2.66</td>
</tr>
<tr>
<td>telstra</td>
<td>0.26</td>
<td>0.88</td>
<td>0.26</td>
<td>0.88</td>
</tr>
<tr>
<td>pacbell</td>
<td>2025</td>
<td>0.65</td>
<td>2025</td>
<td>0.65</td>
</tr>
<tr>
<td>eon</td>
<td>10,042,441.08</td>
<td>5.28</td>
<td>8,404,778.71</td>
<td>3.8</td>
</tr>
<tr>
<td>arpanet</td>
<td>270,932.52</td>
<td>1.39</td>
<td>420,804.94</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Firstly, we should mention that the challenge TM \(d_{max}\) was always in \(\hat{D}\) for CM and LM in all instances except example with LM. This observation obviates any concern about the relevance of this comparison. In terms of the routing cost \(\Phi\), we see that neither of the two outperforms in all cases. The difference is more clear for nsf, abovenet, eon and arpanet where LM is superior in the first three. In the remaining cases, the absolute value of the percent gaps between two methods are in the interval [0, 5.5%] where we calculate the gap as \(100 \times \frac{|\Phi_{CM} - \Phi_{LM}|}{min(\Phi_{CM}, \Phi_{LM})}\). In the overall, LM is superior in 4 instances whereas CM performs better in 3 cases. Next, we compare the congestion rates to assess the fairness of each routing. The two strategies perform equally well in 7 instances. Nevertheless CM routes \(d_{max}\) more fairly in bhvac, abovenet, and arpanet. LM appears to be slightly better in nsf.

We also compare our weight-managed OSPF routing with the unconstrained routing for the challenge
to have an idea about the effectiveness of our traffic engineering efforts. We solve (1)-(6) to determine the routing cost for the unconstrained routing ($\Phi_{UC}$). Then we calculate the cost coefficient $\rho$ of each strategy as $\rho_{CM} = \frac{\Phi_{CM}}{\Phi_{UC}}$ and $\rho_{LM} = \frac{\Phi_{LM}}{\Phi_{UC}}$. Table 4 displays the cost coefficients of the two strategies for the challenge $TM \ d^{max}$.

<table>
<thead>
<tr>
<th>Instance</th>
<th>$\rho_{CM}$</th>
<th>$\rho_{LM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>exodus</td>
<td>1.06</td>
<td>1.06</td>
</tr>
<tr>
<td>nsf</td>
<td>1.49</td>
<td>1.14</td>
</tr>
<tr>
<td>vnsl</td>
<td>1.03</td>
<td>1.03</td>
</tr>
<tr>
<td>example</td>
<td>2.88</td>
<td>2.94</td>
</tr>
<tr>
<td>metro</td>
<td>1.19</td>
<td>1.17</td>
</tr>
<tr>
<td>bhvac</td>
<td>1.45</td>
<td>1.45</td>
</tr>
<tr>
<td>abovevenet</td>
<td>1.49</td>
<td>1.56</td>
</tr>
<tr>
<td>telstra</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>pacbell</td>
<td>1.16</td>
<td>1.16</td>
</tr>
<tr>
<td>eon</td>
<td>3.33</td>
<td>2.79</td>
</tr>
<tr>
<td>arpanet</td>
<td>7.06</td>
<td>10.97</td>
</tr>
</tbody>
</table>

Table 4: OSPF versus Unconstrained routing.

Notice that the unconstrained routing problem is a relaxation of the OSPF routing problem and hence $\rho_{CM}$ and $\rho_{LM}$ can never be less than 1. Moreover, smaller values imply that we could make OSPF routing comparable to unconstrained routing through weight management. Table 4 shows that in 8 of the 11 instances, $\rho$ for both strategies are quite close to 1. This also supports our previous comments on the need for capacity expansion especially in exodus and bhvac. For these instances $\rho$ values are slightly over 1 and hence we have observed relatively larger $U^{max}$ rates for them is not due to the failure of OSPF routing but the insufficient capacity for some arcs. Hence, we can say that the current study provides a tool for network operators to assess the sufficiency of their current network resources. To conclude, we can say that we could make OSPF routing comparable to unconstrained routing by managing OSPF weights.

5 Conclusion

In this work, we studied the oblivious weight-managed OSPF routing problem for a general polyhedral demand uncertainty definition. We used the cost function of Fortz and Thorup [20] to determine the OSPF weight metric and hence the set of shortest paths such that the routing cost for the worst case in the demand polyhedron is minimum. We gave an MIP model for our weight-managed OSPF routing problem for a general definition of polyhedral demand uncertainty. Given the difficulty of the problem, we decided to focus on an algorithmic solution approach based on traffic matrix enumeration and tabu search. We generate an extreme point of the traffic polyhedron at each iteration of the algorithm using two different maximization problems and determine the best OSPF weight metric by a tabu search algorithm. Then we focused on the well-known hose model of demand uncertainty for our experimental tests and gave a comprehensive discussion of our results. We observed that we could make OSPF routing comparable to unconstrained routing by effective weight management for most of our test instances.

6 Acknowledgements

The authors have been partially supported by the Walloon Region (DGTRE) in the framework of the TOTEM Project, the Communauté française de Belgique - Actions de Recherche Concertées (ARC) and
FRFC program “Groupe de Programmation Mathmatique”. Hakan Ümit is supported by the FNRS - Bourse de Séjour Scientifique.

References


