Oblivious OSPF Routing with Weight Optimization under Polyhedral Demand Uncertainty[1]

Ayşegül Altın* Bernard Fortz⋆⋄ Hakan Ümit⋄

*Department of Industrial Engineering, TOBB University of Economics and Engineering
Sögütözü 06560 Ankara, Turkey

⋆ Department of Computer Science, Université Libre de Bruxelles
Boulevard du Triomphe, 1050, Brussels, Belgium

⋄IAG - Louvain School of Management and CORE, Université Catholique de Louvain
Louvain-la-Neuve, Belgium

Abstract

The desire for configuring well-managed OSPF routes to handle today's communication needs with larger networks and changing service requirements has opened the way to use traffic engineering tools with OSPF protocol. Moreover, anticipating possible shifts in expected traffic demands while using network resources efficiently has started to gain more attention. We consider these two crucial issues 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. We provide an algorithmic approach with two variations to tackle the problem. We present several test results and discuss whether we could make our weight-managed OSPF comparable to unconstrained routing.

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

1 Introduction

OSPF is an intra-domain routing protocol where traffic between routers are routed on shortest paths, which are uniquely determined by the link metric. Fixing link weights a priori gives no chance to traffic engineering with OSPF. Consequently, following Fortz and Thorup [4, 5], determining the link metric 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 [1, 6, 8, 9].

For a given network, the traditional routing problem deals with selecting paths of an arbitrary structure to route a 'given' set of demands. However, we consider polyhedral demands and discuss weight-managed OSPF routing, where the optimal routing is *oblivious* since it is determined irrespective of a specific demand matrix. We also apply the ECMP rule and hence for each node j on a shortest path between a source and sink pair, we use all shortest paths from j to the sink node such that each path carries an equal fraction of the corresponding demand passing through j. The current work offers new extensions in several dimensions. Firstly, Ben-Ameur and Kerivin [2] use polyhedral demands for the

[1]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.

general routing problem with no restriction on the path structure or how the flow is split among multiple paths. But, we study OSPF routing with ECMP for polyhedral demands. Mulyana and Killat [8] consider a rather restricted demand polyhedron where there can only be outbound or inbound constraints, but not both. However, we consider the general case to handle any demand polyhedron. Moreover, we provide an alternative approach to Altın et al. [1], where they use a quotient min-max regret performance measure based on link utilization and employ pure mathematical programming tools. In this work, we use the cost function of Fortz and Thorup [4] and extend their tabu search algorithm to handle polyhedral demands.

Let $G = (V, A, c)$ be a capacitated backbone graph with node set V, directed arc set A, and arc capacity vector $c > 0$. Let Q be the nonempty set of commodities. A traffic matrix (TM) $d \in \Re^{|Q|}$ keeps the demand information for each commodity in Q. We do not assume that the link weights or the demand matrix d are known a priori. We rather want to determine a link metric that would yield a shortest paths configuration, which is able to handle 'applicable' changes in demand estimates in the least costly and most fair way. We characterize such possible changes with a polyhedral set $\mathfrak{D} = \{d \in \Re^{|Q|} : Ad \leq \alpha\}$ and the routing cost (ϕ_{ij}) on each arc $(i, j) \in A$ is an increasing piecewise linear convex function of its utilization rate $\frac{l_{ij}}{c_{ij}}$ where l_{ij} is the total traffic load on $(i, j) \in A$ ([4, 5]). Basically, for the break point $z \in Z$ of the cost function, if $\frac{l_{ij}}{c_{ij}} \in [\underline{\rho}^z, \overline{\rho}^z)$, then $\phi_{ij} = u_z l_{ij} - v_z c_{ij}$ where u_z and v_z are the coefficients of the corresponding segment. The rest of the paper is organized as follows. We discuss our tabu search based algorithm in Section 2 and continue with some test results in Section 3. We conclude the paper with a summary of our study in Section 4.

2 A heuristic approach to oblivious OSPF routing

We use an algorithmic approach to tackle the OSPF routing problem for polyhedral demands, which has two main steps, namely the *TM* enumeration and weight optimization. For a given routing, the motivation for *TM* generation is to enumerate the extreme points of $\mathfrak{D}$ which correspond to the 'most challenging' traffic demands in terms of either the arc utilization or the routing cost. Since $\mathfrak{D}$ is a polyhedral set, the algorithm will terminate after a finite number of iterations. We show two different strategies based on *Cost Maximization* (*CM*) in Algorithm 1 and *Load Maximization* (*LM*) in Algorithm 2. In *CM* we search for a feasible *TM* that would increase the average routing cost for the current best OSPF routing scheme. On the other hand in *LM*, we search for a *TM*, which makes some arc of the network more congested. Hence, 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 where we use hashing functions to avoid generating multiple copies of the same *TM*. However, we can modify Algorithm 2 easily to generate multiple *TM*s. Finally, even though we put emphasis either on the congestion rate or routing cost in these two approaches, this does not mean that we focus on just one dimension and ignore the other since our routing cost is a function of utilization rates. We may also 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.

On the other hand, we use IGP-WO of TOTEM ([7]), for weight optimization. IGP-WO uses tabu search based heuristics to find an integer link weight setting and hence an OSPF routing scheme that avoids congestion by minimizing the cost function $\sum_{(i,j)\in A} \phi_{ij}$. It handles multiple *TM*s by minimizing the sum of the cost functions over the set of *TM*s enumerated so far at each iteration. Moreover, 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.

Algorithm 1 Strategy 1 with Cost Maximization - *CM*

Require: directed graph $G = (V, A, c)$, traffic polytope $\mathfrak{D}$, routing cost function Φ;
Ensure: minimum cost OSPF routing f^* and metric ω^* for $(G, \mathfrak{D}, \Phi)$;
INITIALIZE:
Find an initial feasible TM $d_0 \in \mathfrak{D}$;
$d^{rec} \leftarrow d_0$; // d^{rec} : the most recently enumerated TM;
$\tilde{D} \leftarrow d_0$; // $\tilde{D}$: current set of TMs enumerated so far;
$New_{TM} \leftarrow TRUE$, $cnt = 0$;
MAIN:
while $(cnt \leq cnt\_limit)$ and $(New_{TM} = TRUE)$ **do**
TABU: Find an optimized oblivious OSPF routing g^* for $\tilde{D}$ and the associated metric ω_T^*;
Get $\Phi_{\tilde{D}}$: the average routing cost for $\tilde{D}$;
$New_{TM} = FALSE$;
GENERATE TM: Determine the **most costly** *TM* d^{new} for g^* with the routing cost $\sum_{a \in A} \phi_a^*$;
if $\sum_{a \in A} \phi_a^* > \Phi_{\tilde{D}}$ and $d^{new} \notin \tilde{D}$ **then**
$\tilde{D} \leftarrow d^{new}$, $New_{TM} = TRUE$, $cnt \leftarrow cnt + 1$;
$f^* \leftarrow g^*$, $\omega^* \leftarrow \omega_T^*$;
CHALLENGE:
Find the challenge TM $d^{max} = \text{argmax}_{d \in \mathfrak{D}} \sum_{(s,t) \in Q} d_{st}$;
Get $\Phi^*_{d^{max}}$ and $U^*_{d^{max}}$// the routing cost and congestion rate for d^{max} with f^*;

Algorithm 2 Strategy 2 with Arc Load Maximization - *LM*

Require: directed graph $G = (V, A, c)$, traffic polytope $\mathfrak{D}$, routing cost function Φ;
Ensure: minimum cost OSPF routing f^* and metric ω^* for $(G, \mathfrak{D}, \Phi)$;
INITIALIZE // As in Algorithm 1
MAIN:
while $(cnt \leq cnt\_limit)$ and $(New_{TM} = TRUE)$ **do**
TABU: Find an optimized oblivious OSPF routing g^* for $\tilde{D}$ and the associated metric ω_T^*;
U^{max} = maximum link utilization for d^{rec};
$New_{TM} = FALSE$, $a = 0$ // start with the first arc of G;
while $(a < |A|)$ and $(New_{TM} = FALSE)$ **do**
GENERATE TM: $d^{new} = \text{argmax}_{d \in \mathfrak{D}}(g_a^* d)$; // d^{new} : worst case TM for a with routing g^*;
if $(g_a^* d^{new} > c_a)$ or $(\frac{g_a^* d^{new}}{c_a} > U^{max})$ **then**
if $d^{new} \notin \tilde{D}$ **then**
$d^{rec} = d^{new}$, $\tilde{D} \leftarrow d^{rec}$, $New_{TM} = TRUE$, $cnt \leftarrow cnt + 1$;
if $New_{TM} = FALSE$ **then**
$a \leftarrow a + 1$;
$f^* \leftarrow g^*$, $\omega^* \leftarrow \omega_T^*$;
CHALLENGE //As in Algorithm 1

3 Computational experiments

3.1 Hose Model

We consider the hose model ([3]), where each node is assigned an outgoing and incoming traffic bandwidth capacity. Then for $G = (V, A, c)$, we have $\mathfrak{D}_{hose} = \{d \geq 0 : \sum_{t \in W \setminus \{s\}} d_{st} \leq b_s^+; \sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s^- \ \forall s \in W\}$ 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.

3.2 Experimental Results

In this section, we provide test results for *CM* and *LM* under the hose model. We perform our tests on *bhvac, pacbell, eon, metro*, and *arpanet* from the IEEE literature as well as *exodus, abovenet, vnsl*, and *telstra* from the Rocketfuel project [10].

We implement the algorithms in C and use Cplex 11.0 to solve the maximization problems in *CM* and *LM*. We choose *cnt_limit* as 50. For *CM*, we had to reduce *cnt_limit* to 5 and 10 in *eon* and *arpanet* to avoid excessive solution times. We show our results in Table 1 and Table 2 where we give $|V|$ (number of nodes), $|A|$ (number of arcs), $|W|$ (number of terminals), $|\tilde{D}|$ (number of *TM*s enumerated throughout the algorithm), $\Phi_{\tilde{D}}$ (final average routing cost for $\tilde{D}$), Φ_F (routing cost for the final *TM*, d^{rec}, added to $\tilde{D}$), the normalized cost for d^{rec} ($\Phi_F^{norm} = \frac{\Phi_F}{\Phi_U}$ where Φ_U is the cost of routing d^{rec} if all arcs in A had a unit length and unlimited capacity), $U^{max} = \max_{a \in A, d \in \tilde{D}} \frac{l_a}{c_a}$ (the maximum utilization rate), and t (solution time). Finally, * indicates a termination due to *cnt_limit* in both tables.

Instance	$\lvert V \rvert$	$\lvert A \rvert$	$\lvert W \rvert$	$\lvert \tilde{D} \rvert$	$\Phi_{\tilde{D}}$	Φ_F	Φ_F^{norm}	U^{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
eon	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 model of demand uncertainty.

Although Φ_F^{norm} entries are relevant for the most recent *TM*, we see that large values of Φ_F^{norm} are accompanied by large U^{max} values and vice versa. Such high entries in Table 1 for *exodus, example, bhvac,* and *abovenet* suggest the existence of some bottleneck arcs for which capacity expansion is essential. Moreover, Φ_F^{norm} column shows that our traffic engineering efforts have improved the relative performance of the final routing significantly in *nsf, vnsl, metro, telstra,* and *pacbell.* However, *eon* and *arpanet* are difficult instances for *CM* since generating cost-maximizing *TM*s takes relatively longer for them.

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

We can make a comparison of *CM* and *LM* based on Table 1 and Table 2. Naturally, we had to enumerate more *TM*s in *LM* on the average as a result of the difference in the domain of impact for each enumeration. Basically, *LM* generates at least 'locally challenging' *TM*s since it considers arcs one by one. Moreover, as a natural consequence of the difference in their *TM* generation criteria, $\Phi_{\tilde{D}}$ is larger for

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert W\rvert$	$\lvert\tilde{D}\rvert$	$\Phi_{\tilde{D}}$	Φ_F	Φ_F^{norm}	U^{max}	t (sec)
exodus	7	12	7	2	841.07	837.66	178.09	4.53	1
nsf	8	20	5	3	1373.3	1219.33	0.77	0.96	3
vnsl	9	22	3	1	170,331.3	170,331.3	0.25	0.83	1
example	10	30	4	7	2236.8	85.33	1	1.25	23
metro	11	84	5	23	140.61	99.5	0.20	0.73	1029
bhvac	19	44	11	51	7,547,647.45*	3,385,313.33*	368.18*	50.7*	3084*
abovenet	19	68	5	12	254.43	106.85	46.8	3.19	440
telstra	44	88	7	1	0.26	0.26	0.13	0.88	96
pacbell	15	42	7	23	603.72	635	0.18	0.84	485
eon	19	74	15	51	1,068,540.68*	1,991,960.17*	110.63*	6.80*	10,500*
arpanet	24	100	10	45	73,835.82	185,369.83	13.69	1.5	18,331

Table 2: Results for *LM* under the hose demand uncertainty model.

CM in all but one instance whereas U^{max} is larger for *LM* in *metro, bhvac, abovenet,* and *eon.* However, both strategies achieve the same U^{max} values in the remaining 7 instances. Finally, the solution times indicate that *CM* is more efficient especially for smaller networks whereas it becomes less effective for more dense networks with higher number of commodities, i.e., $|W| * (|W| - 1)$, as in *eon* and *arpanet.*

In addition to these preliminary comments, in Table 3, we compare *CM* and *LM* on how good they route the challenge *TM* d^{max} enumerated in the *CHALLENGE* step of both strategies.

Instance	Φ_{CM}	U_{CM}^{max}	ρ_{CM}	Φ_{LM}	U_{LM}^{max}	ρ_{LM}
exodus	844.48	4.53	1.06	844.48	4.53	1.06
nsf	2656.70	0.88	1.49	2037.73	0.76	1.14
vnsl	170,331.3	0.83	1.03	170,331.3	0.83	1.03
example	522.83	1.1	2.88	533.83	1.1	2.94
metro	464.71	0.57	1.19	455.67	0.57	1.17
bhvac	24,166,769.6	36.3	1.45	24,273,340.6	36.95	1.45
abovenet	659.53	2.09	1.49	689.55	2.66	1.56
telstra	0.26	0.88	1	0.26	0.88	1
pacbell	2025	0.65	1.16	2025	0.65	1.16
eon	10,042,441.08	5.28	3.33	8,404,778.71	3.8	2.79
arpanet	270,932.52	1.39	7.06	420,804.94	1.47	10.97

Table 3: *CM* versus *LM* in the *CHALLENGE* step.

In terms of the routing cost Φ, 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 $100 * \frac{|\Phi_{CM} - \Phi_{LM}|}{\min\{\Phi_{CM}, \Phi_{LM}\}}$. 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 (*UR*) for the challenge *TM* d^{max} to judge the effectiveness of our traffic engineering efforts. We show the *cost coefficient* (ρ) measure, which is the ratio of the routing cost of each strategy to the routing cost of unconstrained routing for d^{max}, in Table 3. Since the *UR* problem is a relaxation of the OSPF routing problem, ρ_{CM} and ρ_{LM} can not be less than 1. Moreover, smaller values imply that we could make OSPF routing comparable to *UR* through weight management. Table 3 shows that in 8 of the 11 instances, ρ 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 ρ values are slightly over 1 and hence we have observed relatively larger U^{max} rates due to having insufficient capacity for some arcs rather than the failure to optimize our OSPF routing. 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.

4 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 [4] 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. 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. Our experimental tests with the hose model of demand uncertainty show that we can make OSPF routing comparable to unconstrained routing by effective weight management for most of our test instances.

References

[1] A. Altın, P. Belotti, and M.Ç. Pinar, *Ospf routing with optimal oblivious performance ratio under polyhedral demand uncertainty*, Technical report (2006) Bilkent University Industrial Enginering Department.

[2] W. Ben-Ameur and H. Kerivin, *Routing of uncertain demands*, Optimization and Engineering **3** (2005) pp.283–313.

[3] J. A. Fingerhut, S. Suri, and J. S. Turner, *Designing least-cost nonblocking broadband networks*, Journal of Algorithms **24(2)** (1997) pp. 287–309.

[4] B. Fortz and M. Thorup, *Internet traffic engineering by optimizing OSPF weights*, In Proc. 19th IEEE Conf. on Computer Communications (INFOCOM) (2000) pp. 519–528.

[5] B. Fortz and M. Thorup, *Increasing internet capacity using local search*, Computational Optimization and Applications **29(1)** (2004) pp. 13–48.

[6] B. Fortz and H. Ümit, *Efficient techniques and tools for intra-domain traffic engineering*, Technical Report 583 (2007), ULB Computer Science Department.

[7] G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D'Arienzo, O. Delcourt, J. Domingo-Pascual, S Cerav-Erbas, I. Gojmerac, X. Masip, A. Pescaph, B. Quoitin, S.F. Romano, E. Salvatori, F. Skivée, H.T. Tran, S. Uhlig, and H. Ümit, *An open source traffic engineering toolbox*, Computer Communications **29(5)** (2006) pp. 593–610.

[8] E. Mulyana and U Killat, *Optimizing IP networks for uncertain demands using outbound traffic constraints*, In Proc. INOC 2005 (2005) pp. 695–701.

[9] M. Pióro, A. Szentesi, J. Harmatos, and A. Jüttner, *On OSPF related network optimization problems*, In 8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks **70** (2000), pp. 1–14.

[10] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, *Measuring ISP topologies with rocketfuel*, IEEE/ACM Trans. Netw. **12(1)** (2004) pp. 2–16.