Dynamic IGP Weight Optimization in IP Networks

Olivier Brun and Jean-Marie Garcia
CNRS; LAAS; 7, avenue du Colonel Roche, F-31077 Toulouse, France.
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France.
{brun, jmg}@laas.fr

***Abstract*—With the explosive growth of the Internet and the incredible development of network applications, the variation in traffic volume has become one of the most important problems faced by network operators. Designing a network using a single "busy hour" traffic matrix strains credibility due to the high volatility of traffic patterns. Thus, there is a need for efficient dynamic reconfiguration methods allowing to adapt resource utilization to prevailing traffic. In this paper, we focus on the problem of link weight optimization in IP networks where the traffic is routed along shortest paths according to the link metrics (OSPF and IS-IS-based networks). We propose an online approach to handle time-varying traffic matrices that relies on online traffic monitoring and updates link weights, and thus the routing paths, adaptively as some changes are observed.**

***Index Terms*—IGP routing protocols, SNMP link counts, demand uncertainty, dynamic IGP weight optimization.**

I. Introduction

The success of the Internet – well beyond the initial expectations of its inceptors – has resulted over time in its extensive usage for societal and economical reasons. The Internet is today the fundamental component of the worldwide communication infrastructure playing a crucial role in our society. The success of the Internet is yet to be amplified in the near future with the rapid development of many new applications, including grid and cloud computing applications, but also peer-to-peer and social networks.

Network operators have to regularly upgrade their network infrastructure in order to face the growth of the Internet traffic and the rising QoS requirements. In the current competitive context, a cost-efficient alternative to installing excessive amounts of capacity is to perform route optimization in order to avoid network congestion and service degradation. Route optimisation allows to make more efficient use of existing network resources by tailoring routes to prevailing traffic.

Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) are the most commonly used intra-domain Internet routing protocols (i.e. Interior Gateway Protocol) [37], [35]. Traffic flow is routed along shortest paths, splitting flow at nodes where several outgoing links are on shortest paths to the destination. The weights of the links, and thereby the shortest path routes, can be changed by the network operator. The weights could be set to 1 or proportional to their physical distances, but the standard practice recommended by Cisco is to make the weight of a link inversely proportional to its capacity. Although fairly efficient in some cases, such heuristics often lead to poor network resource utilization.

Given a set of traffic demands between origin/destination (OD) pairs, the link weight optimization problem amounts to finding a set of link weights that optimize a given performance measure. This is in contrast with the traditional routing problem where there is no restriction on the structure of the paths to be used. Since its introduction by Fortz and Thorup [23], [24], many studies have been devoted to the link weight optimization problem.

In the above definition of the link weight optimization problem, it was assumed that the traffic load in the network is known or can be measured. However, in practice, traffic demands between nodes cannot be measured exactly, and moreover they change continuously. Designing a network using a single "busy hour" traffic matrix strains credibility because with the increasing popularity of higher bandwidth applications, traffic patterns are more and more volatile even in the aggregate. Such an approach can lead to a poor utilization of network resources if at some point in time the actual traffic matrix deviates significantly from the one used for route optimization.

An offline approach that has been recently proposed is to optimize over a set of traffic matrices [36], [3], [15], [5]. This set, which is often a polyhedral set, has to contain all possible realizations of the random traffic demands, or at least the most likely ones. 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 traffic matrix. Optimization methods used to obtain this oblivious routing belong to the class of robust optimization methods since they integrate the uncertainty about the time-varying traffic demands.

In this paper, we propose an alternative online approach to handle time-varying traffic matrices. This approach relies on online traffic monitoring using SNMP link counts and updates routes adaptively as some changes are observed. To the extend of our knowledge, this is the first time an online algorithm is proposed for OSPF weight optimization. One of the main problem that one has to solve for online IP route optimization is related to the uncertainty on traffic demands: the link counts provided by the SNMP protocol do not allow to retrieve the actual traffic demands. Our algorithm adapts the local search method proposed in [21] to handle demand uncertainty by optimizing over the set of all traffic matrices that are compatible with the observed link counts. Extensive computational results show that the proposed dynamic OSPF configuration algorithm outperforms static configurations and delivers solutions which are often very close to that obtained assuming that the actual traffic matrix can be known.

In the next section, we review related works. Section III is devoted to the mathematical statement of the link weight setting problem and describes how it can be solved with a local search method. The proposed online algorithm is described in section IV. Computational results are reported in section V. Concluding remarks are drawn in the last section.

II. Related Works

In this section we review the relevant research literature on the traditional link weight optimization problem, on traffic matrix estimation and on robust routing optimization.

A. Link weight optimization with known traffic demands

The traditional link weight optimization problem assumes that the traffic demand is known for each OD pair and aims at finding the link weights, and hence the routing paths, that optimize some given criterion such as the link utilization (see [4] for a complete survey and problem extensions). This problem is known to be NP-hard [23]. Although it can be formulated as an integer programming problem [41], this formulation does not yield an efficient approach to solve it. Therefore, most previous works have focused on heuristics. They can be classified into two broad approaches: those that tries to solve an inverse shortest path problem [8], [10], [11], [13], [43], [48], and those that start from an initial solution, i.e. a set of link weights, and aim at iteratively improving it [23], [19], [12], [25], [51], [21], [49]. In this paper, we extend the local search alogrithm of [21] to our special structure of demand uncertainty.

B. Traffic matrix estimation

As mentioned in the introduction, traditional route optimization methods assume that the traffic demand of each OD pair in the network is known. Unfortunately, it cannot be directly measured in large high-speed networks due to the high processing overhead and to the significant reporting traffic induced by current metrology tools (e.g. Netflow [16]). Some attemps have been done to reduce this overhead using sampling techniques [17]. An alternative approach is to use the link counts provided by the SNMP protocol to retrieve the actual traffic demands. However, since there are usually many more network flows than link load measures, this leads to an ill-posed inverse problem which cannot be solved without additional information. Several generations of methods have been proposed to solve this problem [50], [34], [14], [30], [42], [52], [45], [38], [22], [46], [39]. The first generation methods were not enough accurate for an efficient traffic engineering. Although the accuracy of the most recent methods is signifiantly superior to that of first generations methods, they rely on Netflow measurements during a calibration phase and thus induce a significant network overhead.

In this paper, we dot not try to infer the actual traffic matrix of the network from link loads. Our original approach is to determine the set of all traffic matrices that are compatible with the measured link loads, and to optimize the link weights over this set.

C. Robust routing optimization

One of the most well-known model of demand uncertainty is probably the hose-model [18] introduced by Duffield et al. for VPN design [2], [29], [32], [33], [44], [20]. This model gives the total amount of ingress and egress traffic at each endpoint of the VPN. The uncertainty set is then composed of all traffic matrices such that their row sum is the total ingress bandwidth of the corresponding source node and their column sum is the total egress bandwidth of the corresponding destination node.

Some authors have addressed the general routing problem. Applegate and Cohen [6] discuss this problem with almost no information on traffic demands. Belotti and Pinar [7] incorporate box model of uncertainty as well as statistical uncertainty into the same problem. Ben-Ameur and Kerivin [9] study the minimum cost general oblivious routing problem under polyhedral demand uncertainty and propose an algorithm based on iterative path and constraint generation as a solution procedure.

The robust link weight optimization problem has been considered only recently. Chu and Lea address this problem for IP networks supporting hose-model VPN [15]. Mulyana and Killat [36] deal with the OSPF routing problem, where traffic uncertainty is described by a set of outbound constraints. Altin et al. [3] study polyhedral demand uncertainty with OSPF routing under weight management and provide a compact MIP formulation and a branch-and-price algorithm which can be used to solve small problem instances exactly. In [5], Altin et al. extend the tabu search algorithm of [23] to handle polyhedral demand uncertainty.

Whereas the above mentionned works were focused on offline optimization, our algorithm has been designed for dynamic reconfiguration of IP routes according to observed link loads.

III. Robust OSPF Weight Optimization

Let us assume that we are given a network of N nodes and M links represented by a graph $G = (V, E)$. We let $Q \subset V$ denote the set of edge routers. The capacity of link $l \in E$ is C_l.

A. Demand uncertainty set

We are also given a set of $K = |Q|(|Q| - 1)/2$ OD flows, where each flow $k = 1, \ldots, K$ is defined by its source node $s_k \in Q$, its destination node $t_k \in Q$ and its bandwidth requirement d_k. In the following, we let $\mathbf{d}$ denote the vector of traffic demands. This vector is unknown, but it is assumed to belong to a certain uncertainty set $\mathcal{D}$ of traffic matrices.

B. Feasible solutions

A feasible solution of the weight setting problem is a vector $\mathbf{w} = (w_1, \dots, w_M)$, where $w_l \in \Omega$ is the weight assigned to arc $l \in E$ and $\Omega = \{1, \dots, w_{max}\}$ is the set of allowed integer values for link weights. The OSPF protocol allows for $w_{max} \leq 2^{16}-1$. Given a feasible solution $\mathbf{w} = (w_1, \dots, w_M)$, a shortest path algorithm can be used to compute the following parameters:

- $D_u^t(\mathbf{w})$ is the distance from node u to destination node t,
- $\delta_{u,v}^t(\mathbf{w}) = 1$ if link $l = (u, v)$ is on a shortest path towards destination node t, i.e. $D_u^t(\mathbf{w}) = w_l + D_v^t(\mathbf{w})$, and 0 otherwise,
- $n_u^t(\mathbf{w}) = \sum_v \delta_{u,v}^t(\mathbf{w})$ is the number of outgoing arcs from node u that are on a shortest path towards destination node t.

These parameters summarize all the information on shortest paths from node u to node t for the weight vector $\mathbf{w}$. Assuming that traffic is split evenly between all outgoing links on the shortest paths towards the destination, the fraction of the OD flow (s,t) passing through link $l = (u, v)$ is then

$$f_{s,t}^l(\mathbf{w}) = \frac{\delta_{u,v}^t(\mathbf{w})}{n_u^t(\mathbf{w})} \sum_{(i,u)\in E} f_{s,t}^{i,u}. \tag{1}$$

Note that the above relation holds whatever the traffic matrix $\mathbf{d} \in \mathcal{D}$. It allows the recursive computation of the fractions $f_{s,t}^{u,v}(\mathbf{w})$ for each OD flow (s,t) and each link $(u, v) \in E$ once the parameters $\delta_{u,v}^t(\mathbf{w})$ and $n_u^t(\mathbf{w})$ have been computed.

C. Objective function

Assume a weight vector $\mathbf{w} \in \Omega^M$ has been fixed. Then, the maximum amount of traffic over link $l = (u, v)$ is given by

$$y_l(\mathbf{w}) = \max_{\mathbf{d}\in\mathcal{D}} \sum_{s,t\in Q} f_{s,t}^l \, d_{s,t}. \tag{2}$$

Note that if, as will be the case in Section IV, the demand uncertainty set $\mathcal{D}$ is a polytope specified by linear equalities and inequalities, then problem (2) is just a Linear Programming (LP) problem that can be easily solved with any LP solver.

Our objective is to minimize the congestion rate of the network, i.e., the utilization rate of the most loaded link. This is a standard cost function that is often used by network operators. The cost of a feasible solution $\mathbf{w} \in \Omega^M$ is therefore as follows:

$$\Phi(\mathbf{w}) = \max_{l\in E} \frac{y_l(\mathbf{w})}{C_l}. \tag{3}$$

The problem to be solved can then be stated as follows:

$$\min_{\mathbf{w}\in\Omega^M} \Phi(\mathbf{w}). \tag{4}$$

Note that our algorithm is not restricted to this cost function but can also be used for additive cost functions, e.g., to minimize the average queueing delay in the network.

D. A local search algorithm for weight optimization

To solve the weight optimization problem, we propose a local search heuristic [1], [26], [27]. The pseudo-code in figure 1 describes the proposed local search algorithm in detail.

Algorithm 1 Robust OSPF weight optimization algorithm

```
procedure METRICOPTIMISATION
    Let w = (w_1, ..., w_M) be the initial solution and
        Φ(w) its cost.
    w* = w        ▷ Initialisation of minimum cost solution

    while Convergence() = false do
        Φ_min = ∞
        for l = 1 ... M do
            Compute Δ_l and w^l = w + Δ_l e_l
            Compute shortest paths: δ^t_{u,v}(w^l), n^t_u(w^l)
            Compute maximum load y_l(w), l ∈ E
            Compute cost Φ(w^l)
            if Φ(w^l) ≤ Φ_min then
                w_next = w^l and Φ_min = Φ(w^l)
            end if
        end for
        w = w_next
        if Φ(w) < Φ(w*) then
            w* = w   ▷ Update of minimum cost solution
        end if
    end while
end procedure
```

We describe below the details of this algorithm.

1) Neighbourhood definition: Let $\mathbf{e}_l$ be the M-vector $(0, \dots, 1, \dots, 0)$ with 1 in position l and 0 elsewhere. The neighbourhood of a solution $\mathbf{w} = (w_1, \dots, w_M)$ is defined as follows:

$$\mathcal{N}(\mathbf{w}) = \left\{\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^M\right\}, \tag{5}$$

where $\mathbf{w}^l = \mathbf{w} + \Delta_l \, \mathbf{e}_l$, for $l \in E$, with

$$\Delta_l = argmin_{\Delta\geq 1}\left[\sum_{s,t\in Q} f_{s,t}^l(\mathbf{w} + \Delta\,\mathbf{e}_l) < \sum_{s,t\in Q} f_{s,t}^l(\mathbf{w})\right]. \tag{6}$$

Thus, the neighbourhood of a given solution W contains at most M solutions. Each one is associated to a link l and is obtained by increasing the weight of this link by the minimum amount such that at least one OD flow is deviated from this link (in whole or in part).

2) Neighbourhood generation: Let $\mathcal{F}_l$ be the set of OD flows transmitted over link l ($f_{s,t}^l > 0$). If $\mathcal{F}_l = \emptyset$, then Δ_l cannot be defined since it is impossible to deviate traffic from link l. Otherwise, we proceed as follows to compute Δ_l:

(a) Set $\overline{\mathbf{w}^l} = (w_1, \dots, w_{i-1}, \infty, w_{i+1}, \dots, w_M)$.

(b) For all $u, t \in V$, update distances $D_u^t(\overline{\mathbf{w}^l})$ using a shortest path algorithm.

(c) Let

$$d^l_{\min} = \min_{f\in\mathcal{F}_l} \left[D^{t(f)}_{s(f)}(\overline{\mathbf{w}^l}) - D^{t(f)}_{s(f)}(\mathbf{w}) \right], \qquad (7)$$

be the minimum increase of shortest path length over all flow $f \in \mathcal{F}_l$.

(d) Δ_l is given by

$$\Delta_l = \begin{cases} 1 & \text{if } d^l_{\min} = 0 \\ d^l_{\min} & \text{if } 0 < d^l_{\min} < \infty \\ \infty & \text{if } d^l_{\min} = \infty \end{cases}$$

The solution $\mathbf{w}^l = \mathbf{w} + \Delta_l\, \mathbf{e}_l$ deviates traffic from link l:

- If $d^l_{\min} = 0$, there was multiple shortest paths for at least one OD flow of $\mathcal{F}_l$ (load sharing). By increasing the weight w_l by $\Delta_l = 1$, this flow will be completely deviated from link l,
- If $0 < d^l_{\min} < \infty$, the solution $\mathbf{w}^l$ introduces load-sharing for at least one OD flow belonging to $\mathcal{F}_l$,
- If $d^l_{\min} = \infty$, link l is mandatory for all flows $f \in \mathcal{F}_l$ and it is easy to prove that it can be suppressed from the set of links whose weight has to be optimized (provided the network is connected).

3) Convergence test: The convergence test is based on 3 criteria: (1) maximum number of iterations Q_1, (2) a maximum number of iterations Q_2 before improving the best found solution and (3) an empty neighbourhood for the current solution ($\Delta_l = \infty$, $\forall l \in E$). If $Q_2 = 0$, the algorithm converges to a local minimum, but otherwise it can escape from a local minimum by selecting the neighbour solution corresponding to the minimum cost increase. The algorithm then stops if no solution better than $\mathbf{w}^*$ is found in Q_2 iterations.

IV. Dynamic Optimization of OSPF Weights

In this section, we describe the proposed dynamic routing approach to handle time-varying traffic matrices. As described in Figure 1, this approach relies on online traffic monitoring at regularly spaced time epochs (e.g. every 5 minutes). A central routing agent collects SNMP link counts to infer the demand uncertainty set $\mathcal{D}(\tau)$ at the measurement epoch τ. The routing agent then runs the robust weight optimization algorithm described in Section III-D with the current demand uncertainty set as input and determine wether OSPF weights can be changed so as to reduce the maximum utilization rate of network interfaces. If there is no applicable change, the routing agent falls asleep until the next measurement epoch $\tau+1$. Otherwise, it updates the OSPF weights and waits until the next measurement epoch.

A. SNMP data

The trafic demands between the edge routers evolve in time according to an unknown discrete-time stochastic process. However, we can have indirect observations on these traffic demands through SNMP link counts. More precisely, it is

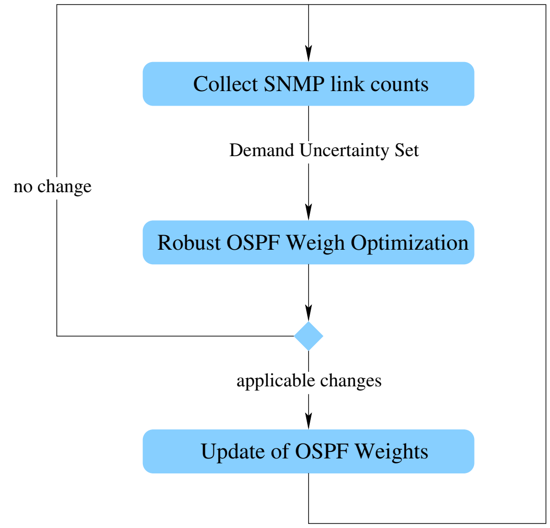


Fig. 1: Online optimization of OSPF weights using SNMP link counts

assumed that at time τ, the routing agent can collect a measure $\widehat{y}_e(\tau)$ of the average traffic on link e between times $\tau - 1$ and τ. We show below how these measures can be used to define a polytope $\mathcal{D}(\tau)$ that contains all the traffic matrices that are consistant with the observations.

In the following, it is also assumed that the total ingress and egress traffics of each edge router can be measured by monitoring the incoming and outgoing interfaces of the edge routers with SNMP. We let $b^{out}_s(\tau)$ and $b^{in}_s(\tau)$ denote the total ingress and egress traffics at time τ of edge router $s \in Q$, respectively.

B. The initial uncertainty set

Let $f^e_{s,t}(0)$ denote the fraction of the OD flow (s,t) passing through link e at time 0, i.e. for the initial weight vector $\mathbf{w}(0)$. The initial uncertainty set $\mathcal{D}(0)$ is defined as the polytope of traffic matrices $(d_{s,t}(0))_{s,t\in Q}$ satisfying the following linear inequalities:

$$\sum_{s,t\in Q} f^e_{s,t}(0) d_{s,t}(0) = \widehat{y}_e(0) \quad , \ e \in E, \qquad (8)$$

$$\sum_{t\neq s} d_{s,t}(0) = b^{out}_s(0) \quad , \ s \in Q, \qquad (9)$$

$$\sum_{s\neq t} d_{s,t}(0) = b^{in}_t(0) \quad , \ t \in Q, \qquad (10)$$

$$d_{s,t}(0) \geq 0 \quad , \ s,t \in Q. \qquad (11)$$

Constraints (8) express that the demands $d_{s,t}(0)$ have to be consistent with the observed link loads $\widehat{y}_e(0)$ for every links $e \in E$, while constraints (9) and (10) express that the trafic demands have to be consistent with the observed total egress and ingess traffics of every edge routers $s \in Q$, respectively. Constraints (11) ensure that the traffic demands are non-negative.

This initial uncertainty set is used to solve the OSPF weight optimization problem as explained in Section III. If no optimization is applicable, then the routing agent waits until measurement epoch 1 and the weight vector at that time is $\mathbf{w}^1 = \mathbf{w}^0$. Otherwise, the OSPF weights are updated and at time 1 the weight vector $\mathbf{w}^1$ is the optimal solution found.

C. The uncertainty set at time τ

As in the initial case, the traffic demands at time $\tau > 0$ have to be consistent with the observed link loads, and we thus get the following linear equalities and inequalities:

$$\sum_{s,t \in Q} f^e_{s,t}(\tau) d_{s,t}(\tau) = \widehat{y}_e(\tau) \quad , \; e \in E, \tag{12}$$

$$\sum_{t \neq s} d_{s,t}(\tau) = b^{out}_s(\tau) \quad , \; s \in Q, \tag{13}$$

$$\sum_{s \neq t} d_{s,t}(\tau) = b^{in}_t(\tau) \quad , \; t \in Q, \tag{14}$$

$$d_{s,t}(\tau) \geq 0 \quad , \; s, t \in Q. \tag{15}$$

Moreover, a significant reduction of the size of the uncertainty set can be obtained by correlating the traffic demand at time τ with past observations. In both [28], [40], the authors independently found from measurements on real data that the so-called fanouts, that is the ratios $d_{s,t}(\tau)/b^{out}_s(\tau)$, have a remarkable stability with respect to the traffic demand themselves. In the following, we shall assume that the fanouts have a slow variation.

Assumption 1: Let $T \geq 1$ be a time horizon. There exists parameters α and β such that $0 < \alpha \leq \beta$ and that the following inequality holds

$$\alpha \, \frac{d_{s,t}(\tau_k)}{b^{out}_s(\tau_k)} \leq \frac{d_{s,t}(\tau)}{b^{out}_s(\tau)} \leq \beta \, \frac{d_{s,t}(\tau_k)}{b^{out}_s(\tau_k)}, \tag{16}$$

for all edge routers $s, t \in Q$, $s \neq t$, and at all time instants $\tau_k = \tau - k \geq 0$, $k = 1, \ldots, T$.

For instance, the case of constant fanouts is obtained with $\alpha = \beta = 1$. The above assumption allows to relate the traffic demands at time τ and the observations at any other time τ_k, $k = 1, \ldots, T$. We thus easily obtain the following linear inequalities.

$$\alpha \, \widehat{y}_e(\tau_k) \leq \sum_{s,t \in Q} f^e_{s,t}(\tau_k) \frac{b^{out}_s(\tau_k)}{b^{out}_s(\tau)} d_{s,t}(\tau) \leq \beta \, \widehat{y}_e(\tau_k) \tag{17}$$

$$\alpha \, b^{in}_t(\tau_k) \leq \sum_{s,t \in Q} \frac{b^{out}_s(\tau_k)}{b^{out}_s(\tau)} d_{s,t}(\tau) \leq \beta \, b^{in}_t(\tau_k) \tag{18}$$

for all $e \in E, t \in Q$ and $k = 1, \ldots, T$. The demand uncertainty set at time $\tau > 0$ is thus obtained as the polytope of traffic vectors $\left(d_{s,t}(\tau)\right)_{s,t \in Q}$ satisfying the linear systems (12)-(18).

Network	Number of nodes	Number of links
ABOVENET	19	68
ARPANET	24	100
METRO	11	84
PACKBELL	15	42

TABLE I: Main characteristics of the test networks.

V. Results

We have tested our dynamic OSPF weight optimization algorithm by simulating it on 4 real network topologies (see Table I). We have collected the information of most instances from the IEEE literature (*pacbell, metro, and arpanet*), and from the Rocketfuel project [47] for the topology *abovenet*. All tests have been carried out with a Pentium Centrino 3 Ghz processor, running under Linux with 2 GB of memory available, and using CPLEX [31] as LP solver. In our simulation, every iteration correspond to five minutes, while the routing agent was sleeping as we described above. We have simulated our algorithm for twelve iterations (plus one initial iteration), which corresponds to one hour. To simulate the traffic changes, in each iteration we modified the traffic matrix with a random value, using a normal distribution, i.e., $d_{s,t}(\tau) = d_{s,t}(\tau - 1) + v_{s,t}(\tau)$ with $v_{s,t}(\tau) \sim \mathcal{N}(0, \sigma^2)$.

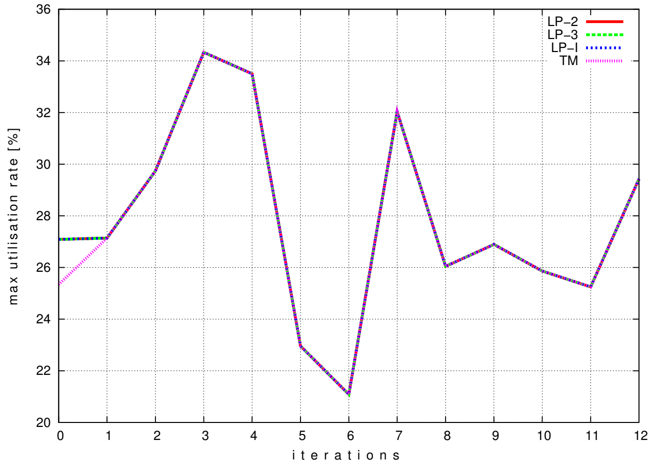
A. Constant fanouts

During the first series of simulations, we assumed that the fanouts are constants, i.e. the parameters α and β are both equal to 1. Results are reported on Figure 2. We plot the evolution of the maximum utilization rate obtained by our online algorithm for $T = \infty$ (LP-I, dashed blue line), for $T = 2$ (LP-2, continous red line) and for $T = 3$ (LP-3, dashed green line). We compare these results against the evolution in the ideal case, when the traffic matrix is known (TM, dashed purple line). Firstly, it can be seen that the algorithm with $T = \infty$ is always close to the ideal case, even for the initial simulation step (2.65% on the average), and quicly converges to the values obtained when we know the traffic matrix.

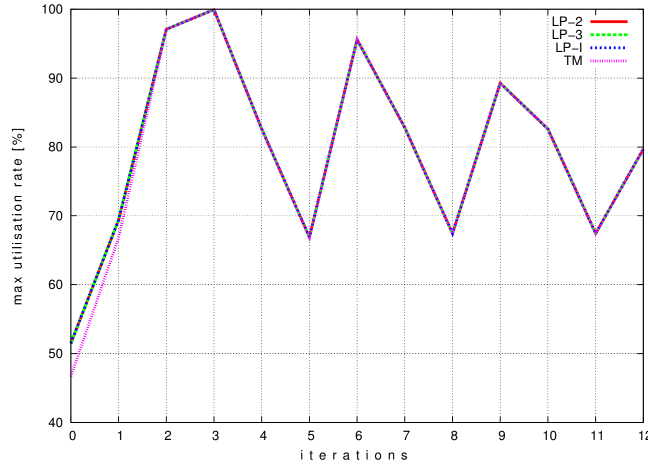
We also observe that integrating the observations of the last 3 time steps is sufficient for constant fanouts. This is important because the more constraints we add to our linear program to determine the values of the link loads, the longer the computation time becomes, and it may not be finished, when the routing agent awakes to begin the new computations. Also, these constraints must be stored in the routing agent memory, which may require significant space.

B. Variable fanouts

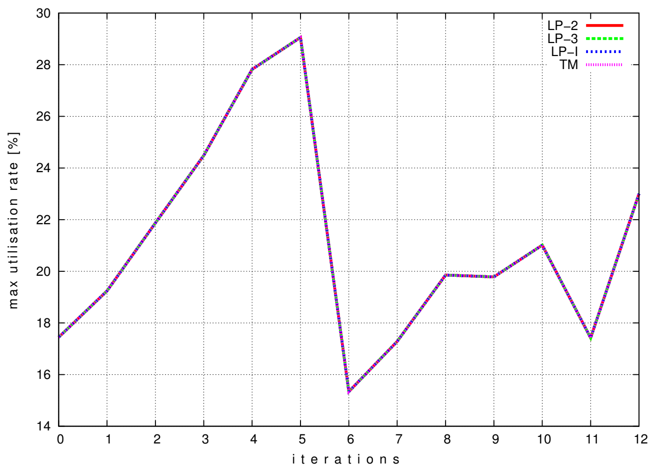
In this section, we assume that only the average of the fanouts is constant within a given time interval (in our simulations we choose the time interval to be 3 time steps, that is 15 minutes) and the value of the fanouts varies around the average by -/+ 5 %. Maximum utilization rates obtained with the online algorithm are reported on Figure 3 for $T = 3$, and are compared with the ideal case. From these results, we can see that for the *metro* network our algorithm is still very accurate



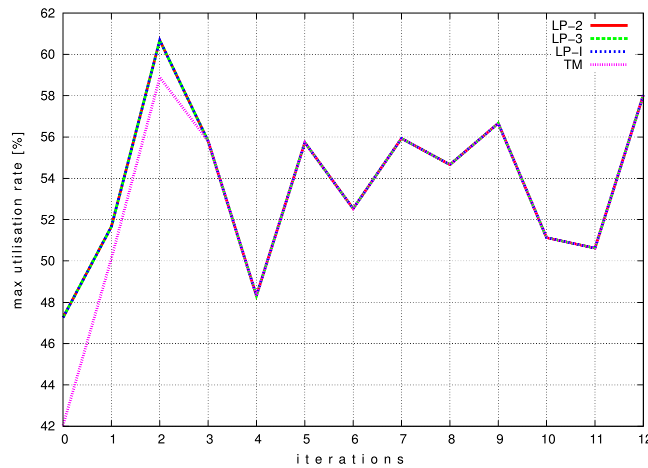
(a) abovenet



(b) arpanet



(c) metro



(d) pacbell

Fig. 2: Evolution of the maximum utilization rate with constant fanouts.

and it gives the same results as the ideal case. However, with most instances the gap in the maximum utilization rate become bigger.

Table II presents the running times of the algorithm at each time instant for every networks. We can see that our algorithm does not scale well for large networks, even if computation times are sufficiently low for a small network like *pacbell*.

VI. Conclusion

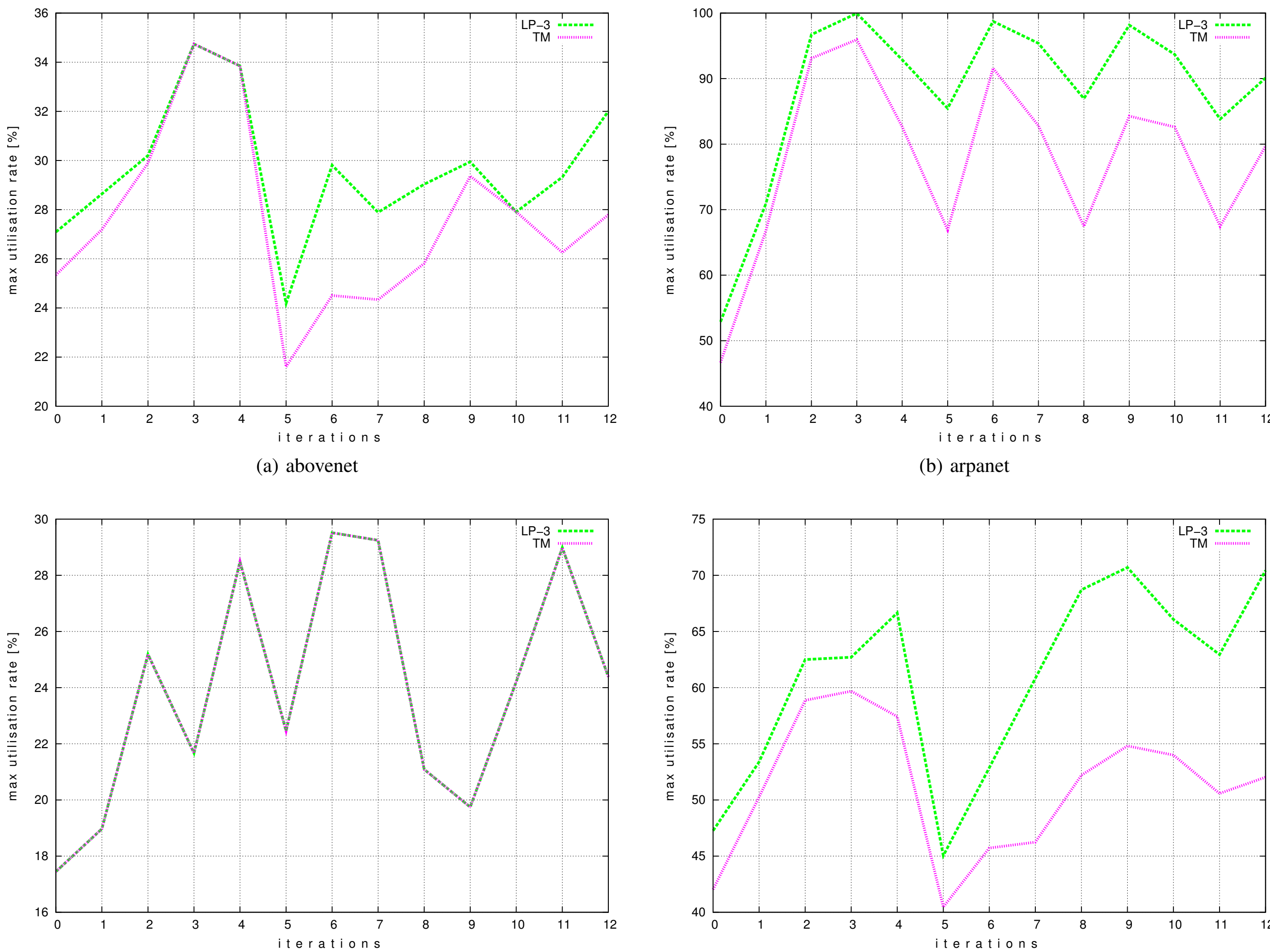
In this paper, we have addressed the problem of dynamic IGP weight optimization in IP networks using SNMP link counts. Assuming that the fanouts have a slow variation, we have shown that the proposed online algorithm performs almost as well as if the traffic matrix were known. However, the use of linear programming techniques gives rise to significant running times that do not allow to solve the problem in real-time for large networks. Future research will therefore tries to improve running times without compromising the quality of the solutions.

TABLE II: Running times of the algorithms with variable fanouts (in seconds)

	abovenet	arpanet	metro	pacbell
0	244	742	237	46
1	364	1408	633	79
2	723	1026	850	92
3	892	1560	616	107
4	921	1815	1401	111
5	1072	1719	1100	162
6	1227	1754	913	121
7	1395	1839	779	114
8	884	1919	4371	111
9	899	1876	1307	139
10	1047	2066	1956	142
11	901	2013	1396	196
12	1218	2038	1366	116

References

[1] E. Aarts and J. K.Lenstra. *Local Search in Combinatorial Optimization*. John Wiley & Sons Ltd., 1997.



(a) abovenet

(b) arpanet

(c) metro

(d) pacbell

Fig. 3: Evolution of the maximum utilization rate with variable fanouts.

[2] A. Altin, E. Amaldi, P. Belotti, and M. Pinar. Provisioning virtual private networks under traffic uncertainty. *Networks*, 20(1):100–115, January 2007.

[3] A. Altin, P. Belotti, and M. Pinar. Ospf routing with optimal oblivious performance ratio under polyhedral demand uncertainty. *Optimization and Engineering*, 2009.

[4] A. Altin, B. Fortz, M. Thorup, and H. Ümit. Intra-domain traffic engineering with shortest path routing protocols. *4OR: A Quarterly Journal of Operations Research*, 7(4):301–335, November 2009.

[5] A. Altin, B. Fortz, and H. Ümit. Oblivious ospf routing with weight optimization under polyhedral demand uncertainty. In *International Network Optimization Conference (INOC 2009)*, Pisa, Italy, April 26-29 2009.

[6] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fondamental tradeoffs. In *2003 conference on applications, technologies, architectures and protocols for computer communications (SIGCOMM'03)*, pages 313–324, New York, USA, 2003. ACM.

[7] P. Belotti and M. Pinar. Optimal oblivious routing under statistical uncertainty. *Optimization and Engineering*, 9(3):257–271, 2008.

[8] W. Ben-Ameur, E. Gourdin, and B. Liau. Dimensioning of internet networks. In *Proceedings of the DRCS'2000*, Munich, 2000.

[9] W. Ben-Ameur and H. Kerivin. Routing of uncertain demands. *Optimization and Engineering*, 3:283–313, 2005.

[10] W. Ben-Ameur and B. Liau. Calcul des mtriques de routage pour internet. *Ann. Telecommun 56*, pages 3–4, 2001.

[11] W. Ben-Ameur, N. Michel, and B. Liau. Routing strategies for ip networks.

[12] L. S. Buriol, M. Resende, C. Ribeiro, and M. Thorup. A hybrid genetic algorithm for the weight setting problem in ospf/is-is routing. *Networks*, 46(1):36–56, August 2005.

[13] D. Burton. *On the inverse shortest path problem*. PhD thesis, MIT, 1993.

[14] J. Cao, D. Davis, S. Wiel, and B. Yu. Time-varying network tomography : Router link data, 2000.

[15] J. Chu and C. T. Lea. Optimal link weights for ip-based networks supporting hose-model vpns. *IEEE/ACM Transactions on Networking*, 17(3):778–786, June 2009.

[16] Cisco Systems. Cisco ios netflow. http://www.cisco.com/en/us/products/ps6601/products_ios_protocol_group_ home.html.

[17] Cisco Systems. Sampled netflow. http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_sanf.html.

[18] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive. A flexible model for resource management in virtual private networks. In *ACM SIGCOMM*, pages 95–108, San Diego, CA, USA, August 1999.

[19] M. Ericsson, M. Resende, and P. Pardalos. A genetic algorithm for

the weight setting problem in ospf routing. *Journal of Combinatorial Optimization*, 6:299–333, 2002.
[20] T. Erlebach and M. Rüegg. Optimal bandwidth reservation in hose-model vpns with multi-path routing. In *INFOCOM*, 2004.
[21] C. Fortuny, O. Brun, and J. M. Garcia. Metric optimization in ip networks. In *19th International Teletraffic Congress*, pages 1225–1234, Beijing, China, 2005.
[22] C. Fortuny, O. Brun, and J. M. Garcia. Fanout inference from link counts. In *4th European Conference on Universal Multiservice Networks (ECUMN 2007)*, pages 190–199, 2007.
[23] B. Fortz and M. Thorup. Internet traffic engineering by optimizing ospf weights. In *Proc. 19th IEEE Conf. on Computer Communications (INFOCOM)*, 2000.
[24] B. Fortz and M. Thorup. Increasing internet capacity using local search. *Computational Optimization and Applications*, 29:13–48, 2004.
[25] B. Fortz and H. Ümit. Efficient techniques and tools for intra-domain traffic engineering. Technical Report 583, ULB Computer Science Departement, 2007.
[26] F. Glover. Tabu search part i. *Operations Research Society of America (ORSA) Journal on Computing*, 1, 1989.
[27] F. Glover. Tabu search part ii. *Operations Research Society of America (ORSA) Journal on Computing*, 2, 1990.
[28] A. Gunnar, M. Johansson, and T. Telkamp. Traffic matrix estimation on a large IP backbone: a comparison on real data. In *Internet Measurement Conference*, pages 149–160, 2004.
[29] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: A network design problem for multicommodity flow. In *33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 389–398, 2001.
[30] I.Juva, S.Vaton, and J.Virtamo. Quick traffic matrix estimation based on link count covariances. *2006 IEEE International Conference on Communications (ICC 2006),Istanbul*, 2006.
[31] ILOG CPLEX. http://www.ilog.com/products/cplex/.
[32] A. Jüttner, I. Szabo, and A. Szentesi. On bandwidth efficiency of the hose resource management model in virtual private networks. In *INFOCOM 2003*, pages 386–395, San Francisco, USA, April 2003.
[33] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener. Algorithms for provisioning virtual private networks in the hose model. *IEEE/ACM Transactions on Networking*, 10(4):565–578, August 2002.
[34] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian. Traffic matrix estimation: Existing techniques compared and new directions, 2002.
[35] J. Moy. *OSPF, Anatomy of an Internet Routing Protocol*. Addison-Wesley, 1998.
[36] E. Mulyana and U. Killat. Optimizing ip networks for uncertain demands using outbound traffic constraints. In *INOC'2005*, pages 695–701, 2005.
[37] Networking Working Group. Ospf version 2. Technical report, Internet Engineering Task Force, 1994.
[38] A. Nucci, R. Cruz, N. Taft, and C. Diot. Design of IGP link weight changes for estimation of traffic matrices. In *IEEE Infocom*, Hong Kong, March 2004.
[39] K. Papagiannaki, N. Taft, and A. Lakhina. A distributed approach to measure IP traffic matrices. In *Internet Measurement Conference*, pages 161–174, 2004.
[40] K. Papagiannaki, N. Taft, and A. Lakhina. A distributed approach to measure IP traffic matrices. In *Internet Measurement Conference*, pages 161–174, 2004.
[41] A. Parmar, S. Ahmed, and J. Sokol. An integer programming approach to the ospf weight setting problem. Technical report, School of Industrial & Systems Engineering, Georgia Tech, 2006.
[42] P.Bermolen, S.Vaton, and I.Juva. Search for optimality in traffic matrix estimation: a rational approcah by cramer-rao lower bounds. *2nd EuroNGI Conference on Next Generation Internet Design and Engineering, Valencia, Spain*, 3-5 april 2006.
[43] M. Pioro, A. Szentesi, J. Harmatos, A. Jttner, P. Gajowniczek, and S.Kozdrowski. On open shortest path first related network optimization problems. *Performance evaluation 48*, pages 201–223, 2002.
[44] G. S. Poo and H. Wang. Multi-path routing versus tree routing for vpn bandwidth provisioning in the hose model. *Computer Networks*, 51(6):1725–1743, 2007.
[45] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot. Traffic matrices: balancing measurements, inference and modeling. *SIGMETRICS Perform. Eval. Rev.*, 33(1):362–373, 2005.
[46] A. Soule, K. Salamatian, A. Nucci, and N. Taft. Traffic matrix tracking using kalman filters. *SIGMETRICS Perform. Eval. Rev.*, 33(3):24–31, 2005.
[47] N. Springs, R. Mahajan, D. Wetherall, and T. Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1):2–16, 2004.
[48] A. Sridharan, R. Guérin, and C. Diot. Achieving near optimal traffic engineering solutions in current ospf/isis networks. In *Proceedings of INFOCOM 2003, San Fransisco, USA*, 2003.
[49] H. Ümit. A column generation approach for igp weight setting problem. In *CoNEXT*, pages 294–295, Toulouse, France, 2005.
[50] Y. Vardi. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):573–??, June 1998.
[51] Z. Wang, Y. Wang, and L. Zhang. Internet traffic engineering without full mesh overlaying. In *Proceedings of INFOCOM 2001*, Anchorage, Alaska, April 2001.
[52] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale ip traffic matrices from link loads, 2003.