Robust Link Weight Optimization Under Demand Uncertainty

Zied Ben Hamouda, Olivier Brun and Jean-Marie Garcia
CNRS, LAAS, 7, av. du Colonel Roche, F-31077 Toulouse, France
Univ de Toulouse, LAAS, F-31400 Toulouse, France
{zbenhamo, brun, jmg}@laas.fr

Anouar Rachdi
QoS Design
7, av. du Colonel Roche, F-31077 Toulouse, France.
{rachdi}@qosdesign.com

***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 to incorporate demand uncertainty into the network design problems explicitly. This paper investigates a practical implementation of a network design model considering traffic demand uncertainty. More precisely, 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 a local search based algorithm for weight optimization under demand uncertainty. This algorithm can be used either in single-start mode, for incremental weight optimization, as well as in multi-start mode, for global weight optimization. Extensive computational results show that this algorithm yields IP routings which are close to the optimal routings within low computing times with respect to known weight optimization algorithms.**

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 computing and SaaS (Software as a Service) 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 [39], [37]. 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 [26], [28], 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. A well-known online approach to handle time-varying traffic matrices is to rely on online traffic monitoring and to update routes adaptively as some changes are observed. However, these distributed load-based updates can lead to complexity and even network instability problems.

An alternative offline approach that has been recently proposed is to optimize over a set of traffic matrices [38], [4], [17], [6]. 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. Most previous works incorporating demand uncertainty have been devoted to the VPN design problem with the well-known *hose model* of uncertainty introduced by Duffield et al. [21], or to the traditional routing problem under a polyhedral demand uncertainty.

The robust link weight optimization problem has been considered only recently. Previous works assume polyhedral demand uncertainty, but otherwise do not make any other assumption on the structure of the uncertainty set. The advantage is that these methods can handle any polyhedral definition of demand uncertainty. The drawbacks however are that they left open the question of what is a meaningful uncertainty set, and that they cannot exploit the specific structure of this set in order to reduce computing times. As we argue below, traffic matrix estimation algorithms can be used to obtain upper and lower bounds on the amount of traffic of each OD flow, as well as upper bounds on the total ingress and egress traffic of each edge router. The resulting uncertainty model is a combination of what is called a box model of uncertainty in [8], and of the hose model of demand uncertainty [21]. It provides a concrete uncertainty structure that makes sense for a network operator. Moreover, by tailoring our algorithm to this specific uncertainty structure, we are able to achieve a significant reduction in computing times.

The weight optimization algorithm proposed in this paper adapts the local search techniques proposed in [24] to this specific demand uncertainty structure. It has several advantages with respect to the above mentioned works. The first one is its versatility since it can be used in single-start mode, for incremental weight optimization, as well as in multi-start mode, for global weight optimization. The incremental weight optimization enables the network operator to improve its current routing with few weight changes. If more weight changes are allowed, then global weight optimization can be used to avoid beeing trapped in a local minimum and thus get a better solution. Extensive computational results show that the proposed algorithm delivers solutions which are often very close to the lower bound given by the optimal robust multipath routing. These results show also that our algorithm is very fast with respect to the other robust link weight optimization algorithms proposed in the literature.

In the next section, we review related works. Section III is devoted to the mathematical statement of the link weight setting problem. The proposed 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 [5] for a complete survey and problem extensions). This problem is known to be NP-hard [26]. Although it can be formulated as an integer programming problem [42], this formulation does not yield an efficient approach to solve it. Therefore, most previous works have focused on approximation algorithms. They can be classified into two broad approaches: those that tries to solve an inverse shortest path problem [9], [11], [12], [15], [44], [50], and those that start from an initial solution, i.e. a set of link weights, and aim at iteratively improving it [26], [22], [14], [29], [53], [24], [51]. In this paper, we extend the local search alogrithm of [24] 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 [18]). Some attemps have to done to reduce this overhead using sampling techniques [19]. 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. Starting with [52], a first generation of methods has tried to use link load covariances as this additional information [36], [16], [33], [43]. These methods are based on statistical assumptions on the traffic demands whose validity is questionable. A second generation of methods use spatial or temporal prior information [54], [47], [40], [25]. The simplest of these methods is the tomogravity one [54] which assumes that the traffic between nodes i and j is proportional to the total incoming traffic at node i and to the total outgoing traffic at node j. As for the first generation of methods, the weak point of these methods is related to the prior information that is assumed and the validity of the estimated traffic matrix cannot be guaranteed. To overcome this difficulty, the latest generation of methods use posterior spatial and temporal information. They require a calibration phase where the traffic matrix is regularly measured using Netflow over a certain period of time and then use filtering techniques

to predict future traffic matrices [48], [41]. Although the accuracy of these methods is signifiantly superior to that of first generations methods, their drawback is the overhead induced on the network during the calibration phase.

Using any of the above traffic matrix estimation algorithms, we can easily obtain upper and lower bounds on the fluctuations over time of each traffic demand around its average value. The lower and upper bounds are obtained by taking respectively the minimum and maximum values observed during the metrology phase for the amount of traffic of this OD pair. Of course, these bounds have to integrate the possible inaccuracy of the estimation. Moreover, the network operator is free to adopt a more conservative approach by scaling these values in order to protect against bursts of traffic.

Moreover, upper bounds on the total amount of ingress and egress traffic at each edge router can be directly obtained from SNMP link counts. Here again, these upper bounds are obtained by taking the maximum values measured with SNMP for the ingress and egress traffics during the metrology phase. Our uncertainty model therefore combines lower and upper bounds on the amount of traffic of each OD flow with upper bounds on the ingress and egress traffics of edge routers. It can be viewed as a combination of the box model of demand uncertainty [8] and of the hose model of demand uncertainty [21].

C. Robust routing optimization

One of the most well-known model of demand uncertainty is probably the hose-model [21] introduced by Duffield et al. for VPN design [3], [32], [34], [35], [45], [23]. 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 [7] discuss this problem with almost no information on traffic demands. Belotti and Pinar [8] incorporate box model of uncertainty as well as statistical uncertainty into the same problem. Ben-Ameur and Kerivin [10] 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 [17]. Mulyana and Killat [38] deal with the OSPF routing problem, where traffic uncertainty is described by a set of outbound constraints. Altin et al. [4] 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 [6], Altin et al. extend the tabu search algorithm of [26] to handle polyhedral demand uncertainty.

As mentionned in the introduction, one the main contribution of the present paper with respect to the above works is related to our demand uncertainty model. By exploiting its specific structure, we are able to reduce the computation of the maximum load of a link to a standard minimum cost flow problem, which can be solved very efficiently. This allows a drastic reduction of the overall computing times.

III. Problem Statement

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. 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 (uncertain) bandwidth requirement $d_{s(k),t(k)}$. In the following, we let $\mathbf{d}$ denote the vector of traffic demands.

A. Structure of the demand uncertainty set

As explained in section II-B, our demand uncertainty model is a combination of the box model of demand uncertainty and of the hose model of demand uncertainty. Let $a_{s,t}$ and $b_{s,t}$ be the lower and upper bounds on the amount of traffic of the OD flow $(s,t) \in Q$, respectively. Let b_s^{out} and b_s^{in} denote the upper bounds on the total ingress and egress traffics of edge router $s \in Q$, respectively. Our demand uncertainty model is the polytope $\mathcal{D}$ of traffic matrices $\mathbf{d}$ satisfying the following linear constraints:

$$a_{s,t} \leq d_{s,t} \leq b_{s,t} \ , \ s,t \in Q, \tag{1}$$

$$\sum_{t \neq s} d_{s,t} \leq b_s^{out} \ , \ s \in Q, \tag{2}$$

$$\sum_{t \neq s} d_{t,s} \leq b_s^{in} \ , \ s \in Q. \tag{3}$$

Throughout the paper, it will be assumed that $\sum_{s \in Q} b_s^{out} = \sum_{t \in Q} b_t^{in}$. We will also assume that $\sum_{t \neq s} b_{s,t} > b_s^{out}$ and $\sum_{s \neq t} b_{s,t} > b_t^{in}$.

B. Feasible solutions

A feasible solution of the weight setting problem is a vector $\mathbf{w} = (w_1, \ldots, w_M)$, where $w_l \in \Omega$ is the weight assigned to arc $l \in E$ and $\Omega = \{1, \ldots, w_{max}\}$ 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 $\mathbf{w} = (w_1, \ldots, w_M)$, 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,

- $\delta^t_{u,v}(\mathbf{w}) = 1$ if link $l = (u, v)$ is on a shortest path towards destination node t, i.e. $D^t_u(\mathbf{w}) = w_l + D^t_v(\mathbf{w})$, and 0 otherwise,
- $n^t_u(\mathbf{w}) = \sum_v \delta^t_{u,v}(\mathbf{w})$ 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

$$f^l_{s,t}(\mathbf{w}) = \frac{\delta^t_{u,v}(\mathbf{w})}{n^t_u(\mathbf{w})} \sum_{(i,u)\in E} f^{i,u}_{s,t}. \tag{4}$$

Note that the above relation holds whatever the traffic matrix $\mathbf{d} \in \mathcal{D}$. It allows the recursive computation of the fractions $f^{u,v}_{s,t}(\mathbf{w})$ for each OD flow (s, t) and each link $(u, v) \in E$ once the parameters $\delta^t_{u,v}(\mathbf{w})$ and $n^t_u(\mathbf{w})$ 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

$$y_l(\mathbf{w}) = \max_{\mathbf{d}\in\mathcal{D}} \sum_{s,t\in Q} f^l_{s,t}\, d_{s,t}. \tag{5}$$

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:

$$\Phi(\mathbf{w}) = \max_{l\in E} \frac{y_l(\mathbf{w})}{C_l}. \tag{6}$$

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

$$\min_{\mathbf{w}\in\Omega^M} \Phi(\mathbf{w}). \tag{7}$$

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.

IV. A LOCAL SEARCH ALGORITHM FOR WEIGHT OPTIMIZATION

To solve the weight optimization problem, we propose a local search heuristic [1], [30], [31]. The main originality of this algorithm lies in the neighbourhood structure it uses and in the method used to evaluate the maximum link loads. The pseudo-code in figure 1 describes the proposed local search algorithm which is further described in the following sections. We emphasize that this algorithm is very similar to the one presented in [24], except for the evaluation of maximum link loads (cf Section IV-D).

Algorithm 1 Robust OSPF weight optimization algorithm

```
1:  procedure METRICOPTIMISATION
2:      Let w = (w_1, ..., w_M) be the initial solution and
            Φ(w) its cost.
3:      w* = w      ▷ Initialisation of minimum cost solution
4:      while Convergence() = false do
5:          Φ_min = ∞
6:          for l = 1 ... M do
7:              Compute Δ_l and w^l = w + Δ_l e_l
8:              Compute shortest paths: δ^t_{u,v}(w^l), n^t_u(w^l)
9:              Compute maximum load y_l(w), l ∈ E
10:             Compute cost Φ(w^l)
11:             if Φ(w^l) ≤ Φ_min then
12:                 w_next = w^l and Φ_min = Φ(w^l)
13:             end if
14:         end for
15:         w = w_next
16:         if Φ(w) < Φ(w*) then
17:             w* = w   ▷ Update of minimum cost solution
18:         end if
19:     end while
20: end procedure
```

A. Neighbourhood definition

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

$$\mathcal{N}(\mathbf{w}) = \left\{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^M\right\}, \tag{8}$$

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

$$\Delta_l = argmin_{\Delta\geq 1}\left[\sum_{s,t\in Q} f^l_{s,t}(\mathbf{w} + \Delta\, \mathbf{e}_l) < \sum_{s,t\in Q} f^l_{s,t}(\mathbf{w})\right]. \tag{9}$$

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).

B. Neighbourhood generation

Let $\mathcal{F}_l$ be the set of OD flows transmitted over link l ($f^l_{s,t} > 0$). 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 $\overline{\mathbf{w}^l} = (w_1, \ldots, w_{i-1}, \infty, w_{i+1}, \ldots, w_M)$.

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

(c) Let

$$d^l_{\min} = \min_{f\in\mathcal{F}_l}\left[D^{t(f)}_{s(f)}(\overline{\mathbf{w}^l}) - D^{t(f)}_{s(f)}(\mathbf{w})\right], \tag{10}$$

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).

C. Dynamic update of shortest paths

Two key operations are often performed during neighbourhood exploration. The first one is the update of shortest paths when a single link weight is increased. This occurs when the weight of link l is set to ∞ in order to compute Δ_l, but also when this weight is set to $w_l + \Delta_l$ in order to evaluate the neighbour $\mathbf{w}^l$.

Considering a single weight change (increase in our case), usually only a small part of the graph is affected. It is therefore sensible to avoid the computation of shortest paths from scratch, as it is the case with Dijkstra's algorithm, but only update the part of the graph affected by the arc weight change. This problem is known as the dynamic shortest path problem. Many algorithms were proposed to solve this problem, one of the most famous being the algorithm of Ramalingam and Reps [46]. As suggested in [26], we have used an improved version of this algorithm [13].

D. Maximum link loads

The other key operations that is frequently performed is the computation of maximum link loads. This operation is performed each time we need to evaluate the cost of a neighbouring solution $\mathbf{w}^m$, $m \in E$. Considering a given neighbour $\mathbf{w}^m$, we observe that we do not need to recompute the maximum load for all links, but only for those links for which the parameters $f^l_{s,t}$ have been updated. These links can easily be identified during the dynamic update of shortest paths.

Let l be one of these links. The maximum link load $y_l(\mathbf{w})$ of link l is the optimal value of (5) with the updated values of the routing coefficients $f^l_{s,t}$. From our assumptions $\sum_{t \neq s} b_{s,t} > b^{out}_s$ and $\sum_{s \neq t} b_{s,t} > b^{in}_t$, it is clear that there exists at least one optimal solution such that $\sum_{t \neq s} d_{s,t} = b^{out}_s$ and $\sum_{s \neq t} d_{s,t} = b^{in}_t$. Note that this implies that

$$\sum_{s,t \in Q} d_{s,t} = \sum_{s \in Q} b^{out}_s = \sum_{t \in Q} b^{in}_t. \tag{11}$$

With respect to the optimal value of (5), constraints (2)-(3) can therefore be replaced by the following linear equality constraints:

$$\sum_{t \neq s} d_{s,t} = b^{out}_s \ , \ s \in Q, \tag{12}$$

$$\sum_{s \neq t} d_{s,t} = b^{in}_t \ , \ t \in Q. \tag{13}$$

Let $\mathcal{D}^*$ be the polytope of traffic matrices defined by constraints (1) and (12)-(13). The maximum link load of link l can now be written as follows:

$$\begin{aligned} y_l(\mathbf{w}) &= \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} f^l_{s,t} \, d_{s,t}, \\ &= \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} f^l_{s,t} \, d_{s,t} + \sum_{s \in Q} b^{out}_s - \sum_{s \in Q} b^{out}_s, \\ &= \sum_{s \in Q} b^{out}_s + \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} f^l_{s,t} \, d_{s,t} - \sum_{s,t \in Q} d_{s,t}, \\ &= \sum_{s \in Q} b^{out}_s + \max_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} (f^l_{s,t} - 1) \, d_{s,t}, \\ &= \sum_{s \in Q} b^{out}_s - \min_{\mathbf{d} \in \mathcal{D}^*} \sum_{s,t \in Q} (1 - f^l_{s,t}) \, d_{s,t}. \end{aligned}$$

The computation of $y_l(\mathbf{w})$ therefore reduces to solving the following minimization problem:

$$\begin{cases} \min_{\mathbf{d} \in \mathbb{R}^K} & \displaystyle\sum_{s,t \in Q} (1 - f^l_{s,t}) \, d_{s,t} \\ \text{s.t.} & \\ & a_{s,t} \leq d_{s,t} \leq b_{s,t} \quad , \ s,t \in V \\ & \sum_{t \neq s} d_{s,t} = b^{out}_s \quad , \ s \in V \\ & \sum_{t \neq s} d_{t,s} = b^{in}_s \quad , \ s \in V \end{cases}$$

As depicted in Figure 1, the structure of the above problem is that of a standard minimum cost flow problem on a bipartite graph. Nodes on the left represent sources nodes while node on the right represent destination nodes. Each source node s has a supply of b^{out}_s units and each destination node t has a demand of b^{in}_t units. The cost of arc (s,t) is $1 - f^l_{s,t}$, its capacity is $b_{s,t}$ and the lower bound on its flow is $a_{s,t}$. This problem can be solved very efficiently using any of the minimum cost flow algorithms described in chapters 9, 10 and 11 of [2].

E. 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 ($\Delta_l = \infty, \ \forall l \in E$).

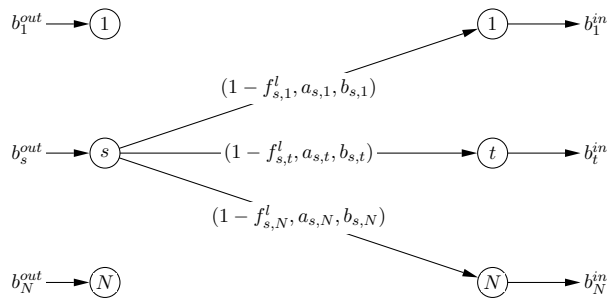

Fig. 1. Minimum cost flow problem.

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.

F. Single-start and multi-start modes

In the following, we will consider several variations of algorithm 1. First, the algorithm can be used in single-start mode, for incremental weight optimization. In the sequel, we refer to this incremental weight optimization algorithm as algorithm $A_{\rm INCR}$. The parameters of this algorithm are $Q_1 = 100$ and $Q_2 = 5$.

Algorithm 1 can also be used in multi-start mode. In this case, algorithm 1 is repeated several times, starting from (a) the current weight settings, (b) weights of each link set to 1 and (c) weights of links set inversely proportional to the link capacities. In the following we will denote such algorithm by $A_{\rm MS}$.

V. Results

We have tested our link weight optimization algorithms $A_{\rm INCR}$ and $A_{\rm MS}$ on medium-to-large-size real network topologies. We have collected the information of most instances from the IEEE literature (*bhvac, pacbell, eon, metro, and arpanet*), and from the Rocketfuel project [49] for the topologies *abovenet* and *vnsl*. For the latter topologies, arc capacities as well as upper bounds on the ingress and egress demands of edge routers have been obtained with the method described in [6]. All tests have been carried out with a Pentium Centrino 3 Ghz processor, running under Linux with 2 GB of memory available. The $A_{\rm INCR}$ and $A_{\rm MS}$ algorithms have been implemented in *C++*. We have used the LEMON library [20] to solve the minimum cost flow problem. In the following results, the initial solution of $A_{\rm INCR}$ was obtained by setting all link weights to 1.

A. Hose model of demand uncertainty

In this section, we compare our algorithms with the methods proposed by Altin et al. [5]. For the purpose of the comparison, we assume the hose model of demand uncertainty and adopt their cost function : $\Phi(\mathbf{w}) = \sum_{l \in E} \Phi_l(\mathbf{w})$, where

$$\Phi_e(\mathbf{w}) = \begin{cases} 0 & \frac{y_l(\mathbf{w})}{C_l} = 0 \\ y_l(\mathbf{w}) & 0 \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{1}{3} \\ 3y_l(\mathbf{w}) - \frac{2C_l}{3} & \frac{1}{3} \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{2}{3} \\ 10y_l(\mathbf{w}) - \frac{16C_l}{3} & \frac{2}{3} \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{9}{10} \\ 70y_l(\mathbf{w}) - \frac{178C_l}{3} & \frac{9}{10} \leq \frac{y_l(\mathbf{w})}{C_l} < 1 \\ 500y_l(\mathbf{w}) - \frac{1468C_l}{3} & 1 \leq \frac{y_l(\mathbf{w})}{C_l} < \frac{11}{10} \\ 5000y_l(\mathbf{w}) - \frac{16318C_l}{3} & \frac{11}{10} \leq \frac{y_l(\mathbf{w})}{C_l} < \infty \end{cases} .$$

The above cost function was initially introduced in [26] and later used by many works on link weight optimization. For the comparison to be fair, we use exactly the same network data that in [5]. The algorithms presented in [5] use the optimization method presented in [27] as a subroutine. The latter method allows the optimization of link weights for a discrete set of traffic matrices. To handle polyhedral demand uncertainty, Altin et al. propose to dynamicaly add traffic matrices to this discrete set. They propose two different algorithms. The first one, called *CM*, generates a demand matrix that puts the network in a worse situation as a whole for a given solution $\mathbf{w}$ on the basis of the total routing cost $\Phi(\mathbf{w})$. On the contrary, the other method, called *LM*, generates a new traffic matrix corresponding to the worst case for an individual link in terms of link utilization.

Note that the costs provided by our methods are upper bounds on the costs evaluated by Altin *et al.* since for any weight vector the following holds

$$\max_{d \in \mathcal{D}} \sum_{l \in E} \Phi_l(\mathbf{w}) \leq \sum_{l \in E} \max_{d \in \mathcal{D}} \Phi_l(\mathbf{w})$$

Thus, our methods optimize an upper bound on the cost as defined in Altin *et al.* As a consequence, when our methods return a weight vector yielding an upper bound which is lower than the cost obtained with the methods of Altin *et al.*, this weight vector gives a better solution than those obtained with LM and CM.

Tables I and II compare the costs and the running times obtained for the different network topologies using our algorithms and the *CM* and *LM* algorithms. It can be seen that except for the *abovenet* instance, our algorithms clearly outperform the *CM* and *LM* methods. Note also that our algorithm is always far more faster than these methods.

B. Combination of the Box and Hose models of demand uncertainty

In this section we consider our original model of demand uncertainty (Box + Hose model). We compare in the following the solutions generated by our algorithm $A_{\rm MS}$ with those obtained using the following algorithms:

TABLE I
COSTS OBTAINED UNDER ALGORITHMS A_{INCR}, A_{MS}, CM AND LM

Instance	A_{INCR}	A_{MS}	CM	LM
abovenet	34.9	34.9	9.4	9.4
arpanet	533.4	533.4	5.51 10^6	1.01 10^7
bhvac	2.45 10^7	2.45 10^7	3.70 10^7	3.90 10^7
eon	3.63 10^7	3.63 10^7	5.15 10^7	5.11 10^7
example	9010.3	9010.3	158094.0	171720.0
metro	19858.0	843.3	1582.9	1492.0
nsf	40213.0	40200.0	2.55 10^6	2.56 10^6
pacbell	4735.6	4735.6	1.49 10^6	541683.0
telstra	0.27	0.27	0.28	0.28
vnsl	128357.0	128357.0	128357.0	128357.0

TABLE II
RUNNING TIMES (SEC) FOR ALGORITHMS A_{INCR}, A_{MS}, CM AND LM

Instance	A_{INCR}	A_{MS}	CM	LM
abovenet	0.57	1.02	71	440
arpanet	52	101	124074	18331
bhvac	4	19	67	3084
eon	59	127	8135	10500
example	0.47	0.94	8	23
metro	1.87	2.35	78	1029
nsf	0.28	0.87	9	3
pacbell	2	4	111	485
telstra	0.8	1.14	200	96
vnsl	0.3	0.88	2	1

- The two standard heuristics, i.e. link weights set to 1 (UNIT) and link weights set inversely proportional to link capacities (INV_CAPA),
- A_{MS} under the hose model of demand uncertainty,

To obtain upper and lower bounds on the fluctuations over time of each traffic demand around its average value, we have used the tomogravity method [54] to compute the average demand and have assumed a variation of $\pm 50\%$ around this average value. Moreover, for each test, we have computed a lower bound on the cost by solving the optimal multi-path routing problem:

$$\min \quad z \tag{14}$$

s.t.

$$\sum_{s,t \in Q} f^l_{s,t}\, d_{s,t} \leq C_l\, z \qquad , \; l \in E, \mathbf{d} \in \mathcal{D} \tag{15}$$

$$\sum_{l \in \delta^+(v)} f^l_{s,t} - \sum_{l \in \delta^-(v)} f^l_{s,t} = h^v_{s,t} \;\; , \; s,t \in Q, v \in V, \tag{16}$$

$$0 \leq f^l_{s,t} \leq 1 \qquad , l \in E, \;\; s,t \in Q \tag{17}$$

where $\delta^+(v)$ (resp. $\delta^-(v)$) denotes the set of incoming (resp. outgoing) arcs at node v, while $h^v_{s,t}$ is 1 if $v = s$, -1 if $v = t$ and 0 otherwise. Note that inequalities (17) ensure that there no restriction on the routing scheme. Constraints (15) have to be ensured for every traffic matrix $\mathbf{d}$ lying in the uncertainty region $\mathcal{D}$. This leads to a semi-infinite optimization problem. As suggested in [3], this difficulty can be bypassed using duality theory.

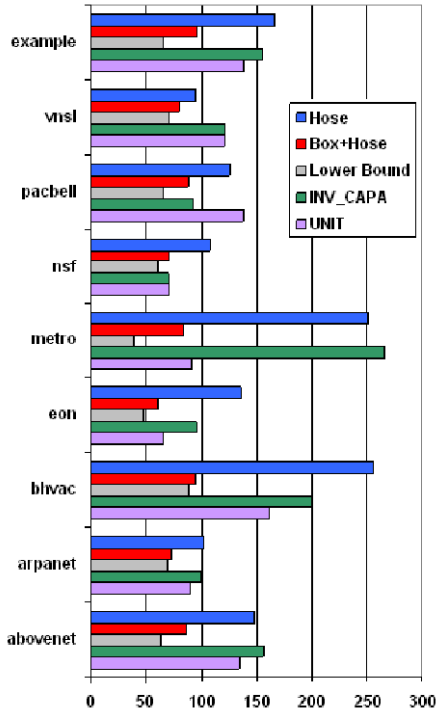

Fig. 2. Maximum utilization rate in (%)

Results are reported in figure 2. Firstly, it can be observed that A_{MS} used under the Box+Hose model of demand uncertainty always provides better solutions than the two standard heuristics UNIT and INV_CAPA (improvements of 39.9% and 77.8% on the average, respectively). This proves that A_{MS} improves significantly its initial solutions. We also observe that for most instances, A_{MS} is fairly close to the lower bound (+20% on the average). The worst gap is 45% (*metro*), and the smallest one is 4% (*arpanet*). For these two instances, the gaps of INV_CAPA are 227% and 30%, respectively.

Another interesting result emerged from the above experiments: the maximum link utilizations obtained under the Box+Hose model are always significantly lower than those of the hose model. For some instances such as *bhvac* and *metro*, the maximum link utilization under the hose model is *3* times that under the Box+Hose model. These results indicate that the hose model is really a too pessimistic model for network design. Box+Hose model allows to reject some traffic patterns which, although consistent with the b^{in} and b^{out} values, may be highly unlikely.

VI. CONCLUSION

In this paper, we have addressed the problem of link weight optimization in IP networks under traffic uncertainty. We have considered a demand uncertainty model that is a combination of the Box model and the Hose model. We have also presented

a local search based heuristic for weight optimization. The originality of the proposed algorithm lies in the neighbourhood structure it uses. One of the main advantage of this algorithm is its versatility since it can be used either in single-start mode, for incremental weight optimization, as well as in multi-start mode, for global weight optimization. Extensive computational results have shown that our heuristic algorithm yields IP routings which are close to the optimal routings within low computing times with respect to known weight optimization algorithms.

REFERENCES

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

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, 1993.

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

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

[5] 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.

[6] 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.

[7] 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.

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

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

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

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

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

[13] L. Buriol, M. G. C. Resende, , and M. Thorup. Speeding up dynamic shortest path algorithms. Technical report, AT&T Labs Research Technical Report TD-5RJ8B, 2003.

[14] 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.

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

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

[17] 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.

[18] Cisco Systems. Cisco ios netflow. http://www.cisco.com/en/us/products/ps6601/products_ios_protocol_group _ home.html.

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

[20] COIN-OR::LEMON. Lemon graph library. https://lemon.cs.elte.hu/trac/lemon.

[21] 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.

[22] 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.

[23] T. Erlebach and M. Rüegg. Optimal bandwidth reservation in hose-model vpns with multi-path routing. In *INFOCOM*, 2004.

[24] C. Fortuny, O. Brun, and J. M. Garcia. Metric optimization in ip networks. In *19th International Teletraffic Congress*, pages 1225–1234, Beijing, China, 2005.

[25] 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.

[26] B. Fortz and M. Thorup. Internet traffic engineering by optimizing ospf weights. In *Proc. 19th IEEE Conf. on Computer Communications (INFOCOM)*, 2000.

[27] B. Fortz and M. Thorup. Optimizing ospf/is-is weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, 2002.

[28] B. Fortz and M. Thorup. Increasing internet capacity using local search. *Computational Optimization and Applications*, 29:13–48, 2004.

[29] B. Fortz and H. Ümit. Efficient techniques and tools for intra-domain traffic engineering. Technical Report 583, ULB Computer Science Departement, 2007.

[30] F. Glover. Tabu search part i. *Operations Research Society of America (ORSA) Journal on Computing*, 1, 1989.

[31] F. Glover. Tabu search part ii. *Operations Research Society of America (ORSA) Journal on Computing*, 2, 1990.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian. Traffic matrix estimation: Existing techniques compared and new directions, 2002.

[37] J. Moy. *OSPF, Anatomy of an Internet Routing Protocol*. Addison-Wesley, 1998.

[38] E. Mulyana and U. Killat. Optimizing ip networks for uncertain demands using outbound traffic constraints. In *INOC'2005*, pages 695–701, 2005.

[39] Networking Working Group. Ospf version 2. Technical report, Internet Engineering Task Force, 1994.

[40] 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.

[41] K. Papagiannaki, N. Taft, and A. Lakhina. A distributed approach to measure IP traffic matrices. In *Internet Measurement Conference*, pages 161–174, 2004.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] G. Ramalingam and T. Reps. An incremental algorithm for a generalization of the shortest path problem. *Journal of Algorithms*, 21:267–305, 1996.

[47] 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.

[48] A. Soule, K. Salamatian, A. Nucci, and N. Taft. Traffic matrix tracking using kalman filters. *SIGMETRICS Perform. Eval. Rev.*, 33(3):24–31, 2005.

[49] N. Springs, R. Mahajan, D. Wetherall, and T. Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1):2–16, 2004.

[50] 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.

[51] H. Ümit. A column generation approach for igp weight setting problem. In *CoNEXT*, pages 294–295, Toulouse, France, 2005.

[52] Y. Vardi. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):573–??, June 1998.

[53] Z. Wang, Y. Wang, and L. Zhang. Internet traffic engineering without full mesh overlaying. In *Proceedings of INFOCOM 2001*, Anchorage, Alaska, April 2001.

[54] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale ip traffic matrices from link loads, 2003.