Optimizing the deployment of Netflow to infer traffic matrices in large networks

Mustapha Bouhtou*, Stephane Gaubert**, Guillaume Sagnol**
*Orange Labs, Issy-Les-Moulineaux, FRANCE.
Email : mustapha.bouhtou@orange-ftgroup.com
** INRIA Saclay & Centre de Mathématiques APpliquées, Ecole Polytechnique, Palaiseau, FRANCE.
Email:{guillaume.sagnol,stephane.gaubert}@inria.fr

ABSTRACT

We study the classical problem of inferring the traffic on each Origin-Destination (OD) pair of a large IP network. The most recent methods take advantage of network-monitoring tools such as Netflow (Cisco Systems), which supplement the link measurements by direct information on the OD flows. The aim is now to reduce the costs of deployment of Netflow, and to optimize its use. We formulate a combinatorial optimization problem, whose objective is to find the "best" set of interfaces on which Netflow should be activated. Our approach relies on an experimental design model, in which a robust measure of the quality of the Netflow deployment is provided by a family of Schatten norm like functions of the correlation matrix of the estimation error. Then, the deployment is found by rounding the optimal solution of a relaxed convex programming problem, that we solve by a sequential quadratic programming (SQP) algorithm, or by a projected gradient algorithm which takes advantage of the sparsity of the observation matrices in order to scale well. We also show that the present combinatorial optimization problem is equivalent to the maximization of a nondecreasing submodular function over a matroid, for which approximation algorithms are known. We report experimental results on the international backbone of France Telecom (Opentransit), which is made of 100 routers, 267 links and 5646 OD pairs. We compared several algorithms, among which the convex relaxation with rounding scheme gives the best results.

Keywords:.
Traffic measurement, Netflow deployment, Experimental design, Combinatorial optimization.

1. INTRODUCTION

1.1 Background

The problem of estimating Origin-Destination (OD) traffic matrices for backbone networks has recently attracted much interest from both Internet providers and the network research community [6, 15, 13, 22], because these traffic matrices serve as important inputs of a variety of network traffic engineering tasks. This estimation problem is generally stated as follows. We are given the graph of the network, with its set of l edges (or links). Direct measurements are provided by the Simple Network Management Protocol (SNMP), which allows us to know some statistics on the links (for instance, the number of bytes seen on each link in a 5 minutes window). We will denote these SNMP link counts by $Y^{\mathrm{SNMP}} = (y_1, ..., y_l)^T$. We are also given the set of routes among the network, that is to say the set of m OD pairs, and for each pair, we are given the set of links that a byte needs to traverse to go from origin O to destination D. The information about the routing is classically gathered in the $l \times m$ incidence matrix A: this is a $0/1-$matrix whose $(e, r)-$entry takes the value 1 if and only if the OD pair r traverses edge e. More general routing policies of the provider may be modeled by considering matrices in which $A_{e,r}$ is a real number representing the fraction of the traffic from OD pair r that traverses link e.

The unknown in our problem is the vector of OD flows $X = (x_1, ..., x_m)$, where x_r is the number of bytes which have been traveling through OD pair r during the observation period. One might easily verify that the following must hold:

$$Y^{\mathrm{SNMP}} = AX. \tag{1}$$

In typical networks, we have $l << m$, so that the equation above is underdetermined. Thus, the estimation of the flow distribution X is an ill-posed problem in the sense that the system equations (1) has an infinite number of solutions, and hence we have to introduce some constraints to ensure the identifiability of the model.

1.2 Optimization of the measurement

A way to introduce some new constraints is the use of a network-monitoring tool such as Netflow (Cisco systems). Liang, Yu and Taft [14] proposed a scheme for selecting the flows that must be measured by Netflow at a certain iteration, in order to improve the accuracy of the dynamic traffic estimation. Of course, activating Netflow everywhere on the network yields an extensive knowledge of the OD flows. However, the use of Netflow induces significant costs. According to [1] indeed, activating Netflow on an interface of a router causes the CPU load of the router to increase by 10 to 50%. Moreover, the installation and maintenance costs cannot be neglected. It is thus of great interest to optimize the use of this tool. We shall distinguish two problems. The first one concerns an Internet provider wishing to have an extensive knowledge of the traffic on the network. Rather than activating Netflow everywhere, this provider may look for a minimal set of interfaces on which Netflow should be activated to measure the traffic on all the routes. The second problem concerns the situation where the provider does not have enough resources to get a full information on the traffic. Then, a challenging issue is to specify the "best" set of interfaces where Netflow should be deployed, taking into account the budget constraint.

The rest of this paper is organized as follows. The statement of the problem and the experimental design background is derived in section 2. We establish NP-hardness and polynomial time approximability results in section 3, and we present different techniques to solve it, such as semidefinite and convex programing relaxations in section 4. We give some experimental results and analysis in section 5.

1.3 Related Work

Before presenting our approach, let us review the work that has been done recently on the Netflow placement problem.

Zang and Nucci [20] posed the Netflow placement problem as an Integer program whose objective is to minimize the cost of deployment of the monitoring tool on the network, taking into account the costs required to upgrade the routers so that they support Netflow. The constraint of the program is that Netflow should monitor at least a fraction α of all the traffic that is traveling on the network. They proposed two heuristics in order to give a near-optimal solution to this NP-Hard integer program.

Bouhtou and Klopfenstein [3] pursued this approach by taking into account the variations of the traffic in time. In most networks, the routing table is not static indeed, and the placement of Netflow must be robust to possible routing modifications. To tackle this issue, the authors formulated an optimization problem with probability constraints, which they approximated by a sequence of integer linear programs.

Cantieni, Iannaccone, Barakat, Diot and Thiran [5] interested themselves in the optimal rates at which Netflow should be sampled on each router. They formulated this problem as a convex minimization problem, which they solved using a projected gradient algorithm.

Bermolen, Vaton and Juva [2] were the first to investigate the optimal placement of Netflow in light of the *experimental design* background. Based on the model proposed by Cao et al. [6], they suggested that the observation vector Y has a normal distribution, whose expected value and covariance matrix depends on the expected value λ of the OD flows, and derived the Fisher information matrix for any placement of the measures. The authors of [2] give a scheme for selecting a few interfaces on which Netflow should be activated in priority.

By comparison with this previous work, the originality of the present approach is to formulate a combinatorial optimization problem, taking both into account the SNMP data and the Netflow measurements, which can be solved by making use of continuous relaxations. We applied the method to the international transit backbone of France-Telecom (Opentransit). We also give a polynomial time approximability ratio for this problem.

2. EXPERIMENTAL DESIGN BACKGROUND AND PROBLEM STATEMENT

The experimental design approach deals with the minimization of covariance matrices. We next present a model of the deployment of Netflow which will lead us to a problem of this nature.

When Netflow is activated on an interface of the network, it will analyze the headers of the packets traversing this interface, and as a result we will have access to some statistics, such as the source and destination IP addresses, and the source and destination AS numbers of these packets. However, we are not directly interested in this information, because we are not trying to estimate the global run of the packets, but only the part of their run which is inside the network of interest, like the backbone of an autonomous system (AS).

Practically, the data that we are able to measure with Netflow depends on the knowledge that we have of the routing policy of an Internet provider. In some cases, when Netflow is activated on an interface, we will be able to simulate the routing protocol and thus to compute each OD flow that is traversing this interface, whereas in other cases, Netflow measurements will provide us some linear combinations of the flows only, such as for example the sum of all OD flows traversing this interface and which will leave the network from a particular router (the final destination inside the network of interest is known, but not the source). In this second case, Netflow will give full information on the OD flows when activated on an ingress interface of the network (we know that the packets have entered the network at this interface). This remark shows how important the location of the measure is: depending on the location of the interface inside the network, the measure can bear some valuable or some poor information.

Without any loss of generality, we can thus assume that, when Netflow is performing a measure on the k^{th}

interface i_k, we get a multidimensional measure $Y^{(k)}$ which is a linear combination of the OD flows which traverse i_k:

$$Y^{(k)} = A^{(k)} X. \tag{2}$$

Now, let $\mathcal{I} = \{i_1, i_2, ..., i_s\}$ be the set of all interfaces where one could activate Netflow. Furthermore, let $\mathcal{I}^0 = \{k_1, ..., k_n\}$ be the subset of $\{1, ..., s\}$ that denotes the interfaces on which Netflow is activated. Similarly, define the *design* w as the 0/1 vector of size s, where w_k equals 1 if and only if Netflow is activated on the interface i_k:

$$(\text{Netflow activated on } i_k) \iff k \in \mathcal{I}^0 \iff w_k = 1.$$

The measurement vector Y is now the concatenation of the SNMP data Y^{SNMP} with all the Netflow measurements $(Y^{(k)})_{k \in \mathcal{I}^0}$. The measurements are never exact in practice, and we have to deal with a noise ϵ, which is a result, among other things, of Netflow sampling and lost packets. This can be modeled as follows:

$$Y = A(w)\ X + \epsilon, \tag{3}$$

$$\text{where } Y = \begin{pmatrix} Y^{\text{SNMP}} \\ Y^{(k_1)} \\ \vdots \\ Y^{(k_n)} \end{pmatrix} \text{ and } A(w) = \begin{bmatrix} A \\ A^{(k_1)} \\ \vdots \\ A^{(k_n)} \end{bmatrix}.$$

Now, assume that we have enough measurements, so that $A(w)$ is of full rank, and assume that the noises on the observations are independent one from another, that is to say that the covariance matrix $\Sigma = \mathbb{E}(\epsilon\epsilon^T)$ only has diagonal entries. These assumptions are sufficient to use a common result in the field of statistics, which states that the best linear unbiased estimator of X is given by a pseudo inverse formula (Aitken estimator). Its variance is given below:

$$\hat{X} = \Big(A(w)^T \Sigma^{-1} A(w)\Big)^{-1} A(w)^T \Sigma^{-1} Y. \tag{4}$$

$$\text{Var}(\hat{X}) = (A(w)^T \Sigma^{-1} A(w))^{-1}. \tag{5}$$

If we further assume that the noise follows a normal distribution $\mathcal{N}(0, \Sigma)$, then the estimator $\hat{X}$ described in Eq.(4) is also the maximum likelihood estimator of X, and the bound given by the Cramer-Rao inequality [2] is attained, i.e. its covariance matrix equals the inverse of the Fisher information matrix:

$$M_F(X) = A(w)^T \Sigma^{-1} A(w). \tag{6}$$

For simplicity of notation, we will assume that the noises have unit variance: $\mathbb{E}(\epsilon\epsilon^T) = I$. We may always reduce to this case by a left diagonal scaling of the matrix $A(w)$. Now, the Fisher information matrix depends only on the interfaces where Netflow has been activated, and we will denote it by $M_F(w)$:

$$\begin{aligned} M_F(w) &= A(w)^T A(w) \\ &= A^T A + \sum_{k=1}^{s} w_k {A^{(k)}}^T A^{(k)} \end{aligned} \tag{7}$$

Our approach will consist in choosing the set of interfaces $\mathcal{I}^0$ (or the design w) in order to make the variance of the estimator (4) *as small as possible*. The interpretation is straightforward: with the assumption that the noise ϵ is normally distributed, for every probability level α, the estimator $\hat{X}$ lies in the confidence ellipsoid centered at X and defined by the following inequality:

$$(X - \hat{X})^T Q (X - \hat{X}) \leq \kappa_\alpha, \tag{8}$$

where κ_α depends on the specified probability level, and $Q = M_F(w)$ is the inverse of the covariance matrix $\text{Var}(\hat{X})$. We would like to make these confidence ellipsoids *as small as possible*, in order to reduce the uncertainty on the estimation of X. To this end, we can express the inclusion of ellipsoids in terms of matrix inequalities. Here and in the sequel, we denote by $\mathbb{S}_m$ the space of symmetric $m \times m$ matrices. We also denote by $\mathbb{S}_m^+ \subset \mathbb{S}_m$ the cone of positive semidefinite matrices, and by $\mathbb{S}_m^{++}$ its interior, which consists of positive definite matrices. The space of symmetric matrices is equipped with the *Loewner ordering*, which is defined by

$$\forall B, C \in \mathbb{S}_m, \qquad B \succeq C \iff B - C \in \mathbb{S}_m^+.$$

Let w and w' denote two designs such that the matrices $M_F(w)$ and $M_F(w')$ are invertible. One can readily check that for any value of the probability level α, the confidence ellipsoid (8) corresponding to $Q = M_F(w)$ is included in the confidence ellipsoid corresponding to $Q = M_F(w')$ if and only if $M_F(w) \succeq M_F(w')$. Hence, we will prefer design w to design w' if the latter inequality is satisfied.

We note that the problem of maximizing $M_F(w)$ with respect to the Loewner ordering remains meaningful even when $M_F(w)$ is not of full rank. This case does arise in under-instrumented situations, in which some constraints may not allow us to deploy Netflow on a number of interfaces which is sufficient to observe the traffic on all the routes.

Since the Loewner ordering on symmetric matrices is only a partial ordering, the problem consisting in maximizing $M_F(w)$ is ill-posed. So we will rather maximize a scalar *information function* of the Fisher matrix, i.e. a function mapping $\mathbb{S}_m^+$ onto the real line, and which satisfies natural properties, as positive homogeneity, monotonicity with respect to Loewner ordering, and concavity. For a more detailed description of the information functions, the reader is referred to the book of Pukelsheim [18], who proposes the use of a class of functions: the matrix means Φ_p, which are defined like the L_p-norm of the vector of eigenvalues of the Fisher information matrix, but for $p \in [-\infty, 1]$. For positive definite matrices, $M \in \mathbb{S}_m^{++}$ with eigenvalues $\{\lambda_1, ..., \lambda_m\}$, the matrix mean Φ_p is represented by

$$\Phi_p(M) = \begin{cases} \lambda_{\min}(M) & \text{for } p = -\infty \ ; \\ (\frac{1}{m} \text{ trace } M^p)^{\frac{1}{p}} & \text{for } p \in \ ]-\infty, 1], \ p \neq 0; \\ (\det(M))^{\frac{1}{m}} & \text{for } p = 0, \end{cases} \tag{9}$$

where we have used the extended definition of powers of matrices M^p for arbitrary real parameters p: trace $M^p = \sum_{j=1}^m \lambda_j^p$. The case $p = 0$ (D-optimal design) admits a simple geometric interpretation: the volume of the confidence ellipsoid (8) is given by $C_m \kappa_\alpha^{m/2} \det(Q)^{-1/2}$ where $C_m > 0$ is a constant depending only of the dimension. Hence, maximizing $\Phi_0(M_F(w))$ is the same as minimizing the volume of every confidence ellipsoid. For singular positive semi-definite matrices, we have

$$\Phi_p(M) = \begin{cases} 0 & \text{for } p \in [-\infty, 0], \ ; \\ (\frac{1}{m} \text{ trace } M^p)^{\frac{1}{p}} & \text{for } p \in \ ]0, 1]. \end{cases} \tag{10}$$

We can finally give a mathematical formulation to the problem of optimizing the use of Netflow. Assume that the cost of deployment/activation of Netflow on interface i_k is c_k. If an Internet provider has a limited budget B, the *Netflow Optimal Deployment* problem is:

$$\max_{w \ \in \{0,1\}^s} \Phi_p\Big(M_F(w)\Big) \tag{11}$$
$$s.t. \quad \sum_k w_k c_k \leq B$$

In the case where the cost of deployment c_k is the same everywhere, the constraint is equivalent to deploy Netflow on no more than n interfaces. We call this special case the *unit-cost* case.

Another interesting idea to find an optimal deployment of Netflow is to choose the design which maximizes the rank of the observation matrix $A(w)$, or equivalently of $M_F(w) = A(w)^T A(w)$. The *rank optimization* is a nice combinatorial problem, where we are looking for a subset of matrices whose sum is of maximal rank:

$$\max_{w \in \{0;1\}^s} \text{rank}\Big(A^T A + \sum_k w_k A^{(k)^T} A^{(k)}\Big) \tag{12}$$
$$s.t. \sum_k w_k c_k \leq B$$

However, this problem could provide a solution for which the matrix $A(w)$ (as consequently $M_F(w)$) has certainly a high rank, but might be very ill-conditioned. This will occur when the smallest non-zero eigenvalue of $M_F(w)$ is very small. Consequently, this model does not take the noise sensibility of the observation into account. The use of information functions Φ_p (9) is a way to get rid of this problem, and the next proposition shows that (11) may be thought as a regularization of the *rank optimization* problem. First notice that when $p > 0$, the maximization of $\Phi_p(M_F(w))$ is equivalent to the maximization of $\varphi_p(w) = \text{trace}(M_F(w))^p$.

PROPOSITION 2.1. *For all positive semidefinite matrix $M \in \mathbb{S}_m^+$,*

$$\lim_{p \to 0^+} \text{trace } M^p = \text{rank } M. \tag{13}$$

PROOF. Let $\lambda_1, ..., \lambda_r$ denote the positive eigenvalues of M, counted with multiplicities, so that r is the rank of M. We have the first order expansion as $p \to 0^+$:

$$\text{trace } M^p = \sum_{k=1}^r \lambda_k^p = r + p \ \log(\prod_{k=1}^r \lambda_k) + \mathcal{O}(p^2) \quad \square \tag{14}$$

Consequently, trace M^0 will stand for rank(M) and $\varphi_0(w)$ will stand for the objective function of (12) in the sequel.

COROLLARY 2.2. *If $p > 0$ is small enough, then every design w^* which is a solution of Problem (11) maximizes the rank of $M_F(w)$. Moreover, among the designs which maximize this rank, w^* maximizes the product of nonzero eigenvalues of $M_F(w)$.*

PROOF. Since there is only a finite number of designs, it follows from Eq. (14) that for $p > 0$ small enough, every design which maximizes φ_p must maximize in the lexicographical order the rank of $M_F(w)$, and then the *sub-determinant* $\prod_{\lambda_k > 0} \lambda_k$. $\square$

As indicated in the introduction, another interesting problem arises when the provider wishes to minimize the budget of deployement of Netflow, under the constraint that the design gives a measurement of a prescribed quality. This leads us to formulate the problem of the *minimal exhaustive deployment of Netflow*:

$$\min_{w \ \in \{0,1\}^s} \sum_{k=1}^s w_k c_k \tag{15}$$
$$s.t. \quad \Phi_p\Big(M_F(w)\Big) \geq \gamma$$

where $\gamma > 0$ is a threshold quantifying the quality of the measure. Note that when $p \leq 0$, the latter constraint forces $M_F(w)$ to have full rank.

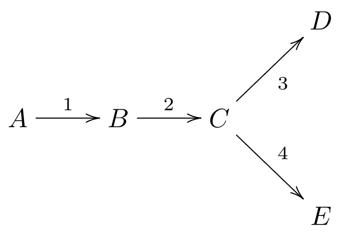
3. HARDNESS AND APPROXIMABILITY RESULTS

After having shown with an example that the greedy algorithm is suboptimal for Problem (11), we will prove that the *rank optimization* problem is NP-hard by reduction of MAX-k-Coverage. Next, we show that problem (11) is equivalent to the maximization of a *nondecreasing submodular* function [10, 19, 16], which will allow us to use known approximability results.

3.1 Non-optimality of greedy schemes

A natural idea to solve problem (11) (in the unit-cost case) is to choose n interfaces using a iterative scheme: at the k^{th} stage of the algorithm, we are given the previously $(k-1)$ selected interfaces, and we choose the k^{th} interface as the one which maximizes the given criteria. However, this algorithm is suboptimal, as shown by this simple counter-example :

Example Consider the network depicted below,



whose routing matrix is given by :

OD Pairs→ ↓*links*	AD	BD	CD	AE	BE	CE
$1(a \to b)$	1	0	0	1	0	0
$2(b \to c)$	1	1	0	1	1	0
$3(c \to d)$	1	1	1	0	0	0
$4(c \to e)$	0	0	0	1	1	1

In this example, we assume that when Netflow analyzes a packet, we are able to find its origin and its destination. For ease of presentation, we have chosen an example where not all the OD pairs are listed (for example, there is no road from A to B). Denoting the i^{th} vector of the canonical basis of $\mathbb{R}^6$ by e_i, we have the following observation matrices :

$$A^{(1)} = [e_1, e_4]^T \tag{16}$$

$$A^{(2)} = [e_1, e_2, e_4, e_5]^T \tag{17}$$

$$A^{(3)} = [e_1, e_2, e_3]^T \tag{18}$$

$$A^{(4)} = [e_4, e_5, e_6]^T \tag{19}$$

In this example, we are taking $p = \frac{1}{10}$. We are looking for the set of interfaces which maximizes $\Phi_{0.1}(M_F(w))$, or equivalently, $\mathrm{trace}(M_F(w)^{0.1})$. In the array below, we have listed the values of this criterion for each set of interfaces of cardinality less than 2 :

w	$\mathrm{trace}(M_F(w)^{0.1})$	w	$\mathrm{trace}(M_F(w)^{0.1})$
0	4.333633	$e_1 + e_2$	6.381055
		$e_1 + e_3$	6.332209
e_1	5.311219	$e_1 + e_4$	6.332209
e_2	6.284268	$e_2 + e_3$	6.489883
e_3	6.189830	$e_2 + e_4$	6.489883
e_4	6.189830	$e_3 + e_4$	6.502424

These values indicate that if we want to place 1 measure, we should select the interface $i_2(b \to c)$, and if we want to place 2 measures, we should rather activate the interfaces $i_3(c \to d)$ and $i_4(c \to e)$. If we had used a greedy algorithm to select 2 interfaces, we would have first activated the interface $i_2(b \to c)$, and then either $i_3(c \to d)$ or $i_4(c \to e)$, which are equivalent by symmetry of the network. This shows that the greedy selection scheme is suboptimal, yet achieving a very good approximation ratio of $\frac{6.4899-4.3336}{6.5024-4.3336} \simeq 0.993$.

An intuitive way to understand this result is to look at the observation matrices : $A^{(2)}$ is the best observation matrix, because it gives some information on 4 OD flows, namely AD, BD, AE and BE, while the others give information on no more than 3 interfaces. Selecting this interface (i_2) plus another one will be redundant, because in the best case, we will have information on one more OD flow (for example if we activate Netflow on i_2 and i_4, we will have some information on the 4 OD flows mentioned above, plus CE). On the other hand, activating Netflow on interfaces i_3 and i_4 will do a better job, because the information given by i_3 is totally disjoint from that of i_4, and activating Netflow on these 2 interfaces gives full information on each of the 6 OD pairs of this network.

3.2 Hardness of *Rank optimization*

THEOREM 3.1. *Problem* (12) *is NP-Hard. For all positive ε, there is no polynomial-time algorithm which approximates* (12) *in a factor of* $1 - \frac{1}{e} + \varepsilon$ *unless* $P = NP$.

PROOF. We will show that the problem MAX-k-coverage, for which the statement of the theorem is true [8], reduces to Problem (11) in polynomial time.

The problem MAX-k-Coverage is defined as follows : We are given a collection of subsets $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ of $\{1, ..., N\}$, as well as an integer k, and the goal is to pick at most k sets of $\mathcal{S}$ such that the size of their union is maximized. Let e_i be the i^{th} vector of the canonical basis of $\mathbb{R}^N$. If the set S_i contains the k elements $\{i_1, i_2, ..., i_k\}$, we define the i^{th} observation matrix as : $A^{(i)} = [e_{i_1}, ..., e_{i_k}]^T$, so that $A^{(i)^T} A^{(i)}$ is a diagonal matrix whose indices of nonzero entries are the elements of S_i. Finally, let A be the all-zero row vector of size N. Since all the matrices $A^{(i)^T} A^{(i)}$ have only diagonal entries, it is straightforward to see that the rank of $A^T A + \sum_k w_k A^{(k)^T} A^{(k)}$ is equal to the number of nonzero elements on its diagonal, i.e. the cardinal of $\cup_{\{i|w_i=1\}} S_i$, which is exactly the objective function of the MAX-k-Coverage problem. $\square$

This is a negative result on the approximability of the best Netflow deployment. Nevertheless, we show in the next subsection that the bound provided by Theorem 3.1 is the worst possible ever, and that the greedy algorithm always attains it in the unit-cost case.

3.3 Submodularity of φ_p and polynomial time approximability of *Netflow Optimal Deployment*

Definition A real valued set function $f : 2^E \longrightarrow \mathbb{R}$ is *nondecreasing submodular* if it satisfies the following conditions :

- $f(I) \leq f(J)$ whenever $I \subseteq J \subseteq E$;
- $f(I) + f(J) \geq f(I \cup J) + f(I \cap J)$ for all $I, J \subseteq E$.

In the following, we will identify the function $\varphi_p : \{0,1\}^s \longrightarrow \mathbb{R}$ with the set function $\varphi_p : 2^{\{1,\ldots,s\}} \longrightarrow \mathbb{R}$. The next lemma will be useful to show that φ_p is submodular. Its proof is provided in appendix.

LEMMA 3.2. *For all* $X, Y, Z \in \mathbb{S}_m^+$, $\forall p \in [0,1]$,

$$\operatorname{trace}(X+Y+Z)^p + \operatorname{trace} Z^p \leq \operatorname{trace}(X+Z)^p + \operatorname{trace}(Y+Z)^p \quad (20)$$

The next results show that the problems (11) and (12) are $1-\frac{1}{e}-$approximable in polynomial time. This can be attained with the help of the greedy algorithm, whose principle is to add sequentially in $\mathcal{G} = \emptyset$ the interfaces which provide the best ratio

$$\frac{\varphi_p(\mathcal{G} \cup i_k) - \varphi_p(\mathcal{G})}{c_k}$$

until the budget constraint is violated.

THEOREM 3.3. (SUBMODULARITY OF φ_p) *For all* $p \in [0,1]$, φ_p *is a nondecreasing submodular set function.*

PROOF. The function φ_p is nondecreasing, because $X \longrightarrow X^p$ is a matrix monotone function [21] for $p \in [0,1]$. Let $I, J \subseteq 2^{\{1,\ldots,s\}}$. We define

$$M^{(k)} = {A^{(k)}}^T A^{(k)},\ X = \sum_{k \in I \setminus J} M^{(k)},$$
$$Y = \sum_{k \in J \setminus I} M^{(k)},\ Z = A^T A + \sum_{k \in I \cap J} M^{(k)}.$$

Now it is easy to check that $\varphi_p(I) = \operatorname{trace}(X+Z)^p$, $\varphi_p(J) = \operatorname{trace}(Y+Z)^p$, $\varphi_p(I \cap J) = \operatorname{trace} Z^p$ and $\varphi_p(I \cup J) = \operatorname{trace}(X+Y+Z)^p$. Hence, Lemma 3.2 proves the submodularity of φ_p. $\square$

COROLLARY 3.4. (APPROXIMABILITY OF OPTIMAL NETFLOW DEPLOYMENT: UNIT-COST CASE) *The greedy algorithm for problem* (11) *yields a* $1-\frac{1}{e}$ *approximation factor in the unit-cost case.*

PROOF. Nemhauser proved it for any nondecreasing submodular function over a uniform matroid. Moreover when the maximal number of interfaces which can be selected is k, this approximability ratio can be improved to $1 - \left(1 - 1/k\right)^k$. $\square$

COROLLARY 3.5. (APPROXIMABILITY OF OPTIMAL NETFLOW DEPLOYMENT) *Problem* (11) *is still* $1-\frac{1}{e}$*-approximable in polynomial time in the budgeted case, but the greedy algorithm for problem* (11) *yields a constant approximation factor of only* $\frac{1}{2}\left(1-\frac{1}{e}\right)$.

PROOF. This was proven for an arbitrary nondecreasing submodular function in the recent paper [12]. In order to attain the $1 - 1/e-$approximation guarantee, one can associate the greedy algorithm with the partial enumeration of all triples of interfaces. $\square$

Remark The previous corollaries hold in particular for $p = 0$, and hence for the *rank maximization* problem.

4. RESOLUTION OF THE PROBLEM

In this section, we investigate the methods to solve *Netflow optimal deployment* and *Minimal exhaustive deployment of Netflow*, namely (11) and (15). After a brief discussion on the choice of p, we will present some continuous relaxations to solve (11) and (15), and we propose rounding schemes to approximate the integer solution. In order to make some comparisons, we also investigated the use of some classical metaheuristics.

4.1 Discussion on the choice of p

The use of nonpositive values of p ($p \leq 0$) forces the matrix $M_F(w)$ to be of full rank. We are indeed maximizing a nonnegative function whose value is 0 as long as $M_F(w)$ is singular (10). In particular, the problems with the values $p = -\infty, p = -1, p = 0$ have been extensively studied in the experimental design literature, and are known respectively as E-, A-, and D-optimal design. However, when we have limited resources, it is not always possible to find a design w for which $M_F(w)$ is of full rank. For such cases, positive values of p ($0 < p \leq 1$) are well adapted.

A solution for the case $p = 1$ is known as T-optimal design, because it maximizes the **T**race of the Fisher information matrix. Notice that in this case, the routing matrix A appears only in the additive constant $\frac{1}{m}\operatorname{trace} A^T A$ of the objective value of (11): the T-optimal design does not take into account the SNMP data. Moreover, when the routing matrix only has 0/1 entries, the criterion which can also be seen as the Frobenius norm of the observation matrix $\|A(w)\|_F$, counts exactly the number of 1 is the matrix $A(w)$. In other words, the objective function of the T-optimal problem is the number of flows that Netflow monitors, counted with their order of multiplicity. Therefore, T-optimal design does not take into account the redundancy of the measure, which is probably the main difficulty of this optimal deployment problem. This remark should warn us not to take a value of p close to 1.

On the other hand, we have shown (proposition 2.1) that the rank optimization problem can be recovered by letting $p \longrightarrow 0^+$. Moreover, as explained in Corollary 2.2, when $p > 0$ is small enough, we are looking for a design which maximizes, in the lexical order, the rank and then the product of the nonzero eigenvalues of $M_F(w)$. This can be seen as a generalization of the D-optimal design, which maximizes the determinant of $M_F(w)$, i.e. minimizes the volume of the confidence ellipsoid of the estimator. Thus, small values of p might be a good choice in the underinstrumented case. The experimental results presented in the next section show the robustness of the optimal design for p taking several values in the range $[0.05; 0.2]$.

4.2 Continuous relaxations for the *Netflow optimal deployment*

To solve Problem (11), we first replace the integer constraint $w \in \{0,1\}^s$ by

$$0 \leq w_k \leq 1, \forall k \in \{1,...,s\}.$$

This relaxation lets problem (11) become a convex program. We can therefore apply different techniques to solve it, such as projected gradient algorithms. Moreover, the solution of this relaxed program might be interpreted as a sampling of Netflow. Indeed, a simplified way to model a sampled Netflow, which analyzes only a fraction w_i of the packets on the i^{th} interval, would be to change the variance of the noise on the i^{th} interface from 1 to the new value $\frac{1}{w_i}$. The variance of the Aitken estimator (5) becomes :

$$(A(w)^T\Sigma^{-1}A(w))^{-1} = \Big(A^TA + \sum_k w_k {A^{(k)}}^T A^{(k)}\Big)^{-1}.$$

With this model, we find a continuous version of the fisher information matrix $M_F(w)$, defined for all vector $w\ in[0,1]^s$.

Denoting by $\mathcal{F}$ the feasible set

$$\{w \in \mathbb{R}^s | \quad \sum_k w_k c_k \leq B; \quad \forall j \in \{1,...,s\}, 0 \leq w_j \leq 1\},$$

the problem takes a simpler form (for $0 < p \leq 1$):

$$\max_{w\in\mathcal{F}} \quad \varphi_p(w) \equiv \operatorname{trace}\Big(A^TA + \sum_{k=1}^s w_k {A^{(k)}}^T A^{(k)}\Big)^p \quad (21)$$

As proposed in [17], we solved this program with a sequential quadratic programming (SQP) algorithm, whose principle is to replace, at each step, the objective function φ_p with its quadratic approximation around the current value w_c, and to solve the quadratic optimization problem

$$\max_{w\in\mathcal{F}} \ \nabla\varphi_p(w - w_c) + (w - w_c)^T \nabla^2 \varphi_p(w - w_c). \quad (22)$$

To this end, we need compute the gradient and the hessian of φ_p, for which explicit forms are known thanks to basic results concerning central functions of matrices (functions of a matrix whose value depends only on the unordered $m - uple$ of eigenvalues [4]). If we have the spectral decomposition $M_F(w) = Q^T D Q$, with $D = \operatorname{diag}([\lambda_1, ..., \lambda_m])$,

$$\frac{\partial \varphi_p(w)}{\partial w_k} = p \operatorname{trace}\Big(M^{(k)} \quad {M_F(w)}^{p-1}\Big)$$
$$\frac{\partial^2 \varphi_p(w)}{\partial w_i \partial w_j} = \operatorname{trace}\Big(M^{(i)} \quad Q\big(\Delta^{[2]} \odot (Q^T M^{(j)} Q)\big) Q^T\Big),$$

where $\odot$ denotes the elementwise (Hadamard) product of matrices, $M^{(k)} = {A^{(k)}}^T A^{(k)}$ and $\Delta^{[2]}$ is the matrix of second divided differences, defined by

$$\Delta^{[2]}_{i,j} = \begin{cases} p(p-1)\lambda_i^{p-2} & \text{if} \quad \lambda_i = \lambda_j \\ p\frac{\lambda_i^{p-1} - \lambda_j^{p-1}}{\lambda_i - \lambda_j} & \text{otherwise.} \end{cases}$$

This method, which can be seen as a generalization of Newton's algorithm, has the advantage of converging rapidly. On networks of a reasonable size, this algorithm converge within 5 iterations. However, it is known that Newton's iterations exhibit a local convergence behavior only, and there is no theoretical guarantee for this algorithm to converge. Moreover, the computation of the Hessian becomes impracticable when the network is large.

For this reason, we propose a projected gradient algorithm which takes advantage of the sparsity of the observation matrices. Once the spectral decomposition of $M_F(w)$ is achieved, we need the matrix ${M_F(w)}^{p-1} = Q^T D^{p-1} Q$ to compute the gradient. Each entry of this matrix requires $\mathcal{O}(m)$ multiplications. Denote by $\mathcal{J}$ the set of indices for which a there is an observation matrix with a nonzero entry :

$$\mathcal{J} = \{(i,j) \mid \exists k : M^{(k)}_{i,j} \neq 0\}.$$

As each of the $M^{(k)}$ is very sparse (not more than a few nonzero entries on each row), $|\mathcal{J}|$ is in the order of m (rather than m^2). Therefore, computing the gradient with

$$\frac{\partial \varphi_p(w)}{\partial w_k} = p \sum_{(i,j)\in\mathcal{J}} {M_F(w)}^{p-1}_{i,j} M^{(k)}_{i,j}$$

requires $\mathcal{O}(m^2)$ multiplications instead of $\mathcal{O}(m^3)$.

At each iteration, we perform a step of length δ from the current position $w^{(i)}$ in the direction of the gradient: $w_g = w^{(i)} + \delta \nabla\varphi(w^{(i)})$, and the new value of w is obtained by projecting w_g on $\mathcal{F}$:

$$w^{(i+1)} = \arg\min_{w\in\mathcal{F}} \|w_g - w\|^2. \tag{23}$$

The bottleneck of this algorithm is currently the spectral decomposition. For future work, we hope to overcome it by using sparse SVD routines.

A natural idea to approximate the integer solution of Problem (11) is to round the continuous solution $w^* \in [0,1]^s$ provided by (21) to an integer vector $\tilde{w} \in \{0,1\}^s$. We propose a special rounding scheme : First, order the interfaces according to the (real) values of w_k. Then, select a subset of *good* interfaces whose total cost is larger than the budget B. Finally, try each combination of interfaces among this subset which satisfies the budget constraint. In the unit-cost case, where one looks for a design with no more than n selected interfaces, this rounding scheme is equivalent to choosing the subset with the $n+3$ or $n+4$ best interfaces (ordered according to the solution of the continuous relaxation), and to try each $n-uple$ of interfaces among this subset.
Other rounding schemes exist, as the randomized rounding, where $\tilde{w}_i$ is set to 1 with probability w_i^* and to 0 with probability $1-w_i^*$. The efficiency of these rounding schemes is still an open question, but the results presented in next section are very encouraging.

4.3 A semidefinite relaxation for the *Minimal exhaustive deployment of Netflow*

The constraint of problem (15) that the design w should bring at least a certain amount of information $\gamma > 0$ takes a lot of sense when $p \leq 0$, because it allows one to look for full-rank designs (i.e. designs for which the observation matrix $A(w)$ is of full rank). The case $p = -\infty$ is an interesting one, because setting the constraint $\Phi_{-\infty} > \epsilon$ means that all the eigenvalues of $M_F(w)$ are greater than ε. This can be formulated with the help of a linear matrix inequality (LMI) :

$$M_F(w) \succeq \varepsilon \mathrm{I}.$$

We can use techniques of semidefinite programming, along the lines of Goemmans and Williamson [9], in order to formulate the integer constraint in the form of a linear matrix equality : the constraint $w \in \{0,1\}$ is equivalent to $w = w^2$. Introducing the new semidefinite variable

$$W = [w^T, 1]^T \cdot [w^T, 1] = \left(\begin{array}{c|c} ww^T & w \\ \hline w^T & 1 \end{array}\right),$$

this means that the diagonal of W equals its last column: $\mathrm{diag}(W) = W \cdot [0,...,0,1]^T$. In order to be able to associate an SDP variable W with a design w, we must also make sure that W is of rank 1, and that $W_{s+1,s+1} = 1$. Problem (15) is equivalent to the program :

$$\begin{aligned} \min_{W} \quad & \mathrm{trace}(W \cdot \mathrm{diag}(c)) \\ s.t. \quad & \mathrm{diag}(W) = W \cdot [0,...,0,1]^T; \\ & W_{s+1,s+1} = 1; \\ & \mathrm{rank(W)} = 1; \\ & W \succeq 0; \\ & A^T A + \sum_{k=1}^{s} W_{k,1} A^{(k)^T} A^{(k)} \succeq \varepsilon \mathrm{I}. \end{aligned} \tag{24}$$

Dropping the rank-1 constraint, we obtain a semidefinite program. The vector w that we find with the best rank-1 approximation of the solution W^* of the relaxed problem has no reason to be a $0/1$ vector. We used a rounding scheme in order to construct a binary solution w^*, for which $M_F(w^*)$ is of full rank : we can order the interfaces according to the value of w_k, and add sequentially the interfaces in this order, until a design of full rank is found. In order to find the optimal integer design, we also used the above semidefinite program to trim a *Branch and Bound* tree.

4.4 Integer programming metaheuristics

For comparison's sake, we also investigated the use of metaheuristics, such as neighborhood descents or tabu search. These techniques are briefly described here.
Fedorov proposed in 1972 a descent heuristic to compute optimal designs [7] in the unit-cost case, which we will call EXCHANGE: Starting with an arbitrary set of n interfaces $\mathcal{I}^0$, we make at each stage of the algorithm the best possible substitution of an interface of $\mathcal{I}^0$ versus an inactive interface. The algorithm stops when no exchange can increase the value of $\varphi_p(w)$. We can slightly change this algorithm to a Variable Neighborhood Search (VNS) by first selecting greedily k new interfaces, and then deselecting k interfaces at each step of the algorithm.

Another variant is the tabu search, where we keep in memory the interfaces which do not increase significantly the value of $\varphi_p(w)$. At the next iteration, we will not investigate these interfaces. This tabu search can be useful when the network (and hence m) is large, because the evaluation of $\varphi_p(w)$ requires a costly $m \times m$ matrix diagonalization.

5. EXPERIMENTAL RESULTS

We ran computations on the Abilene backbone, which is made of 11 nodes (routers), 28 links (each link on the figure is bidirectional), and 110 OD-pairs, and on the much larger France Telecom Opentransit backbone (100 nodes, 267 links and 5646 OD-pairs). For both networks, we considered several instances of problem (11),

Criterion	Algorithm	Interfaces found	value of the criterion	rank	CPU
$\varphi_{0.05}$	SQP	[non integer]	106.454635	110	2.303
	Rounding	[7 11 13 15 19]	71.249755	60	2.819
	Greedy	[7 11 13 15 19]	71.249755	60	1.381
	Exchange	[7 11 13 15 19]	71.249755	60	1.107
	VNS	[7 11 13 15 19]	71.249755	60	3.108
$\varphi_{0.20}$	SQP	[non integer]	102.320021	110	2.290
	Rounding	[7 11 13 15 19]	73.050126	60	2.794
	Greedy	[7 11 13 15 19]	73.050126	60	1.342
	Exchange	[7 11 13 15 19]	73.050126	60	1.209
	VNS	[7 11 13 15 19]	73.050126	60	2.871
$\varphi_{0.50}$	SQP	[non integer]	125.332464	110	2.275
	Rounding	[11 13 15 17 19]	115.073658	59	2.761
	Greedy	[11 13 15 17 19]	115.073658	59	1.346
	Exchange	[11 13 15 17 19]	115.073658	59	1.186
	VNS	[11 13 15 17 19]	115.073658	59	3.128
$\varphi_{1.00}$	Linear Programming	[non integer]	369.000000	52	0.006
	Rounding	[4 11 13 14 17]	369.000000	52	0.602
	Greedy	[4 11 13 14 17]	369.000000	52	1.430
	Exchange	[4 11 13 14 17]	369.000000	52	1.351
	VNS	[4 11 13 14 17]	369.000000	52	3.246
CRAMER RAO [2]	Greedy	[7 11 13 15 19]	81.263014	60	1.628
	Exchange	[7 11 13 15 19]	81.263014	60	7.419
	VNS	[7 11 13 15 19]	81.263014	60	6.150
RANK	Greedy	[7 11 13 15 19]	60.000000	60	2.022
	Exchange	[7 11 13 15 19]	60.000000	60	6.492
	VNS	[7 11 13 15 19]	60.000000	60	3.912

Table 1: Computation results for the placement of 5 Netflow measurements on the Abilene backbone

depending on the value of p, the assumption made on the information that one can retrieve from Netflow measurements, and the type of localization allowed for Netflow. We also show by an example the robustness of our approach when a modification of the routing occurs, and we give experimental results for the *minimal exhaustive deployment of Netflow* (15).

5.1 Abilene backbone

Activation of 5 *interfaces.*

We solved the problem of optimal deployment in the unit-cost case, looking for a design with no more than 5 selected interfaces. This was done under the assumption that Netflow reports allow one to find the outgress node of each packet (the router where the packet leaves the Abilene backbone), but not the ingress node. The algorithms used for the comparison are the sequential quadratic programming (SQP) associated with a search among the *best* interfaces (ROUNDING), and the metaheuristics described in previous section (EXCHANGE AND VNS). For a complete comparison of the results, we also ran the GREEDY algorithm.

On the other hand, several criteria are considered : the Schatten-p like functions φ_p for different values of p, as well as the rank criterion (Problem (12)) and the trace of the Cramer-Rao bound considered in the work of Bermolen, Vaton and Juva [2]. The computation results are displayed in Table 1.

The subset of 5 interfaces [7,11,13,15,19] (namely Seattle $\rightarrow$ Denver, Sunnyvale $\rightarrow$ Los Angeles, Los Angeles $\rightarrow$ Houston, Denver $\rightarrow$ Kansas City, Chicago $\rightarrow$ Indianapolis) maximizes the criteria $\varphi_{0.05}$ and $\varphi_{0.2}$, but also the Cramer-Rao criterion and the rank of the observation matrix. This *robust* placement of 5 interfaces on Abilene Network is, to our sense, the best possible solution, and it is represented on Figure 1. For larger values of p, another optimal design is found. But as mentioned earlier, these designs might lead to redundant measures.

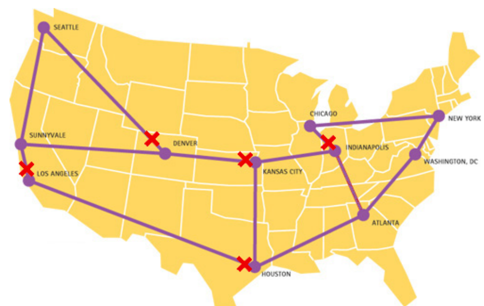


Figure 1: Best placement for 5 interfaces

We have enumerated all the combinations of 5 interfaces, and we found that our rounding scheme of the continuous solution returned the optimal solution for every criterion. These results show that our approach of rounding the solution of the continuous relaxation

provides the same designs as the other studied algorithms. This, however, will not be the case in the next example, where we want to activate Netflow simultaneously on all the incoming interfaces of a router.
We can also show with this instance that the optimal design actually depends on the presence of SNMP data, which is one of the reason which has motivated our experimental-design-based approach. For several values of p in the range $[0.05, 0.2]$, Table 2 shows indeed that, whether one considers the SNMP data or not, the best placement of Netflow is different.

Criterion	SNMP data	optimal design
$\varphi_{0.05}$	yes	[7 11 13 15 19]
	no	[11 13 15 17 19]
$\varphi_{0.1}$	yes	[7 11 13 15 19]
	no	[11 13 15 17 19]
$\varphi_{0.2}$	yes	[7 11 13 15 19]
	no	[11 13 15 17 19]

Table 2: *Netflow Optimal Deployment*, with and without the presence of SNMP data

Another interesting experiment is to study the robustness of the optimal placement when a modification of the routing matrix occurs, which is often the case in practice. To this end, we computed the optimal deployment of Netflow on 5 interfaces of Abilene Backbone for several routing matrices. For each origin-destination pair OD, the route was set at random either to the shortest path from O to D, or to the second shortest path, or could be split between these two paths (half of the traffic from O to D takes the shortest path, half of the traffic takes the second shortest). We computed the optimal deployment with the value $p = 0.2$, for 100 routing matrices generated randomly. The results are represented in Table 2: over the 100 optimal designs computed, we indicate the links which have been selected at least once, as well as the number of times where they have been selected. This experiment shows that the interfaces that we suggest to select (7,11,13,15 and 19), are robust to modifications of routing. If we are aware of the variations of routing which are likely to occur, this kind of simulation can suggest to select some more robust interfaces, even if they are not optimal for the current routing matrix (here, we could select the 5 most robust interfaces 7,13,15,16 and 19).

Activation of the incoming interfaces of 4 *routers.*
Under the assumption that the costs of deployment of Netflow are paid only once per router where it is installed (activating Netflow on 5 interfaces of a router is not more expensive than activating Netflow on a single interface of this router), we tried to find the optimal set of 4 routers where Netflow should be activated simultaneously on all incoming interfaces. This time, we

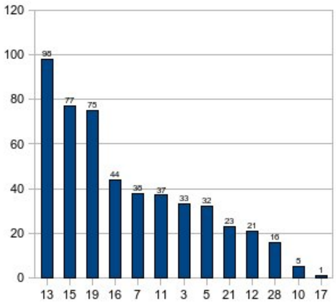


Figure 2: Most selected interfaces of Abilene (labeled on the X-axis) over 100 random routings

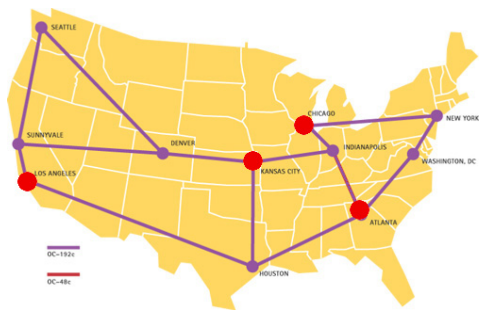


Figure 3: Best placement on 4 routers

assumed that Netflow would be able to find both the ingress and the outgress router for each packet. For this problem, the deployment which maximizes the criteria $\varphi_{0.05}, \varphi_{0.01}$, RANK and CRAMER-RAO is the activation of Netflow on the routers $1, 2, 6$ and 7, namely Atlanta, Chicago, Los Angeles and Kansas City. This deployment shows a large robustness with respect to the different criteria again, and is depicted on Figure 3. We summarized the computation results on Table 4. This time, we incorporated the enumeration of all combinations in the table of results, because it could be done in the same order of time as other descent techniques : The value of the criterion in the rows $ENUM$ are always the true optimal.
Interestingly, for $p = 0.05$, our rounding procedure was the only one which found the optimal solution. This is another example where the greedy scheme is suboptimal. However, we can notice that it approximates the optimal solution within an excellent factor (much better than the theoretical $1 - \frac{1}{e}$).

Minimal exhaustive Deployment of Netflow.
We also solved the problem (15) for the Abilene backbone. We ran the computations with different procedures : A rounding of the solution provided by the SDP relaxation (24), a Branch&Bound procedure applied to the same SDP, and a greedy procedure which selects

Value of p	Continuous solution	Number of Iterations
0.5	$w_6 = 0.644$; $w_{11} = 1$; $w_{49} = 0.305$; $w_{63} = 0.475$; $w_{73} = 1$; $w_{74} = 0.576$	27
0.2	$w_6 = 0.530$; $w_7 = 0.084$; $w_{11} = 1$; $w_{32} = 0.059$; $w_{48} = 0.053$; $w_{49} = 0.375$ $w_{63} = 0.477$; $w_{73} = 1$; $w_{74} = 0.422$	65
0.05	$w_5 = 0.099$; $w_6 = 0.433$; $w_7 = 0.131$; $w_{11} = 0.920$; $w_{32} = 0.097$; $w_{48} = 0.103$ $w_{49} = 0.379$; $w_{54} = 0.062$; $w_{63} = 0.422$; $w_{72} = 0.084$; $w_{73} = 0.872$; $w_{74} = 0.398$	87

Table 3: computation results for the *Netflow Optimal Deployment* on Openteransit

.

Criterion	Algorithm	Routers found	value of the criterion	rank	CPU
$\varphi_{0.05}$	SQP	[non integer]	112.616005	110	1.111
	Enum	[1 2 6 7]	108.477557	105	2.020
	Rounding	[1 2 6 7]	108.477557	105	1.316
	Greedy	[2 4 6 10]	106.263333	102	0.302
	Exchange	[5 6 7 11]	107.297467	103	0.376
	VNS	[3 4 5 8]	106.926601	103	0.598
$\varphi_{0.20}$	SQP	[non integer]	124.464483	110	1.079
	Enum	[1 2 6 7]	118.634057	105	2.242
	Rounding	[1 2 6 7]	118.634057	105	1.280
	Greedy	[4 5 6 8]	118.210746	101	0.240
	Exchange	[5 6 7 11]	118.600239	103	0.320
	VNS	[1 2 6 7]	118.634057	105	0.478
$\varphi_{0.50}$	SQP	[non integer]	170.944263	110	1.137
	Enum	[4 5 6 7]	168.041160	99	1.897
	Rounding	[4 5 6 7]	168.041160	99	1.340
	Greedy	[4 5 6 7]	168.041160	99	0.236
	Exchange	[4 5 6 7]	168.041160	99	0.163
	VNS	[4 5 6 7]	168.041160	99	0.626
$\varphi_{1.00}$	Linear Programming	[non integer]	426.000000	98	0.048
	Enum	[4 5 6 7]	426.000000	99	1.923
	Rounding	[4 5 6 7]	426.000000	99	0.259
	Greedy	[4 5 6 7]	426.000000	99	0.234
	Exchange	[4 5 6 7]	426.000000	99	0.319
	VNS	[4 5 6 7]	426.000000	99	0.692
RANK	Enum	[1 2 6 7]	105.000000	105	3.173
	Greedy	[2 4 6 10]	102.000000	102	0.484
	Exchange	[1 2 6 7]	105.000000	105	1.206
	VNS	[3 4 5 11]	103.000000	103	0.910
CRAMER RAO [2]	Enum	[1 2 6 7]	155.906250	105	20.941
	Greedy	[2 4 6 10]	150.729167	102	2.269
	Exchange	[1 2 6 7]	155.906250	105	8.850
	VNS	[4 5 10 11]	152.516250	103	8.752

Table 4: computation results for the deployment of Netflow on 4 routers of Abilene backbone

Figure 5: minimal Deployment with exhaustivity constraint

the interface which maximizes the rank of the observation matrix at each iteration, until a full rank matrix is obtained. All these techniques provided a subset of 14 interfaces where Netflow has to be activated if we want to observe each flow. We compared the running time of each technique in Table 5 : The Branch&Bound technique takes naturally much more time than the other procedures, but gives a certificate on the optimality of the solution. The solution found with the greedy procedure is of same cardinality as the optimal solution (14 interfaces), but its smallest eigenvalue is smaller than the one provided with the rounding procedure of the SDP, which indicates a greater noise sensibility. This minimal deployment is depicted on Figure 5. It is important to notice that with only 14 interfaces over 28, the rank of the observation matrix is full, which means that we are able to infer each OD flow up to the sensibility of the Netflow measurements.

Algorithm	CPU	$\lambda_{\min}(M_F(w))$
Branch&Bound	1504.208	0.176676
SDP - Rounding	3.934	0.176676
Greedy	5.118	0.170402

Table 5: computation results for *minimal exhaustive deployment of Netflow* on Abilene backbone

6. FRANCE TELECOM OPENTRANSIT BACKBONE

The Opentransit backbone is much more challenging because of its size ($m = 5646$ OD pairs). As mentioned earlier, the bottleneck of our approach is the computation of the spectral decomposition of the $m \times m$ matrix $M_F(w)$, which takes approximatly 15 minutes on a PC at 2.4 GHz. This diagonalization step is needed both for the computation of the gradient and the value of the criterion $\varphi_p(w)$. We ran the projected gradient algorithm for several values of p, looking for an optimal deployment of Netflow on 4 routers of the network (in the unit-cost case), and stopped the computation when the gradient was (almost) orthogonal to the feasible set at interior points. Interestingly, the algorithm converged to continuous solutions with a very small support, as shown in Table 3.
In order to solve the combinatorial problem, we applied our ROUNDING scheme, trying all $4-$uple of interfaces among the support of the continuous relaxation. For comparison's sake, we have also implemented a TABU search, which converged each time to the same design as the one provided by the rounding scheme. We also give the N*etflow optimal deployment* for $p = 1$, which was easily found by linear programming (Knapsack instance with unit costs). These results are presented in Table 6. We can make the same observation that was done for Abilene : the same design maximizes $\varphi_{0.2}$ and $\varphi_{0.05}$. Moreover, it also maximizes the rank of the Fisher Information matrix, which shows once more the robustness of our approach. This design (nodes $6, 11, 49$ and 73) is depicted on Figure 4, where the nodes which support the continuous solution have been circled in red. It is interesting to remark that, corresponding to intuition, all these routers are located at strategic points of the networks where many flows intersect.

7. CONCLUSIONS AND PERSPECTIVES

Based on experimental design, this paper proposes a new approach to find an optimal deployment of Netflow on large IP networks in order to estimate traffic matrices. We have formulated a combinatorial problem, which we have proven to be $1 - 1/e$-approximable in polynomial time. Our experiment indicates that activating Netflow only on a subset of nodes is enough to obtain a complete information on the traffic. This model, which only assumes that the observations are linear combinations of the traffic on the different routes, is flexible enough to allow one to deal with several variants of the optimization of measurement problem which occur in practice. For example, this approach could be used to design experiments which measure the Quality of Service in a network.
Note that the rates of sampling of Netflow arise as an input in our model, since they determine the covariance matrix of the noise of observation. Hence, our approach, which focuses on the purely combinatorial problem consisting in choosing the set of interfaces on which Netflow is deployed, may be seen as complementary to the computation of the optimal sampling rates which is performed in [5], and it may be combined with it, which is the object of a further work.

Criterion	Algorithm	Routers found	Rank
φ_1	LP	[6 11 73 74]	4640
$\varphi_{0.50}$	ROUNDING	[6 11 63 73]	4856
	TABU	[6 11 63 73]	110
$\varphi_{0.20}$	ROUNDING	[6 11 49 73]	4923
	TABU	[6 11 49 73]	4923
$\varphi_{0.005}$	ROUNDING	[6 11 49 73]	4923
	TABU	[6 11 49 73]	4923
RANK	TABU	[6 11 49 73]	4923
	GREEDY	[5 11 63 73]	4877

Table 6: Optimal integer designs for the placement of Netflow on 4 routers of Opentransit

8. REFERENCES

[1] Netflow performance analysis, CISCO technical white paper.

[2] P. Bermolen, S. Vaton, and I. Juva. Search for optimality in traffic matrix estimation : a rational approach by Cramer-Rao lower bounds. In *NGI'06*, pages 224 – 231, 2006.

[3] M. Bouhtou and O. Klopfenstein. Robust optimization for selecting netflow points of measurement in an IP network. In *Proceedings of the Global Telecommunications Conference (GLOBECOM '07)*, pages 2581–2585. IEEE, 2007.

[4] J. Brinkhuis, Z.-Q. Luo, and S. Zhang. Matrix convex functions with applications to weighted centers for semidefinite programming. Technical report, SEEM, The Chinese University of Hong Kong, 2005.

[5] G. Cantieni, G. Iannaccone, C. Barakat, Ch. Diot, and P. Thiran. Reformulating the monitor placement problem: Optimal network-wide sampling. In *CISS*, 2006.

[6] J. Cao, D. Davis, S. Wiel, and B. Yu.

Time-varying network tomography : Router link data. In *Inf. Theory, 2000.*
[7] V.V. Fedorov. *Theory of optimal experiments.* New York : Academic Press, 1972.
[8] U. Feige. A threshold of ln n for approximating set cover. *Journal of ACM*, 45(4):634–652, July 1998.
[9] M.X. Goemans. Semidefinite programming in combinatorial optimization. *Math. Programming*, 79:143–161, 1997.
[10] P.R. Goundan and A.S. Schulz. Revisiting the greedy approach to submodular set function maximization. Working paper, July 2007.
[11] E. Jorswieck and H. Boche. *Majorization and matrix-monotone functions in wireless communications.* Now Publishers Inc., 2006.
[12] A. Krause and C. Guestrin. A note on the budgeted maximization of submodular functions. Technical report, Carnegie Mellon University, Pittsburgh, 2005.
[13] G. Liang, N. Taft, and B. Yu. A fast lightweight approach to origin-destination IP traffic estimation using partial measurements. *IEEE/ACM Trans. Netw.*, 14(SI):2634–2648, 2006.
[14] G. Liang, B. Yu, and N. Taft. Maximum entropy models: Convergence rates and applications in dynamic system monitoring. *Proc. Int. Symp. Information Theory*, page 168, 2005.
[15] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian. Traffic matrix estimation: Existing techniques compared and new directions. In *Proc. of SIGCOMM*, Pittsburgh, Aug. 2002.
[16] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.
[17] N.Gaffke and B.Heiligers. *S.H. Park, G.G. Vining (eds): Statistical Process Monitoring and Optimization*, chapter Optimal approximate designs for B-spline regression with multiple knots, pages 339–358. New York, 2000.
[18] F. Pukelsheim. *Optimal Design of Experiments.* Number 50 in Classics in Applied Mathematics. SIAM, Philadelphia, PA, USA, 2006.
[19] J. Vondrák, G. Calinescu, C. Chekuri, and M. Pál. Maximizing a submodular set function subject to a matroid constraint. In *12th ICIPCO*, volume 4513, pages 182–196, 2007.
[20] H. Zang and A. Nucci. Optimal netflow deployment in IP networks. In *19th International Teletraffic Congress (ITC)*, Beijing, China, August 2005.
[21] X. Zhan. *Matrix Inequalities (Lecture Notes in Mathematics)*. Springer, 2002.
[22] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *SIGMETRICS '03*, pages 206–217, New York, NY, USA, 2003.

APPENDIX

Proof of Lemma 3.2

The inequality (20) becomes an equality when $p = 1$ by linearity of the trace.
Since the eigenvalues of a matrix are continuous functions of its entries, and since the set of positive definite matrices $\mathbb{S}_m^{++}$ is dense in the set of positive semi-definite matrices $\mathbb{S}_m^{+}$, it suffices to establish the inequality when X,Y,Z are positive definite. Let us consider the map:

$$\begin{aligned} \psi : \mathbb{S}_m^{+} &\longrightarrow \mathbb{R} \\ T &\longmapsto \text{trace } (X+T)^p - \text{trace } T^p. \end{aligned}$$

The inequality to be proved can be rewritten as

$$\psi(Y+Z) \leq \psi(Z).$$

We will prove this by showing that ψ is nonincreasing with respect to Loewner ordering in the direction generated by any positive semidefinite matrix. To this end, we compute the Frechet derivative of ψ at $T \in \mathbb{S}_m^{++}$ in the direction of an arbitrary matrix $H \in \mathbb{S}_m^{+}$. By definition,

$$D\psi(T)[H] = \lim_{\epsilon \to 0} \frac{1}{\epsilon}\big(\psi(T+\epsilon H) - \psi(T)\big).$$

When f is an analytic function, $X \longmapsto \text{trace } f(X)$ is Frechet-differentiable, and an explicit form for the derivative is known [11]: $D\big(\text{trace } f(A)\big)[B] = \text{trace}\big(f'(A)B\big)$. As the map $z \longrightarrow z^p$ is analytic at all point of the positive real axis for $p \in]0,1[$, provided that the matrix T is positive definite (and hence $X+T$), we have

$$D\psi(T)[H] = p \text{ trace}\Big(\ \big((X+T)^{p-1} - T^{p-1}\big)H\Big).$$

Since $0 < 1-p < 1$, the Loewner-Heinz inequality (Theorem 1.1 in [21]) yields $(X+T)^{1-p} \succeq T^{1-p}$. By antitonicity of the matrix inversion we infer that the matrix $W = T^{p-1} - (X+T)^{p-1}$ is positive semidefinite.

$$\begin{aligned} D\psi(T)[H] &= -p \text{ trace } (WH) \\ &= -p \text{ trace } (W^{1/2}HW^{1/2}) \leq 0 \end{aligned}$$

Consider now $f(s) = \psi(sY+Z)$. For all $s \in [0,1]$, we have

$$f'(s) = D\psi(sY+Z)[Y] \leq 0,$$

and so, $f(1) = \psi(Y+Z) \leq f(0) = \psi(Z)$, from which the desired inequality follows.
Finally, the inequality in the case $p = 0$ is obtained by letting $p \to 0$. $\square$

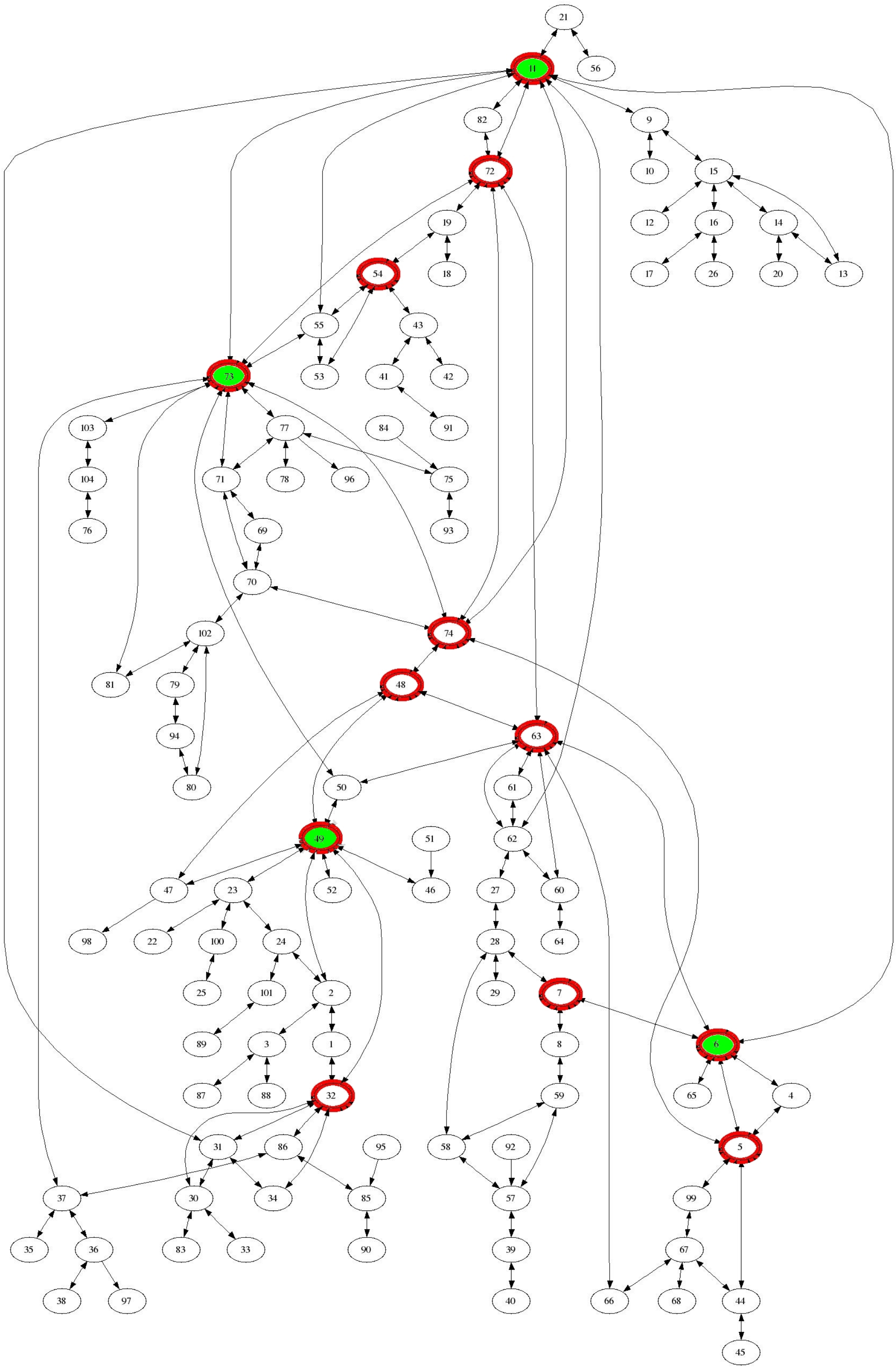


Figure 4: Optimal Deployment (filled nodes) and support of the continuous selection (circled nodes) for the deployment of Netflow on 4 routers of Opentransit