Optimal monitoring in large networks by Successive **c**-Optimal Designs

Guillaume Sagnol, Stéphane Gaubert
INRIA Saclay & CMAP, Ecole Polytechnique
Email: {guillaume.sagnol,stephane.gaubert}@inria.fr

Mustapha Bouhtou
Orange Labs
Email: mustapha.bouhtou@orange-ftgroup.com

***Abstract*—We address the problem of optimizing the use of Network monitoring tools, such as Netflow, on a large IP network. We formulate a convex optimization problem which allows one to handle, in a unified framework, the combinatorial problem of selecting the "best" set of interfaces on which Netflow should be activated, and the problem of finding the optimal sampling rates of the network-monitoring tool on these interfaces, when the aim is to infer the traffic on each internal Origin-Destination (OD) pair. We develop a new method, called "Successive c-optimal Design", which is much faster than the classical ones. It reduces to solving a stochastic sequence of Second Order Cone Programs. We give experimental results relying on real data from a commercial network, which show that our approach can be used to solve instances that were previously intractable, and we compare our method with previously proposed ones.**

I. Introduction

A. Background

The estimation of Origin-Destination (OD) traffic matrices for backbone networks is a crucial problem for Internet providers which has attracted much interest from the network research community [1], [2], 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 n nodes (routers) and l edges (links). Link measurements are provided by the Simple Network Management Protocol (SNMP), which counts the number of bytes seen on each link in a given time window. We use boldface letters to denote (column) vectors. The vector of SNMP link counts is denoted by $\boldsymbol{y}^{\mathrm{SNMP}}$. We are also given the routing matrix A of the network, which is the incidence matrix between the $m = n^2$ OD pairs and the links. The matrix A is of size $l \times m$, and its (e, r)-entry represents the fraction of the traffic of the OD pair r that traverses link e. The unknown in our problem is the vector of OD flows $\boldsymbol{x} = (x_1, \ldots, x_m)^T$, where x_r is the traffic of the OD pair r during the observation period. The following relation is easily seen to hold:

$$\boldsymbol{y}^{\mathrm{SNMP}} = A\boldsymbol{x}.$$

In typical networks, we have $l \ll m$, and so the estimation of $\boldsymbol{x}$ based on the link counts $\boldsymbol{y}^{\mathrm{SNMP}}$ is an ill-posed problem.

B. Optimization of the measurement

A way to introduce new constraints is to use a network-monitoring tool such as Netflow (Cisco systems). This was considered by Liang, Taft and Yu [1], who proposed a scheme for selecting dynamically the flows to be measured by Netflow, in order to improve the accuracy of the traffic estimation. Of course, activating Netflow everywhere on the network yields an extensive knowledge of the OD flows. According to [3] however, activating Netflow on an interface of a router causes its CPU load to increase by 10 to 50%. It is thus of great interest to optimize the use of this tool. It is now possible to use a sampled version of Netflow, which substantially decreases the CPU utilization needed to handle Netflow packets. On the other hand, the lower the sampling rates are, the less accurate are the Netflow measurements. The problem is thus both to decide where to activate Netflow, and at which sampling rate.

Most operators collect Netflow information for multiple purposes, such as security or billing, not only for estimating the traffic. However, we believe that the present approach, which addresses the latter goal, might also be of some interest for other purposes, since it indicates which routers or interfaces meet a maximal proportion of the traffic. Moreover, we will see that this approach leads to a nice mathematical formulation and to scalable algorithms.

When Netflow is activated on an interface of the network, it analyzes the headers of the packets traversing this interface. As a result we obtain some statistics, such as the source and destination IP addresses of these packets. However, we are not trying to infer the global path of the packets from IP source to IP destination, but only the part of their path which is inside the network of interest, like the backbone of an autonomous system (AS). In the sequel, we will use the terms *internal source* and *internal destination* to refer to the ingress and egress routers of a packet within the backbone of interest.

Practically, we will assume throughout this paper that when Netflow performs a measurement on the k^{th} interface, we are able to break out the flows traversing this interface according to their internal destination. This results in a multidimensional observation $\boldsymbol{y_k}$, whose entry d is the sum of flows traversing k and having the destination d. The model is linear:

$$\boldsymbol{y_k} = A_k\boldsymbol{x} \ . \tag{1}$$

Note that this assumption is more general and more realistic than the one made in [4], where the authors assume that when the monitoring tool analyzes a packet, it is able to find both its internal source and destination. In practice, one can find the internal destination of a packet by simulating the path toward its ultimate destination with the forwarding tables of

the routers, but finding the internal source of a packet is a challenging issue. The main difference with the simplified model considered in [4] is that the information matrices $A_k^T A_k$ are not diagonal anymore, which makes the problem much harder computationally.

A precise description of the classical methodology used to infer the origin-destination traffic from Netflow measurements is made in [5]. It is common to activate Netflow only on ingress links of a backbone in order to cope with the uncertainty on the internal source of the packets. In this paper, we show that other deployment strategies can be useful.

C. Related Work

Many authors from the network research community [6], [7], [8], [9], [4] investigated the placement of Netflow. Recently, Song, Qiu and Zhang [6] used classical criteria from the theory of experimental design to choose a subset of interfaces where Netflow should be activated, and developed an efficient greedy algorithm to find a near optimal solution to this combinatorial problem. In a recent work, we show indeed [8] that the greedy algorithm always finds a solution within $1 - 1/e \simeq 62\%$ of the optimum.

Singhal and Michailidis [4] considered a state-space model representing the evolution of the traffic matrix over time, in which the estimation of the traffic can be done by a Kalman filter. They successfully applied the experimental design theory to formulate the problem of finding the sampling rates that minimize the covariance matrix of the Kalman filter as a Semidefinite Program (SDP). Since the covariance matrix is computed recursively in the filtering process, it contains information on the past measurements, and computing new sampling rates at each time step makes the estimation more and more accurate.

The main contribution of this paper is an alternative to the greedy algorithm of [6]. The approach that we develop here (Section III-C), which we call "Successive $\boldsymbol{c}-$Optimal Designs" (SCOD), can efficiently be applied to very large networks, and we show that it allows one to infer the traffic more precisely. The idea is to replace the classical "$A-$optimal design" problem, in which one minimizes the sum of the inverses of the eigenvalues of the information matrix, by a sequence of optimal design problems with a scalar functional. Each of these scalar problems is solved to optimality by means of a reduction to a moderate size second order programming problem, which can be solved rapidly (and with a limited memory requirement) by interior point methods. Whereas SDP approaches, like the one of [4], implemented with state of the art solvers, are typically limited to a few hundreds of OD pairs, the present approach allowed us to solve instances from a commercial network with 436 interfaces and 13456 OD pairs.

The rest of this paper is organized as follows: the problem and the experimental design background are presented in Section II. Next, we discuss previous methods to solve this problem, in particular the "Netquest" greedy approach of Song, Qiu and Zhang [6], as well as the optimal design problem in a state-space model studied by Singhal and Michailidis [4]. The Successive $\boldsymbol{c}-$Optimal Designs approach (SCOD), which is the main contribution of this manuscript, is presented in Section III-C. Finally, we give some experimental results in Section IV, showing the usefulness of our approach for both the discrete problem (Netflow deployment) and the continuous one (Netflow optimal sampling).

II. Experimental design background

A. Netflow optimal deployment

Let $\mathcal{I} = \{1, \ldots, s\}$ be the set of all interfaces on which Netflow can be activated. We start with the discrete problem, in which the operator wants to choose a subset of these interfaces for the Netflow measurements. Note that this problem is also meaningful when a network is not yet or is only partially instrumented with routers supporting Netflow, and when the Internet provider wants to equip a number of additional routers with a network-monitoring tool.

We denote by $\mathcal{I}^{\mathrm{a}}$ the set of interfaces on which Netflow is activated. The measurement vector $\boldsymbol{y}$ is now the concatenation of the SNMP data $\boldsymbol{y}^{\mathrm{SNMP}}$ with all the Netflow measurements $(\boldsymbol{y}_k)_{k \in \mathcal{I}^{\mathrm{a}}}$. We define the *design* variable $\boldsymbol{w}$ as the $0/1$ vector of size s, where w_k equals 1 if and only if $k \in \mathcal{I}^{\mathrm{a}}$. The measurements are never exact in practice, and so we have to deal with a noise ϵ, which is a result, among other things, of lost packets, misalignment of SNMP polling intervals, and Netflow sampling. This can be modeled as follows:

$$\boldsymbol{y} = A_{\boldsymbol{w}}\ \boldsymbol{x} + \boldsymbol{\epsilon}, \tag{2}$$

$$\text{where } \boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}^{\mathrm{SNMP}} \\ \boldsymbol{y}_{k_1} \\ \vdots \\ \boldsymbol{y}_{k_n} \end{pmatrix} \text{ and } A_{\boldsymbol{w}} := \begin{bmatrix} A \\ A_{k_1} \\ \vdots \\ A_{k_n} \end{bmatrix}.$$

Now, assume that we have enough measurements, so that $A_{\boldsymbol{w}}$ is of full rank, and assume that the noises on the observations are mutually independent, that is to say that the covariance matrix $\Sigma = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]$ is known and only has diagonal entries ($\mathbb{E}[\cdot]$ denotes the expectation of a random variable). The best linear unbiased estimator of $\boldsymbol{x}$ is given by a pseudo inverse formula (Gauss Markov Theorem). Its variance is given below:

$$\hat{\boldsymbol{x}} = \left(A_{\boldsymbol{w}}^T \Sigma^{-1} A_{\boldsymbol{w}}\right)^{-1} A_{\boldsymbol{w}}^T \Sigma^{-1} \boldsymbol{y}. \tag{3}$$

$$\mathrm{Var}(\hat{\boldsymbol{x}}) = (A_{\boldsymbol{w}}^T \Sigma^{-1} A_{\boldsymbol{w}})^{-1}. \tag{4}$$

To simplify the notation, we will assume that $\Sigma = I$ (one may always reduce to this case with a left diagonal scaling by $\Sigma^{-1/2}$ of $\boldsymbol{y}$, $A_{\boldsymbol{w}}$ and $\boldsymbol{\epsilon}$). The inverse of the variance of the estimator $\hat{\boldsymbol{x}}$ is called *information matrix*, and we denote it by $M_F(\boldsymbol{w})$. Note that it can be decomposed as the sum of the information matrices $A_k^T A_k$ over the interfaces k on which Netflow is activated:

$$M_F(\boldsymbol{w}) = (\mathrm{Var}\,\hat{\boldsymbol{x}})^{-1} = A_{\boldsymbol{w}}^T A_{\boldsymbol{w}} = A^T A + \sum_{k=1}^{s} w_k A_k^T A_k\ . \tag{5}$$

The experimental design approach consists in choosing the design $\boldsymbol{w}$ in order to minimize the variance of the estimator $\hat{\boldsymbol{x}}$, or equivalently to maximize the information matrix $M_F(\boldsymbol{w})$. This maximization should be done with respect to the natural ordering of the space $\mathbb{S}_m$, which is induced by the cone $\mathbb{S}_m^+$ of positive semidefinite matrices, namely the Löwner ordering:

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

Since this is only a partial ordering, the problem consisting in maximizing $M_F(\boldsymbol{w})$ is ill-posed. So we rather maximize a scalar *information function* of $M_F(\boldsymbol{w})$, i.e. a function mapping $\mathbb{S}_m^+$ onto the real line, and which satisfies natural properties such as positive homogeneity, monotonicity with respect to Löwner ordering, and concavity. For a more detailed description of these information functions, the reader is referred to the book of Pukelsheim [10]. A standard choice is to use the class of matrix means Φ_p, which are essentially the "L_p-norms" of the vector of eigenvalues of the information matrix, but for $p \in [-\infty, 1]$. For a positive definite matrix M with positive eigenvalues $\{\lambda_1, \ldots, \lambda_m\}$, the function Φ_p is given by

$$\Phi_p(M) := (\frac{1}{m}\text{trace } M^p)^{1/p} = (\frac{1}{m}\sum_{j=1}^{m} \lambda_j^p)^{1/p} \qquad (6)$$

Interestingly, Φ_p can also be defined for $p = -\infty$ and $p = 0$ as limiting cases, i.e. $\Phi_{-\infty}(M) = \lambda_{\min}(M)$ and $\Phi_0(M) = (\det(M))^{\frac{1}{m}}$, which are the classical optimality criteria referred to in the literature as $E-$ and $D-$optimality respectively. The $A-$optimality criterion, which aims at maximizing the harmonic average of the eigenvalues of $M_F(\boldsymbol{w})$ is obtained for $p = -1$. The definition of Φ_p is also extended to the case where M is a singular positive semidefinite matrix, by continuity, so that $\Phi_p(M) = 0$ for all $p \leq 0$, and $\Phi_p(M)$ is defined by Equation (6) for all $p \in]0, 1]$.

We now give a mathematical formulation to the problem of optimally deploying Netflow on no more than n interfaces:

$$\max_{\boldsymbol{w} \in \{0,1\}^s} \Phi_p\Big(M_F(\boldsymbol{w})\Big) \qquad \text{s.t.} \quad \sum_i w_i \leq n \qquad (7)$$

B. Optimal sampling rates

We now show that the optimal sampling problem can be formulated in a similar form as Problem (7). Following [4], we assume that Netflow performs a random sampling with rate w_k on the interface k. As explained in section I-B, Netflow data can be used to count the packets depending on their internal destination. Let N_{kd} be a counter that records the number of sampled packets from interface k which have the internal destination d. This number follows a binomial distribution with $y_{kd} = A_{kd}\boldsymbol{x}$ trials and probability of success w_k, where A_{kd} is the row corresponding to the destination d in the matrix A_k. The best unbiased linear estimator of $\boldsymbol{y_k}$ is given by $(\hat{\boldsymbol{y}}_{\boldsymbol{k}})_d = w_k^{-1} N_{kd}$, and we have:

$$\begin{aligned} \text{Var}(\hat{\boldsymbol{y}}_{\boldsymbol{k}})_{d,d} = w_k^{-2}\,\text{var}(N_{kd}) &= w_k^{-2}\ w_k(1-w_k)\ y_{kd} \\ &\approx w_k^{-1}\ (A_k\boldsymbol{x})_d, \end{aligned} \qquad (8)$$

where the latter approximation is valid in the (expected) case where the sampling rates are small. The aggregate observation matrix is now $\tilde{A} := [A^T, A_1^T, \ldots, A_s^T]^T$, since measurements are performed on all interfaces: $\boldsymbol{y} = \tilde{A}\boldsymbol{x} + \epsilon$. The design variable $\boldsymbol{w}$ (the sampling rates) is now involved in the covariance matrix of the noise $\boldsymbol{\epsilon}$. The noise on the SNMP data is usually small compared to the noise resulting from Netflow sampling, and we model its variance as $\sigma^2 I$ for a small parameter σ. From (8), we find the form of the covariance matrix:

$$\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \Sigma(\boldsymbol{w}) = \begin{pmatrix} \sigma^2 I & & & \\ & \frac{\text{diag}(A_1\boldsymbol{x})}{w_1} & & \\ & & \ddots & \\ & & & \frac{\text{diag}(A_s\boldsymbol{x})}{w_s} \end{pmatrix}.$$

As done previously in the discrete case, we can make explicit the best linear estimate of the flows, as well as the information matrix of the sampling design $\boldsymbol{w}$:

$$\begin{aligned} \hat{\boldsymbol{x}} = \Big(\tilde{A}\Sigma(\boldsymbol{w})^{-1}\tilde{A}\Big)^{-1}\tilde{A}^T\Sigma(\boldsymbol{w})^{-1}\boldsymbol{y}. \qquad (9) \\ M_F(\boldsymbol{w}) = \tilde{A}^T\Sigma(\boldsymbol{w})^{-1}\tilde{A}. \end{aligned}$$

Finally, we define the normalized observation matrices $\overline{A} = \sigma^{-1}A$ and $\overline{A_i} = \text{diag}(A_i\boldsymbol{x})^{-1/2}A_i$, so that the information matrix can be written as

$$M_F(\boldsymbol{w}) = \overline{A}^T\ \overline{A} + \sum_{k=1}^{s} w_k \overline{A_k}^T\ \overline{A_k}\ . \qquad (10)$$

Hence, the Φ_p-maximization of $M_F(\boldsymbol{w})$ takes a similar form as Problem (7), with a continuous variable $\boldsymbol{w}$ that is subject to linear constraints: the Internet provider typically sets a threshold on the volume of packets to be analyzed with Netflow at each router location, so as to limit the overhead. For a specific router $\mathcal{R}$, the number of sampled packets can be approximated by

$$\sum_{k \in \mathcal{I}_\mathcal{R}} w_k f_k,$$

where the sum is carried out over the interfaces of router $\mathcal{R}$, and f_k is the total number of packets traversing interface k. In practice, f_k can be estimated from previous values of the SNMP data. The constraints can thus be summarized as the set of inequalities $R\boldsymbol{w} \leq \boldsymbol{b}$, where R depends on $\boldsymbol{y}^{\text{SNMP}}$ and $\boldsymbol{b}$ is a target set by the Internet provider. This is an alternative approach to that of Singhal and Michailidis [4], who use a matrix R depending only on the topology of the network.

It remains to cope with the fact that the normalized observation matrices $\overline{A_i}$ explicitly depend on the unknown $\boldsymbol{x}$. Similarly to what is done in [4], we use a prior estimate of $\boldsymbol{x}$ to compute an approximate version of the $\overline{A_i}$. In the numerical studies presented in Section IV, we track the OD flows over time in a network, and we use the previous estimate $\hat{\boldsymbol{x}}_{t-1}$ in place of $\boldsymbol{x}_t$. At $t = 1$, we can use a tomogravity estimate [11] of $\boldsymbol{x}$, which is a classical prior in the traffic matrix estimation literature.

III. Resolution of the problem

In this section, we review previous methods to solve the discrete *Netflow optimal deployment* problem as well as its continuous relaxation (*Netflow optimal sampling*), and we develop a new one. For simplicity of notation, we assume that the observation matrices have already been normalized by the left diagonal scaling mentioned in Section II-A, so that $M_F(\boldsymbol{w})$ takes the form (5).

Note that any method which solves the continuous problem can be applied to obtain an approximate solution to the discrete problem, by applying simple rounding heuristics.

A. Greedy Algorithm

In the discrete case, and when there is a single constraint of the form $\sum_i w_i \leq n$, we can make use of a greedy algorithm, which is suggested by the results of [8]. The principle is to add sequentially in $\mathcal{G} = \emptyset$ the interfaces which provide the best increment $\Phi_p(\mathcal{G} \cup i_k) - \Phi_p(\mathcal{G})$ until $\mathcal{G}$ contains n interfaces.

On a network with $m = 10^4$ OD pairs, the computation of the objective function $\Phi_p(\boldsymbol{w})$ requires about 5 minutes on a PC at 4GHz, since it involves the diagonalization of a $m \times m$ matrix. Consequently, selecting only one out of one hundred interfaces already requires more than 3 hours. The authors of [6] proposed to use the special values $p = 0$ or $p = -1$, for which we can implement the Fedorov sequential design algorithm [12], which computes efficiently the increment of the criterion thanks to Sherman-Morrison like formulae.

At the beginning of the algorithm, the initial observation matrix $M_0 = A^T A$ is not invertible. The authors of [6] remedy this problem by regularizing the initial observation matrix: they set $M_0 = A^T A + \varepsilon I$, with $\varepsilon = 0.001$. If we leave aside the information from the SNMP measurements ($M_0 = \varepsilon I$), this algorithm performs astonishingly well, and the set of interfaces of a very large network can be ordered very quickly. However, if we want to take into account the SNMP measurement (so as to avoid redundancy), M_0 is not sparse anymore, and small-rank updates of the information matrices are intractable. The authors of [6] work on a similar experimental design problem, and store a sparse LU decomposition of M in place of the full M^{-1} matrix, which still allows one to compute $M^{-1}A_k^T$. In our case though, the LU decomposition is full and the greedy updates are intractable.

B. Semidefinite Programming

It is known that the continuous relaxation of Problem (7) can be formulated as a Semi-definite program (SDP) for the special cases of $E-$optimality ($p = -\infty$) and $A-$optimality ($p = -1$) [13]. With the per-router constraints discussed in Section II-B, the $A-$optimal sampling design $\boldsymbol{w}^*$ is the solution of the following problem:

$$\begin{aligned} \min_{q} \quad & \sum_{j=1}^{m} q_j \\ s.t. \quad & \left(\begin{array}{c|c} M_F(\boldsymbol{w}) & \boldsymbol{e_j} \\ \hline \boldsymbol{e_j}^T & q_j \end{array} \right) \succeq 0, \quad j = 1, ..., m \\ & R\boldsymbol{w} \leq \boldsymbol{b}, \ \boldsymbol{w} \geq 0, \end{aligned} \tag{11}$$

where $\boldsymbol{e_j}$ denotes the j^{th} vector of the canonical basis of $\mathbb{R}^m$. This SDP approach has been used in [4] in a Kalman filtering context, where $M_F(\boldsymbol{w})$ contains an additional constant term which accounts for the covariance matrix of the errors of the past measurements. However, this SDP is intractable by state-of-the-art solvers for networks with more than $m \simeq 300$ OD pairs.

C. Successive $\boldsymbol{c}-$Optimal Designs

The hardness of the optimal experimental design is linked to the large dimension of the parameter that we want to estimate, which leads to large size covariance matrices. Rather than estimating the full parameter $\boldsymbol{x}$, a natural idea is to estimate a linear combination $z = \boldsymbol{c}^T \boldsymbol{x}$ of the flows, for which the best linear unbiased estimator $\hat{z}$ and its variance are known [14] :

$$\text{var}(\hat{z}) = \boldsymbol{c}^T M_F(\boldsymbol{w})^\dagger \boldsymbol{c},$$

where $M^\dagger$ denotes the Moore-Penrose inverse of M. The variance of this estimator is a scalar, but it still depends (non-linearly) on a $m \times m$ matrix.

The $\boldsymbol{c}-$optimal design problem has a SDP formulation which is similar to (11), (see e.g [15]). While this SDP can still be intractable on some large instances, it actually reduces to a second order cone Program. A related result appeared in [16] for an application to trust topology design (see also [17]). A proof of our theorem relies on the fact that the dual of SDP (11) admits a solution that is a matrix of rank 1 (see [18], [19] for more information). Second-order cone programs are optimization problems that are somehow harder than linear programs (LP), but which can be solved by interior point codes in a much shorter time than SDP of the same size.

Theorem III.1. *If $\boldsymbol{c} \in \text{range}(\tilde{A}^T)$, then the $\boldsymbol{c}-$optimal design problem (minimizing $\boldsymbol{c}^T M_F(\boldsymbol{w})^\dagger \boldsymbol{c}$ under the constraints $R\boldsymbol{w} \leq \boldsymbol{b}$) is equivalent to the following SOCP:*

$$\begin{aligned} \min_{\boldsymbol{w},\ \boldsymbol{\mu},\ (\boldsymbol{y_i})_{i=0,\ldots,s}} \quad & \sum_{i=0}^{s} \mu_i \\ & A^T \boldsymbol{y_0} + \sum_{i=1}^{s} A_i^T \boldsymbol{y_i} = \boldsymbol{c} \\ & R\boldsymbol{w} \leq \boldsymbol{b}, \ \boldsymbol{w} \geq \boldsymbol{0} \\ & \left\| \begin{bmatrix} 2\boldsymbol{y_0} \\ 1 - \mu_0 \end{bmatrix} \right\| \leq 1 + \mu_0 \\ & \left\| \begin{bmatrix} 2\boldsymbol{y_i} \\ w_i - \mu_i \end{bmatrix} \right\| \leq w_i + \mu_i, \ (i = 1, \ldots, s). \end{aligned} \tag{12}$$

This theorem shows how to compute the optimal weights $\boldsymbol{w}^*$ on the measurements for the $\boldsymbol{c}-$combination of the flows by SOCP. This can be done very efficiently with interior points codes such as SeDuMi [20]. Moreover, this method takes advantage of the sparsity of the matrices A_i, while the equivalent SDP formulation involves the information matrices $A_i^T A_i$, which are *not very sparse* in general).

Remark III.2. When there is a single constraint imposed on the design (i.e. when R is a row vector, so that the

constraint reduces to $\sum_i p_i w_i \leq b$), the SOCP admits a simpler equivalent form without hyperbolic constraints, see our previous work [21].

SCOD We recall that the $\boldsymbol{c}-$optimal design should be chosen if we want to estimate the quantity $z = \boldsymbol{c}^T \boldsymbol{x}$. An advantage of this method is that the operator can choose $\boldsymbol{c}$ so that it gives more weight to the most important flows. This feature will be studied numerically in Section IV. There are many possible choices for this vector $\boldsymbol{c}$, and so an interesting possibility is to choose several vectors (either for their significance or tossed randomly), and finally to combine (for instance by taking the mean) the resulting optimal designs: we call this method SCOD (Successive $\boldsymbol{c}-$Optimal Designs).

A stochastic SCOD Let us study the latter randomized scheme more precisely in the case where the vectors $\boldsymbol{c}$ are normally distributed random vectors ($\boldsymbol{c} \sim \mathcal{N}(\boldsymbol{0}, I)$): the mean of the resulting $\boldsymbol{c}-$optimal designs are a Monte-Carlo approximation approximation of the design defined by

$$\mathbb{E}[\text{argmin}_{\boldsymbol{w}}\, \boldsymbol{c}^T M_F(\boldsymbol{w})^\dagger \boldsymbol{c}], \tag{13}$$

where the minimum is taken over the samplings $\boldsymbol{w}$ that respect the constraint $R\boldsymbol{w} \leq \boldsymbol{b}$. Remarkably, the A-optimality criterion for $\boldsymbol{w}$ can be expressed as follows:

$$\begin{aligned} \text{trace}\, M_F(\boldsymbol{w})^{-1} &= \text{trace}(\mathbb{E}[\boldsymbol{c}\boldsymbol{c}^T] M_F(\boldsymbol{w})^{-1}) \\ &= \mathbb{E}[\boldsymbol{c}^T M_F(\boldsymbol{w})^\dagger \boldsymbol{c}], \end{aligned}$$

where we have used the fact that $\mathbb{E}[\boldsymbol{c}\boldsymbol{c}^T] = \text{Var}(\boldsymbol{c}) = I$.
We do not claim that the presented stochastic SCOD converges to an $A-$optimal design, since $\mathbb{E}[\cdot]$ and argmin$(\cdot)$ do not commute in general. However, we observed numerically on a large number of examples that the design obtained by averaging several $\boldsymbol{c}-$optimal designs (for random vectors $\boldsymbol{c} \sim \mathcal{N}(\boldsymbol{0}, I)$) was very close to the $A-$optimal design. This nice property of the stochastic SCOD will be illustrated in Section IV-A.

IV. Experimental Results

In this section we evaluate the performance of our SCOD approach. To this end, we investigate several issues: in a first part, we compare our SCOD to the (exact) $A-$optimal design for a special instance of the problem. Then, we examine the quality of the estimation of the traffic matrix in different situations, in order to compare the performance of our method with previously proposed ones. We study separately the discrete problem of finding a subset of interfaces for Netflow, and the optimal sampling problem.

The data we used for those experiments comes from two networks. On the one hand, from the Abilene Internet2 backbone, which is a major academic network in the USA, and consists in $n = 11$ nodes $m = n^2 = 121$ OD pairs and $l = 50$ links. Real traffic matrices from this network are available through the Internet2 Observatory project. We used the measurements of the second week of April 2004, as collected by Zhang [22]. The data has a resolution of 10 minutes, resulting in 1008 time steps over the week.

Design ($\times 10^{-1}$)	Stochastic SCOD ($N = 10$)	Stochastic SCOD ($N = 50$)	A-optimal
CPU (sec.)	3.72	18.7	492.6
w_1 (Atlanta)	0.559	0.779	0.749
w_2 (Chicago)	0.883	0.854	0.898
w_3 (Denver)	1.721	1.592	1.510
w_4 (Houston)	0.692	0.772	0.720
w_5 (Indiana)	1.458	1.291	1.361
w_6 (Kansas)	1.252	1.262	1.171
w_7 (LA)	0.556	0.572	0.657
w_8 (NY)	1.329	1.134	1.121
w_9 (Sunnyv.)	1.076	1.184	1.201
w_{10} (Seattle)	0.000	0.002	0.000
w_{11} (Wash.)	0.433	0.557	0.613

TABLE I
Abilene: comparison of the $A-$optimal design and SCOD.

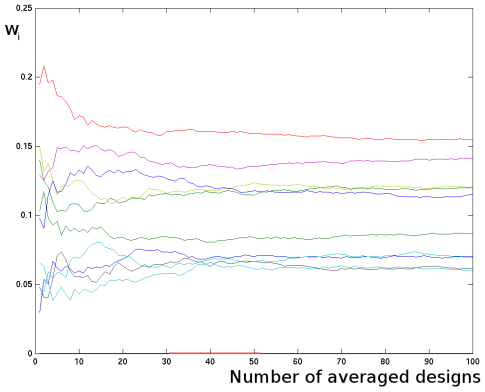

Fig. 1. Convergence of the SCOD method. Each curve represents the evolution of a coordinate of $\boldsymbol{w}$ with the number of averaged designs.

On the other hand, we use measurements from a much larger commercial network, the international backbone "Opentransit", which consists in $n = 116$ nodes, $m = n^2 = 13456$ OD-pairs, and $l = 436$ links. Since this network is only partially instrumented with Netflow (we dispose of Netflow measurements on 34 out of 116 routers), we simulated the missing data for the sake of experiments, by following the instructions of [23]. Namely, we noticed that the fit of the partially available data with a lognormal distribution was very good, so we simulated the missing flows with respect to this distribution, and we assigned them to the non-measured OD pairs of the network thanks to a heuristic procedure based on the topology of the network [23]. The data has a resolution of 2 hours and was collected during 40 hours, and so we track the flow volumes over 20 time steps.

The SNMP and Netflow measurements were simulated from the traffic matrices. The SNMP data was supposed to be almost perfect ($\sigma = 1$), and the Netflow sampling was simulated with a binomial distribution, as seen in Section II-B.

A. Analysis of Stochastic SCOD

We study an experimental design problem on Abilene, where the objective is to find the optimal amount of experimental effort to spend on each router (we handle the data collected on all incoming interfaces of a given router as a single experiment). In Table I, we compare the $A-$optimal sampling rates found by solving the SDP (11), and the design obtained by the stochastic successive $\boldsymbol{c}-$optimal design

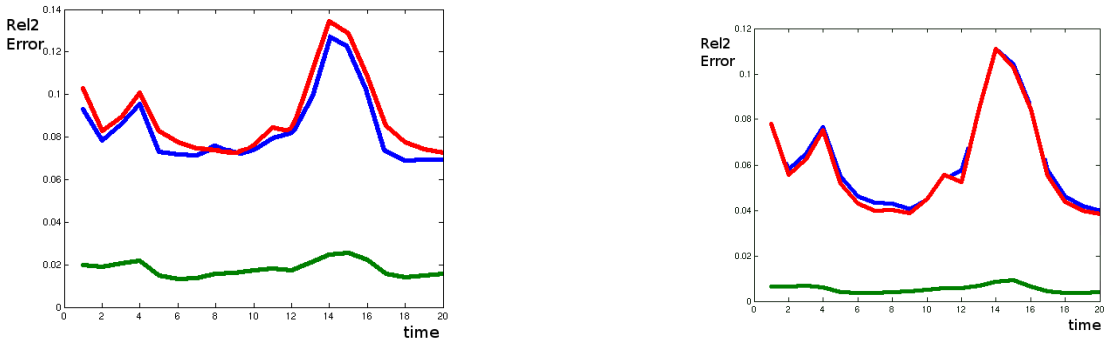

Fig. 2. Relative $L2$ error on Opentransit, for Netflow activated on 16 routers (left) and 30 routers (right), as selected by the stochastic SCOD[this paper] (blue), weighted stochastic SCOD[this paper] (green), and greedy "Netquest" [6] (red).

approach described in Section III-C. The constraint considered here was the unit cost case: $\sum_i w_i \leq 1$. The $\boldsymbol{c}-$optimal designs are computed by Program (12). The designs indicated in the tables were obtained by averaging $N = 10$ and $N = 50$ $\boldsymbol{c}-$optimal designs. To see the convergence of the SCOD, we have plotted in Figure 1 the evolution of each coordinate of the design with N.

It is striking that the designs found by these two approaches are very close and that the computation is much shorter for the SCOD. Namely, solving one instance of the $\boldsymbol{c}-$optimal problem requires only 345ms on average for this network, which is 3 orders of magnitude faster than the 514s required to solve the SDP (11). Furthermore, the SDP approach is intractable on large networks with more than 300 OD pairs.

B. Estimation methodology and Error metrics

Before studying the quality of the estimation of the traffic matrix, we describe the methodology used for the inference. For the optimal deployment problem (Section IV-C), we use the entropic projection approach [1] to track the flow volumes over time. Namely, we choose at each time step the vector of flows which is the closest to a prior (in terms of Kullback-Leibler divergence), among all the flows satisfying the measurement equation (2). At time $t = 1$, the prior is taken equal to the tomogravity estimate [11] of the flows. Then, we choose as prior the previous estimate $\hat{\boldsymbol{x}}_{t-1}$. The entropic projection is carried out by the Iterated Proportional Fitting (IPF) algorithm (see e.g. [1]).

For the optimal sampling problem (Section IV-D), the observation matrix $\tilde{A}$ usually has full column rank, so that we can use the inversion formula (9) to compute the best linear unbiased estimate $\hat{\boldsymbol{x}}$ of the flows, where $\Sigma(\boldsymbol{w})$ is estimated thanks to a prior estimate of the flows. To avoid eventual negative values, we next apply the IPF procedure, as in [24].

To measure the quality of an estimator of the flows $\hat{\boldsymbol{x}}_t$ at a time step t, we use the classical *Relative $L2-$norm*, defined as:

$$\mathrm{Rel}_2(\hat{\boldsymbol{x}}_t) = \frac{\|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2}{\|\boldsymbol{x}_t\|_2}. \tag{14}$$

Similarly, the spatial distribution of the errors can be measured by the spatial relative L_2-error, which is defined for each OD flow time series $\boldsymbol{x}_{\mathrm{OD}}$:

$$\mathrm{Rel}_2(\hat{\boldsymbol{x}}_{\mathrm{OD}}) = \frac{\|\hat{\boldsymbol{x}}_{\mathrm{OD}} - \boldsymbol{x}_{\mathrm{OD}}\|_2}{\|\boldsymbol{x}_{\mathrm{OD}}\|_2}.$$

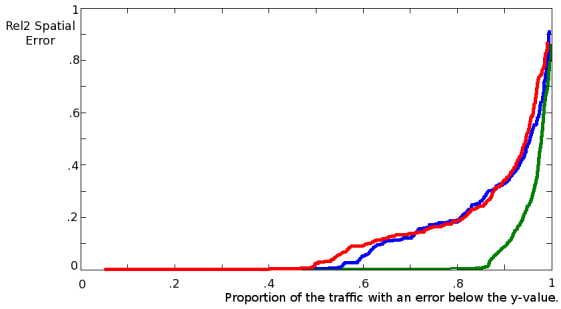

Fig. 3. Quantile function of the spatial errors on Opentransit, for Netflow activated on 16 nodes. These nodes are selected by [SCOD (blue), weighted SCOD (green), and greedy (red)].

C. Netflow Optimal Deployment

We now study the case of the discrete problem, where the objective is to activate Netflow only on a subset of interfaces of the network. We assume throughout this section that when Netflow is activated on an interface, it samples packets at a rate of 10^{-3}. This problem may look very academic, since routing changes occur quite often in practice, and the deployment of Netflow should not be decided in a special routing configuration. However, we show in this section that our SCOD improves on the greedy design, and we want to develop for future work a more robust version of our model. For example, if we are given several potential forms $M^{(i)}(\boldsymbol{w})$ for the information matrix, we could with little change write a version of our SOCP which minimizes the worst variance $\max_i \ \boldsymbol{c}^T M^{(i)}(\boldsymbol{w})^\dagger \boldsymbol{c}$.

We can see in Figure 2 the relative $L2$ error plotted over time in different situations, on Opentransit: Netflow was activated on a subset of 16 or 30 nodes, either selected by the greedy algorithm or by the SCOD procedure. Interestingly, we also computed a design by a weighted stochastic SCOD procedure, where the vectors of the linear combination follow $\boldsymbol{c} \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\hat{\boldsymbol{x}}))$, where $\hat{\boldsymbol{x}}$ is the tomogravity estimate of the flows at time $t = 1$. The number of averaged optimal designs was set to 20, so as to keep the time of computation reasonable, and because we felt that the process almost

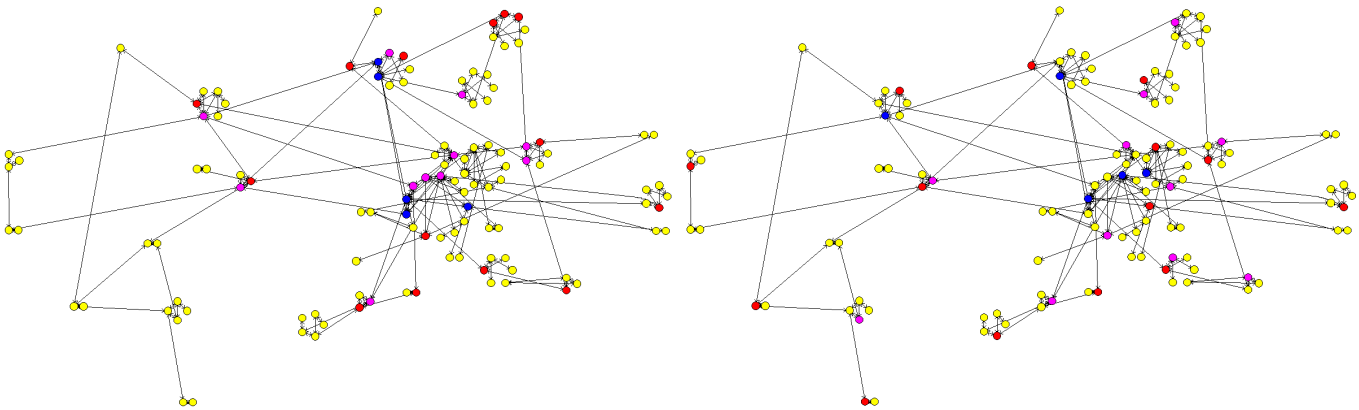

Fig. 4. Opentransit: Location of the routers found by weighted stochastic SCOD[this paper] (left), and greedy "Netquest" [6] (right). The routers in blue, purple, and red correspond to the subsets of 5, 16, and 30 "best" routers found by each method.

converged. The vector $\boldsymbol{c}$ now gives more weight to important flows, and so these flows are better estimated. While the SCOD gives results of a similar quality as the greedy design, the weighted SCOD substantially improves the relative $L2$ error.

In order to illustrate the spatial distribution of the errors, we have plotted on Figure 3 the weighted quantile function of the spatial $L2$ relative error: the graph indicates the fraction of traffic (on the x-axis) which is estimated with a $L2$ relative error below the value on the y-axis. We see that the weighted SCOD outperforms the uniform SCOD and the greedy design for the estimation. In fact, some small flows, which account for less than 1% of the total traffic, are best estimated with a uniform scheme.

We show in Figure 4 the location of the routers found by weighted stochastic SCOD and the greedy algorithm. We notice here that the weighted SCOD procedure yields a design which is more concentrated at the "center" of the network, where the flows are probably more important.

D. Optimal Sampling

We now turn to the study of the optimization of the sampling rates for Netflow. As for the discrete problem, the SCOD are computed by averaging 20 $\boldsymbol{c}-$optimal designs, with $\boldsymbol{c} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\hat{\boldsymbol{x}}))$. New sampling rates are evaluated at each time step, with the prior $\hat{\boldsymbol{x}}$ taken as the previous estimate of the flows. In a more realistic setup, we could recompute sampling rates each time a routing change occurs. In order to avoid numerical issues, we imposed a minimal sampling rate of 10^{-6} on each interface. We have assumed the unit-cost case ($\sum w_i \leq 10^{-3}$), i.e. $\boldsymbol{c}-$optimal designs are computed by Program (12) with $R = \mathbf{1}^T$ (the row vector of all ones) and $b = 10^{-3}$. We found that our SCOD procedure could handle per-router constraints very shortly before the time of submission, and had not the time to make the experiments with these more realistic constraints.

To illustrate the fact that one can recover the volume of the flows without any Netflow measurements on the ingress interfaces of the network (as the standard methodology suggests [5]), we studied the case where we activate Netflow only

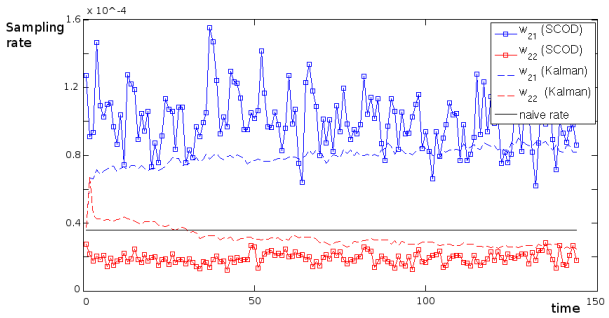

Fig. 5. Evolution of the sampling rates on 2 interfaces of Abilene.

on the 28 internal links of Abilene (resp. 320 links on Abilene), and we computed the SCOD sampling on these interfaces. Each $\boldsymbol{c}-$optimal design problem was solved by Program (12) within roughly 0.3s for Abilene and 120s for Opentransit with SeDuMi on a PC at 4GHz.

We have compared our method with the $A-$optimal design approach in a Kalman filtering context, as proposed in [4]. So we computed optimal rates on a period of 144 time steps with this technique, using the same settings that the authors described, for the case of a "noisy initialization". Figure 5 shows the evolution of the sampling rates on 2 interfaces of Abilene, as well as the value of the naive sampling rate ($w_{\text{naive}} = \frac{10^{-3}}{s}$ on every interface). Interestingly, it seems that the design computed in a Kalman filtering process *converges* to our design. This could be explained by the fact that, due to the high variability of the traffic, the prediction step $\boldsymbol{x}_{t|t-1} = C\hat{\boldsymbol{x}}_{t-1}$ (with $C = I$ as in [4]) of the Kalman filter is of poor quality compared to the correction step which uses the Netflow measurements. Moreover, the flows computed by our approach had a relative $L2$ error in the order of 10^{-3}, while it attained 20% with the Kalman filter: Simply inverting the sampling measurements (as we do) yields better results than processing them in a Kalman filter, since the state transition equation from one time step to the next one may be inaccurate.

We still want to evaluate the benefits of considering the past measurements for the computation of the sampling rates. So

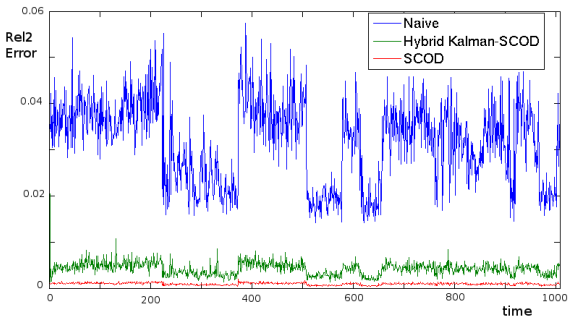

Fig. 6. Relative $L2$ error for different sampling rates of Netflow, on Abilene.

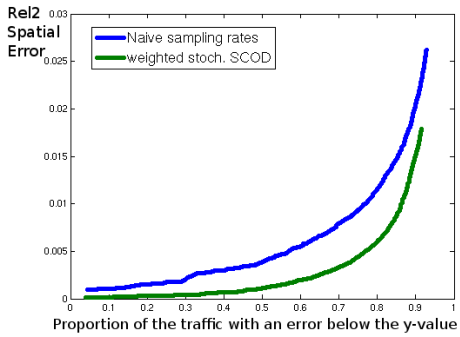

Fig. 7. Distribution of the spatial relative $L2$ error in Opentransit.

we built a new estimate of the flows, where the Kalman filter is used only to update the covariance matrix of the estimation which we "minimize" to select the sampling rates. Then, the estimation is carried out by the inversion Formula (9) and the IPF. To speed up the computation, we used our heuristic SCOD scheme in place of the $A-$optimal design SDP (11). We compare in Figure 6 the relative $L2$ error of this new estimate (called "Hybrid Kalman-SCOD") with the estimations based on the naive sampling rates and the SCOD. Our sampling rates perform much better than the naive ones. Note that the estimation of the flows is very accurate with our sampling rates, although no Netflow measurement was performed on the ingress links. It is also clear that taking into account the past measurements does not yield any improvement.

We finally analyze the quality of the estimation with a sampled Netflow on Opentransit, with the constraint $\sum_i w_i \leq 10^{-2}$. This constraint looks weaker that the one we imposed on Abilene, but yields similar sampling rates because the number of interfaces is larger ($s = 320$). Since we are not aware of any other sampling selection scheme that can be used on a network of this size, we only compare our estimate with the case of naive sampling ($\boldsymbol{w} = \frac{10^{-2}}{320}\mathbf{1}$). We have plotted in Figure 7 the spatial distribution of the errors obtained with our SCOD sampling scheme and the naive one. Once again, the estimation of the traffic is much better with the SCOD.

V. Conclusion

In this paper, we have proposed a new method to optimize the traffic measurement, based on the estimation of a sequence of linear combinations of the flows, rather than on the estimation of the full vector of flows. This method remains tractable for very large instances, and it allows one to identify the traffic accurately. Our numerical results moreover show that our method can be tuned by the operator in order to best estimate some flows of special importance.

References

[1] G. Liang, N. Taft, and B. Yu, "A fast lightweight approach to origin-destination ip traffic estimation using partial measurements," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2634–2648, 2006.

[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 : 2nd Conference on Next Generation Internet Design and Engineering, Valencia*, 2006, pp. 224–231.

[3] CISCO, "Netflow performance analysis, technical white paper," 2007.

[4] H. Singhal and G. Michailidis, "Optimal sampling in state space models with applications to network monitoring," in *SIGMETRICS'08*, Annapolis, Maryland, USA, 2008.

[5] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational ip networks: Methodology and experience," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 265–280, 2001.

[6] H. Song, L. Qiu, and Y. Zhang, "Netquest: A flexible framework for largescale network measurement," in *ACM SIGMETRICS'06*, St Malo, France, 2006.

[7] G. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran, "Reformulating the monitor placement problem: Optimal network-wide sampling," in *CISS*, 2006, pp. 1725–1731.

[8] M. Bouhtou, S. Gaubert, and G. Sagnol, "Optimization of network traffic measurement: a semidefinite programming approach," in *Proceedings of the International Conference on Engineering Optimization (ENGOPT)*, Rio De Janeiro, Brazil, 2008, ISBN 978-85-7650-152-7.

[9] H. Zang and A. Nucci, "Optimal netflow deployment in IP networks," in *19th International Teletraffic Congress (ITC)*, Beijing, China, 2005.

[10] F. Pukelsheim, *Optimal Design of Experiments*. Wiley, 1993.

[11] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *SIGMETRICS '03*, 2003, pp. 206–217.

[12] V. Fedorov, *Theory of optimal experiments*. New York : Academic Press, 1972, translated and edited by W. J. Studden and E. M. Klimko.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[14] G. Elfving, "Optimum allocation in linear regression theory," *The Annals of Mathematical Statistics*, vol. 23, pp. 255–262, 1952.

[15] P. Richtarik, "Simultaneously solving seven optimization problems in relative scale," 2008, optimization online, preprint number 2185.

[16] Y. Nesterov and A. Nemirovsky, "Interior-point polynomial algorithms in convex programming," *SIAM Studies in Applied Mathematics*, vol. 13, 1994.

[17] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, no. 284, pp. 193–228, 1998.

[18] G. Sagnol, "A class of semidefinite programs with a rank-one solution," 2009, arXiv:0909.5577.

[19] ——, "Computing optimal designs of multiresponse experiments reduces to second-order cone programming," 2009, Submitted, arXiv:0912.5467.

[20] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.

[21] G. Sagnol, M. Bouhtou, and S. Gaubert, "Successive c-optimal designs : A scalable technique to optimize the measurements on large networks," in *ACM SIGMETRICS'10*, New York, NY, USA, June 2010, to appear. Extended abstract (2 pages). Preprint: http://www.cmap.polytechnique.fr/~sagnol/papers/sig-2pages.pdf.

[22] Abilene measurements: http://www.cs.utexas.edu/~yzhang/research/Abilene-TM/.

[23] A. Nucci, A. Sridharan, and N. Taft, "The problem of synthetically generating ip traffic matrices: initial recommendations," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 19–32, 2005.

[24] Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information: Making use of multiple data sources," in *ACM SIGMETRICS'06*, St Malo, France, 2006, pp. 133–144.