Internet Mathematics Vol. 3, No. 2: 233-252

An Identification Problem for Multiterminal Networks: Solving for the Traffic Matrix from Input-Output Measurements

F. Alberto Grünbaum and Laura Felicia Matusevich

Abstract. We consider the problem of determining the unknown characteristics of a *random routing strategy* from *end-to-end* measurements. More specifically, we construct a Markov chain that models the traffic of messages in a multiterminal network consisting of *input*, *intermediate*, and *output* terminals. The topology of the network is assumed to be known, but the Markovian routing strategy is not known. We solve the problem of determining the unknown one-step transition probability matrix of our random walk from input-output measurements of "travel time." We give explicit inversion formulas (up to a natural gauge) in a nontrivial example. The result holds for a large (but not arbitrary) class of multiterminal networks, many of which are indicated here. The networks that we display here are constructed in a canonical fashion from certain graphs. Some of these graphs as well as the way to go from graphs to networks are also displayed. One example of a graph for which our method works is the edge graph of a hypercube in any dimension.

1. Introduction

Consider the following exceedingly simple network, consisting of an input terminal I, a intermediate terminal H, and an output terminal O, as seen in Figure 1. Messages can be sent from I to O in two ways: either directly or by means of the intermediate terminal H. The probability of a direct transmission is denoted by P_{IO}. The probability of a message going from I to H is denoted by P_{IH}.

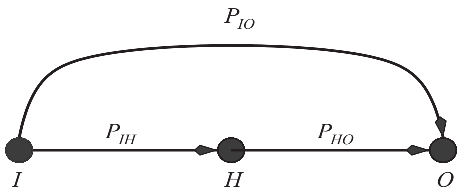

Figure 1. The simplest network.

Once a message gets to H, the probability that it makes its way to O is given by P_{HO}. Notice that we allow for messages to be lost during transmission. We assume that any message that is not lost takes one unit of time to travel between two terminals and that messages do not interfere with each other.

Suppose that we can only make the following *input-output* measurements: we can detect the probability that a message sent from I arrives to O, and we can look at the *random variable* given by the travel time between I and O and determine its expected value. Question: what can one learn about the network from these measurements?

The answer here is quite simple: the probability of going from I to O is given by

$$P_{IO} + P_{IH}P_{HO},$$

while the expected time of travel is given by

$$P_{IO} + 2P_{IH}P_{HO}.$$

It is then clear that both the value of P_{IO} as well as the value of the product $P_{IH}P_{HO}$ can be read off from the data. The individual values of P_{IH} and P_{HO} can only be known up to a common scalar parameter F: we can multiply P_{IH} by F^{-1} and P_{HO} by F, and nobody will be able to notice. Notice that the knowledge of *higher-order moments* of the distribution of the travel time between I and O will not give any extra information.

The purpose of this paper is to show that, for a large class of fairly complicated networks, one can obtain results as nice and explicit as those described above. The only difference will be that scalars are replaced by matrices to reflect the fact that instead of one terminal of each type, we will have several terminals. The free parameter F will become a matrix too, and in general the formulas for the unknowns for which we can solve will become more complex than in the simplest case depicted in Figure 1.

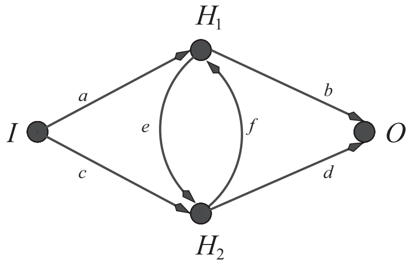

Figure 2. A very simple network.

Before describing a large class of networks with the property alluded to above, we give an example showing that there are also simple networks for which the results are entirely different. For this purpose consider the modification of the example in Figure 1 given by Figure 2, where the only change is that now the number of intermediate terminals does not match the number of input and output terminals.

We consider the network from Figure 2. For simplicity in the notation, let us label the unknown transition probabilities attached to the arrows a through f as in the figure. It is easy to write down the distribution of the travel time random variable. In fact, all we need to do is count paths from source to sink: if k is an integer, the probability of going from source to sink in $2k$ steps is

$$\delta_{2k} = a(ef)^{k-1}b + c(fe)^{k-1}d,$$

and the probability of doing this using $2k+1$ steps is

$$\delta_{2k+1} = a(ef)^{k-1}ed + c(fe)^{k-1}fb.$$

Note that from δ_2, δ_3, and δ_4 we can compute all the other δ_k, since $\delta_4/\delta_2 = ef$. This means that even if we know the whole distribution of travel time, we cannot solve for more than three of the six unknowns. This contrasts with the previous example, where the number of free parameters *coincided* with the number of intermediate terminals. In the examples discussed in the rest of this paper, we will solve for all the unknowns except for a certain number of *free parameters*, which will always coincide with the number of intermediate terminals.

We stress that in the model considered in Section 3, messages are allowed to *revisit* any intermediate terminal an arbitrary number of times: a message starts at an input terminal, meanders its way through a possibly complicated chain of intermediate terminals, and, if not lost along the way, arrives at an output terminal. We record the (random variable) *travel time*, and from this and other such measurements for *all pairs* of input-output terminals, we try to infer information about the routing policy.

2. The Connection with Optical or Diffuse Tomography

The most common applications of mathematical methods to medical imaging involve CAT scanners and MRI imaging. Both of these inverse problems amount to trying to reconstruct the local or hidden characteristics of human tissue from *boundary measurements*, i.e., from certain measurements corresponding to pairs of sources and detectors, one tries to infer the characteristics of the tissue that cannot be observed directly. Both of these techniques are part of the standard tools of many modern hospitals, and they have both revolutionized diagnostic medicine. In 1979 and in 2003, the Nobel Prize in medicine was awarded in connection with these developments.

These ideas have spilled into the analysis of networks, as in the pioneering paper [Vardi 96].

More recently, there has been an effort to use lower energy sources, such as an infrared laser, to obtain images of diagnostic value. See for instance [Singer et al. 90], [Arridge 99] and its references, [Dorn 98], and the papers [Grünbaum 90, Grünbaum 92, Grünbaum 01a, Grünbaum and Patch 92a, Grünbaum and Patch 92b, Patch 95, Patch 99, Patch 94]. In the last collection of papers, one finds a discretized version of the problem where one tries to model the inner characteristics of tissue from a discretized model of the transport of photons among pixels or voxels in the body. The main difference between the physics of this case and the simpler one of x-rays and magnetic resonance is that scattering can no longer be ignored. One has to deal with *nonlinear* equations from the beginning.

In the series of papers [Grünbaum and Matusevich 02a, Grünbaum and Matusevich 02b, Grünbaum and Matusevich 04], we have begun using some of the tools developed earlier for the medical problem in the context of networks. The main difference with the considerations in [Vardi 96] is that we allow for cycles in our networks. Cycles are the counterpart of accounting for multiple scatterings in the medical setup. We are interested mainly in the inverse problem of determining the (random) routing strategy from *source-destination* measurements.

The connection alluded to in this section will be a guide in the selection of the kind of graphs and networks that we consider; more explicitly: the *nodes* of a certain undirected graph will be thought of as *pixels*. To each pixel we associate certain incoming states (infrared sources), certain intermediate or hidden states (transitions to neighboring parts of the body), and certain outgoing states (infrared detectors).

The results in this paper extend quite nontrivially the application of the methods discussed above to problems in network identification.

3. The General Approach

All the networks that we consider are instances of directed graphs.

In the construction of these networks, we will have use for an intermediate step involving another graph. The procedure that takes us from an arbitrary graph to a directed graph will be explained in Section 4. In this section we start from an arbitrary directed graph and introduce the inverse problem whose solution is the purpose of this paper.

Let $G = (V, E)$ be a directed graph with node set V and directed-edge set E. Let I be the set of sources of V, O its set of sinks, and $H = V \backslash (I \cup O)$, the set of intermediate or "hidden" nodes. We think of each node of G as a state in a Markov chain and call the elements of I input terminals, the elements of O output terminals, and the elements of H intermediate terminals. To each directed edge (a, b) of G, we associate an indeterminate $p_{(a,b)}$. This indeterminate represents the one-step transition probability of going from terminal a to terminal b. We refer to these graphs as *multiterminal networks.*

We consider four matrices: P_{IO}, P_{IH}, P_{HH}, and P_{HO}. The rows of P_{IO} are indexed by the input terminals, its columns are indexed by the output terminals, and the a, b entry of P_{IO} is $p_{(a,b)}$ if (a, b) is a directed edge in G. All other entries are zero. The other matrices are defined analogously, and either rows or columns (or both) are indexed by the intermediate terminals. These matrices are the blocks in the one-step transition probability matrix of our Markov chain. We allow for all input and intermediate terminals to be *absorbing states*; that is, we do not insist that the sum $\sum_b p_{(a,b)}$ be unity when a is either an input or an intermediate terminal.

We consider the problem of recovering the matrices P_{IO}, P_{IH}, P_{HH}, and P_{HO} from the distribution of a random variable called *travel time* for any pair made up of an input and an output terminal. If we had access to the complete distribution of this *collection* of random variables, we would have access to the matrices

$$\delta_1 = P_{IO}, \delta_2 = P_{IH}P_{HO}, \delta_3 = P_{IH}P_{HH}P_{HO}, \ldots, \delta_j = P_{IH}P_{HH}^{j-2}P_{HO}, \ldots.$$

These matrices give the probability of making a transition in one, two, three, etc. units of time from an arbitrary input terminal to an arbitrary output terminal. Notice that, under appropriate conditions, this information is mathematically equivalent to the knowledge of the *moments* of travel time, which are given by the matrices

$$Q_{IO}^{(j)} = P_{IO} + \sum_{m=0}^{\infty} (m+2)^j P_{IH} P_{HH}^m P_{HO}.$$

From a practical standpoint, a few of the moments $Q_{IO}^{(j)}$ can be measured, but measuring the matrices δ_j may not be practical.

In view of this, we will take the position that the only available information are the zeroth- and first-order moments of travel time. We now compute these two moments in terms of our *unknowns*,

$$P_{IO}, P_{IH}, P_{HH}, P_{HO}.$$

From the definition of the zeroth moment of travel time, we obtain (after an appropriate summation of the corresponding geometric series) the following expression for $Q_{IO} \equiv Q_{IO}^{(0)}$:

$$Q_{IO} = P_{IO} + P_{IH}(I - P_{HH})^{-1} P_{HO}, \tag{3.1}$$

where I is the identity matrix.

If we denote

$$R = P_{IH}(I - P_{HH})^{-2} P_{HO}, \tag{3.2}$$

then one can see that the first moment of the travel time can be expressed as

$$Q_{IO} + R.$$

Proof. Recall that the jth moment of the travel time is given by

$$Q_{IO}^{(j)} = P_{IO} + \sum_{k=0}^{\infty} P_{IH} P_{HH}^k P_{HO} (k+2)^j.$$

For $j = 1$, we get

$$\begin{aligned} Q_{IO}^{(1)} &= P_{IO} + 2P_{IH}(I - P_{HH})^{-1} P_{HO} + P_{IH} P_{HH} (I - P_{HH})^{-2} P_{HO} \\ &= Q_{IO}^{(0)} + P_{IH}(I - P_{HH})^{-2}[I - P_{HH} + P_{HH}] P_{HO} \\ &= Q_{IO}^{(0)} + R. \qquad \square \end{aligned}$$

Since Q_{IO} is taken as data, we can consider R as the extra information provided by the expected value of travel time.

4. From Graphs to Multiterminal Networks

We now describe a procedure to obtain multiterminal networks from graphs, including a labeling convention for the states. This convention will be extremely important at the time of solving the actual equations. The networks obtained from this procedure are amenable to treatment using our methods. We exemplify this construction using the edge graph of the four-dimensional hypercube. It is important to notice that we have two kinds of graphs: our networks are *directed graphs*, and they are built (as we will see in this section) from *undirected graphs*.

Consider a graph $\mathcal{G}$, for instance, the edge graph of the four-dimensional hypercube from Figure 3. The vertices in this figure represent pixels in our tomographic inspiration. Notice that each vertex in this example has four edges incident to it; we will provide, for any vertex, four states of each of the three kinds.

We start by drawing the intermediate states, which are obtained from the edge graph of the four-dimensional cube (Figure 3) by splitting each edge into two arrows pointing in opposite directions, as in Figure 4. The vertices in this figure represent the pixels, and the arrows are states anchored at each pixel.

We still need to anchor four incoming and four outgoing states at each vertex. If we make the convention that incoming and outgoing states have the same labels, we need only draw the outgoing states. We do this by anchoring dotted arrows at each vertex. These arrows point in directions that are opposite to the solid arrows already anchored there (see Figure 5). One should imagine that at each vertex there is also an incoming state represented by an arrow that ends at

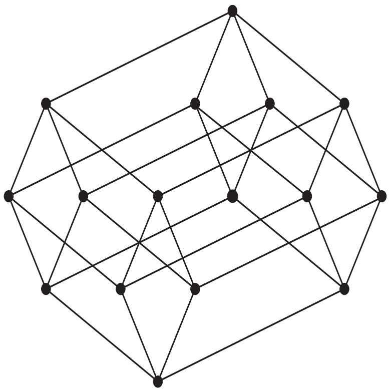

Figure 3. The edge graph of a four-dimensional cube.

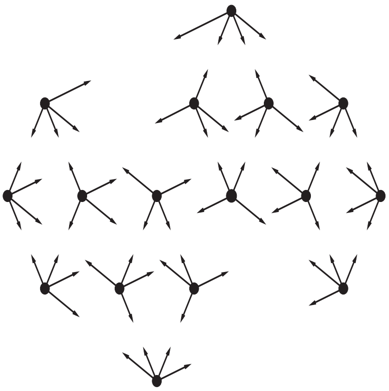

Figure 4. The intermediate states are solid arrows anchored at the vertices.

this vertex. These extra arrows are not drawn in our figures to avoid cluttering the picture, except in the very simple case of Figure 9.

We summarize what we have so far: messages enter the pixels that are now represented by vertices using one of the incoming states which are now input terminals. They travel from vertex to vertex by using intermediate terminals, and they end their journey by leaving one of the vertices using an output terminal.

Finally, we explain our labeling convention. Outgoing (and also incoming) states anchored at a given vertex are numbered consecutively. We finish our labeling by numbering the intermediate states so that the intermediate states entering a given vertex receive the same labels as the outgoing states emerging from that vertex. The best way of doing this is by assigning to a solid arrow the same number as the dotted arrow pointing in the same direction. This is illustrated in Figure 6.

Notice that we have some freedom in doing our labeling because we are not determining the order in which the outgoing states anchored at a given vertex are numbered. We just specify that these numbers are consecutive.

Since incoming and outgoing states are labeled consistently, the matrix P_{IO} is block diagonal, with 16 blocks, each of size 4×4. The matrix P_{HO} has this

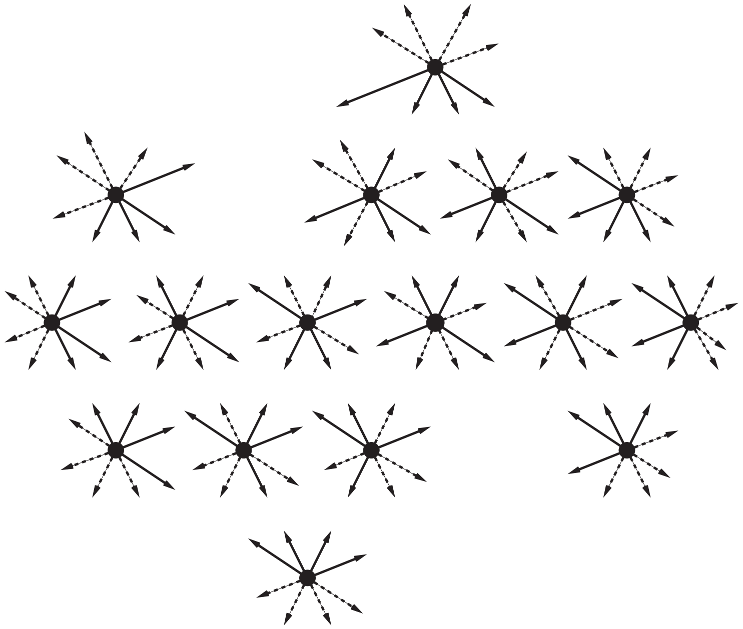

Figure 5. The outgoing states are dotted arrows anchored at the vertices.

same block structure since the intermediate states entering a given vertex have the same labels as the outgoing states emerging from it.

Our labeling also implies that the matrices P_{HH} and P_{IH} have a common block structure, although they are not block diagonal. We show the block structure for one possible labeling in Figure 7. The squares are 4×4 blocks of a 64×64 matrix, and each thin black region is a 4×1 block. Labeling the outgoing states anchored at a vertex in a different order, we obtain other block structures, but in all cases the 4×4 blocks in the diagonal are empty, and once again in all cases between rows $4i + 1$ and $4(i + 1)$, the matrices P_{HH} and P_{IH} have four 4×1 blocks.

We emphasize that the "support" of the matrices P_{IH} and P_{HH} is disjoint from the "support" of the matrices P_{IO} and P_{HO}.

In this case, arising from the hypercube, the corresponding inverse problem has been handled in [Grünbaum and Matusevich 04]. In the next section we consider a simpler situation where the initial (undirected) graph in Figure 3 is replaced by the one on the left in Figure 9. This drastic simplification allows us to display the steps of our general inversion method in complete detail.

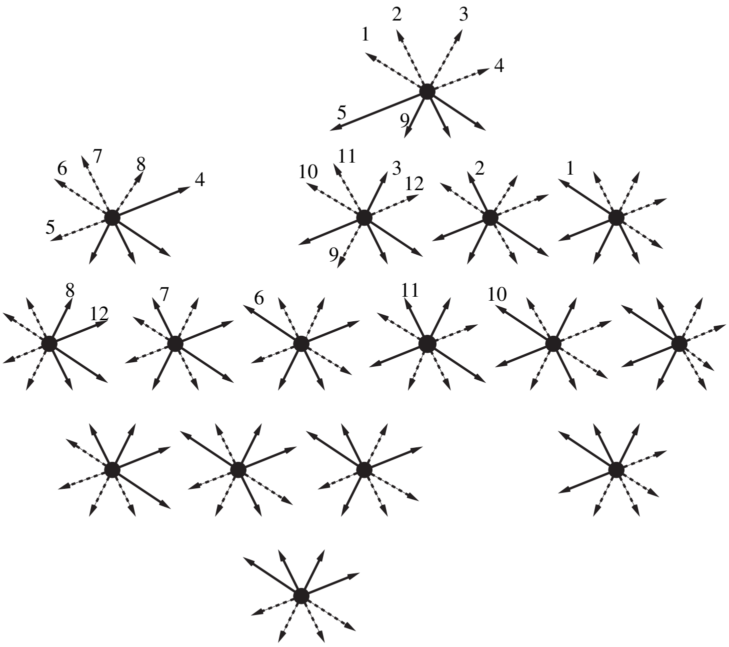

Figure 6. Labeling the states.

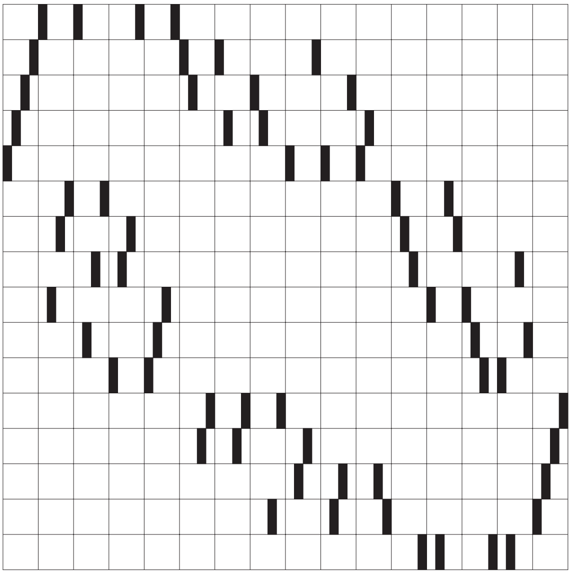

Figure 7. The block structure for P_{HH} and P_{IH} corresponding to the network constructed from the four-dimensional hypercube. The corresponding structure for P_{IO} and P_{HO} is block diagonal and is not worth displaying here.

5. How to Obtain Inversion Formulas

The goal of this section is to outline a method for recovering the matrices P_{IO}, P_{HH}, P_{IH}, and P_{HO} in terms of the entries of the data matrices Q_{IO} and R and of some free parameters.

The next section illustrates the general method in a small but highly nontrivial example.

Recall that the equations to solve are equations (3.1) and (3.2). It is clear that these are nonlinear systems in the unknowns (the entries of the matrices P_{IO}, P_{IH}, P_{HH}, P_{HO}). To remedy this problem, we introduce the following change of variables:

$$A = P_{HO}^{-1}, \quad X = P_{IO}A, \quad W = AP_{HH}, \quad Y = XA^{-1}W - P_{IH}.$$

It is easy to see that under the assumption that P_{HO}^{-1} exists, this is an invertible mapping from the *old variables* P_{IO}, P_{IH}, P_{HH}, and P_{HO} to the *new variables* A, X, Y, and Z.

We now apply our change of variables to equations (3.1) and (3.2) to obtain the following lemma.

Lemma 5.1. $Q_{IO}(A - W) = X - Y$ *and* $Q_{IO}A - X = R(A - W)$.

The following observation is crucial.

Remark 5.2. The equations in Lemma 5.1 are *linear* systems whose unknowns are the entries of the matrices A, X, Y, and W. These equations are equivalent to the nonlinear systems (3.1) and (3.2).

At this point, if we have numerical values for the entries of the matrices Q_{IO} and R, we need only solve the linear systems from Lemma 5.1 and invert the change of variables to recover numerical values for the entries of the matrices P_{IO}, P_{IH}, P_{HH}, and P_{HO}, with the caveat that we might not have enough constraints and that some of the desired entries might be left as free parameters.

As a matter of fact, it is impossible to avoid having free parameters: if F is an invertible diagonal matrix, then the replacement

$$\begin{array}{rcl} P_{IO} & \to & P_{IO} \\ P_{IH} & \to & P_{IH}F^{-1} \\ P_{HH} & \to & FP_{HH}F^{-1} \\ P_{HO} & \to & FP_{HO} \end{array} \tag{5.1}$$

does not alter Q_{IO}, R, or any of the moments of the travel time random variable. In consequence, we will always have at least as many free parameters as there are intermediate terminals. Our goal is to have only this minimal number of free parameters and no more.

It is now time to make the second crucial observation for our method.

Lemma 5.3. *Consider a network arising from an undirected graph in the manner described in Section 4 and labeled as indicated in that section. The block structure of the matrices* P_{IO}, P_{IH}, P_{HH}, *and* P_{HO} *is preserved under the change of variables* (5.1). *That is,* P_{HO} *and* A *have the same block structure, and so do* P_{IO} *and* X, P_{HH} *and* W, *and* P_{IH} *and* Y.

Proof. By our labeling convention, P_{HO} is a block diagonal matrix, and this is not changed under inversion. Thus, A and P_{HO} have the same block structure.

Since P_{IO} and A are block diagonal with the same size blocks, $X = P_{IO}A$ has the same block structure.

Let us now show that P_{HH} and $W = AP_{HH}$ have the same block structure. Assume that A has r blocks of sizes $s_1 \times s_1, s_2 \times s_2, \ldots, s_r \times s_r$, respectively. By our labeling convention, between rows $s_1 + \cdots + s_{i-1} + 1$ and $s_1 + \cdots + s_i$ there are s_i blocks of size $s_i \times 1$. These thin blocks do not occur on the diagonal, and in each column of P_{HH}, there is exactly one block of size $s_i \times 1$ for some i.

Let $s_1 + \cdots + s_{i-1} + 1 \leq t \leq s_1 + \cdots + s_i$, and consider the t, r entry of the product $W = AP_{HH}$. If r is *not* the index of a column where an $s_i \times 1$ block occurs, then the scalar product of the tth row of A and the rth column of P_{HH} will be zero (one vector has zeros where the possibly nonzero entries of the other are). This implies that the block structure is preserved.

A similar argument shows that P_{IH} and Y have the same block structure. □

Now, the fact that A and W, and X, and Y have "disjoint" block structures allows us to solve the linear systems from Lemma 5.1 *symbolically*, meaning that we obtain formulas for the entries of A, X, Y, and W in terms of the entries of Q_{IO} and R without first giving numerical values for these last two matrices. This is tantamount to solving for P_{IO}, P_{IH}, P_{HH}, and P_{HO} symbolically.

The problem still remains that we will have free parameters. As we mentioned before, it is impossible to avoid having as many free parameters as intermediate terminals. The question now is whether the systems in Lemma 5.1 give enough constraints so that we have exactly the minimal number of free parameters. We have been able to check that this is the case in every example we tried of networks arising from undirected graphs as explained in Section 4. We conjecture that this holds for all such networks.

Here is our procedure to recover the transition probabilities for a given network arising from an undirected graph as in Section 4:

1. We set up the linear systems from Lemma 5.1.

2. We solve these equations symbolically. (The block structure ensures that we can do this: we can simply use Cramer's rule to obtain explicit formulas.)

3. Then, if we have the minimal number of free parameters, we invert the change of variables.

It is clear that the class of networks constructed in Section 4 is quite restricted. These networks are tailor-made so that the equations in Lemma 5.1 can be solved symbolically. Notice that not only do we need to have the same number of incoming, outgoing, and intermediate terminals (so that the change of variables makes sense), we also use the very special block structure that disentangles the variables.

We do not believe that this method of obtaining explicit formulas will work for general networks on the same number of incoming, outgoing, and intermediate terminals, let alone networks with more intermediate terminals than incoming or outgoing (these are the most interesting networks for applications). On the other hand, from general principles, one would believe that the inversion problem that we have stated is completely intractable in a symbolic manner, and we were pleasantly surprised to find a class (even a restricted class) of examples for which the solution is feasible. It is our hope that more work on this type of inverse problems will eventually produce solutions for more realistic networks with a view toward concrete applications.

6. An Explicit Example

In this section we consider the multiterminal network drawn in Figure 8. This seemingly complicated network was constructed from the very simple graph in Figure 9 following the prescription given in Section 4.

The data are the matrices $Q_{IO} = (q_{ij})$ and $R = (r_{ij})$. Then, the unknown P_{IO} can be solved explicitly in terms of the data, while the unknowns P_{IH}, P_{HH}, and P_{HO} depend on the data and *four* extra parameters a, b, c, d.

More specifically, we will see that the matrix P_{IO} as well as the matrices $P_{IH}F^{-1}$, $FP_{HH}F^{-1}$, and FP_{HO} (with F an arbitrary diagonal matrix) can be given in terms of the data. We note that the expressions that follow are valid *generically*, i.e., only when the data is such that the denominators involved do not vanish. We also recall that the matrices in which we are interested have the block structure given in Figure 10.

We start by giving the matrix P_{HH} up to conjugation by an arbitrary diagonal matrix F as indicated above. Using this freedom, we can assume that the $(3,2)$ and $(4,1)$ entries of P_{HH} are equal to 1; then (see Figure 10), the 2×2 nontrivial block in P_{HH} is given by the matrix

$$\frac{1}{e+f+g+h}\begin{pmatrix} 2a & e+k-m+h \\ e-k+m-h & -2b \end{pmatrix},$$

where the following abbreviations are used:

$$a = \begin{vmatrix} q_{13} & q_{23} \\ r_{13} & r_{23} \end{vmatrix}; \quad b = \begin{vmatrix} q_{14} & q_{24} \\ r_{14} & r_{24} \end{vmatrix}; \quad e = \begin{vmatrix} r_{13} & r_{14} \\ r_{23} & r_{24} \end{vmatrix};$$

$$f = \begin{vmatrix} q_{13} & q_{14} \\ r_{23} & r_{24} \end{vmatrix}; \quad g = \begin{vmatrix} r_{13} & r_{14} \\ q_{23} & q_{24} \end{vmatrix}; \quad h = \begin{vmatrix} q_{13} & q_{14} \\ q_{23} & q_{24} \end{vmatrix};$$

$$k = \begin{vmatrix} q_{13} & r_{14} \\ q_{23} & r_{24} \end{vmatrix}; \quad m = \begin{vmatrix} r_{13} & q_{14} \\ r_{23} & q_{24} \end{vmatrix}.$$

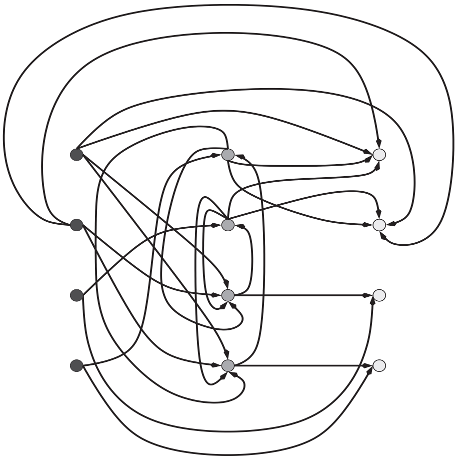

Figure 8. The network arising from a two-point star.

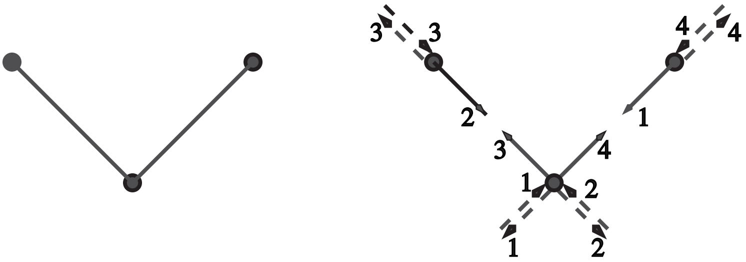

Figure 9. The two-point star graph on the left. On the right, a diagram representing the states: dotted arrows are incoming or outgoing depending on how they point; solid arrows are intermediate states.

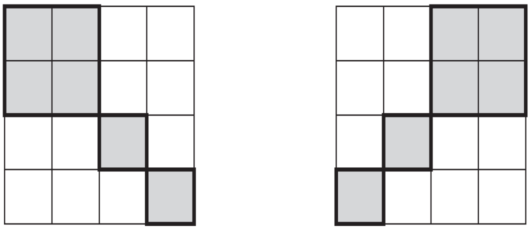

Figure 10. The block structure of P_{IO} and P_{HO} on the left, and that of P_{HH} and P_{IH} on the right. These correspond to the network arising from the two-point star.

The matrix P_{HO} can be assumed (using the freedom in choosing F) to have its $(3,3)$ and $(4,4)$ entries equal to 1; then, the only nontrivial 2×2 block is given by the matrix

$$\frac{1}{e+f+g+h}\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},$$

where the following abbreviations are used:

$$\alpha = -\begin{vmatrix} r_{11} & r_{13} \\ r_{21} & r_{23} \end{vmatrix} - \begin{vmatrix} r_{11} & r_{21} \\ q_{13} & q_{23} \end{vmatrix}; \quad \beta = -\begin{vmatrix} r_{12} & r_{13} \\ r_{22} & r_{23} \end{vmatrix} - \begin{vmatrix} r_{12} & r_{22} \\ q_{13} & q_{23} \end{vmatrix};$$

$$\gamma = \begin{vmatrix} r_{11} & r_{14} \\ r_{21} & r_{24} \end{vmatrix} + \begin{vmatrix} r_{11} & r_{21} \\ q_{14} & q_{24} \end{vmatrix}; \quad \delta = \begin{vmatrix} r_{12} & r_{14} \\ r_{22} & r_{24} \end{vmatrix} + \begin{vmatrix} r_{12} & r_{22} \\ q_{14} & q_{24} \end{vmatrix}.$$

The matrix P_{IH} (up to multiplication by F^{-1}) has nonzero elements given as follows. The entries $(3,2)$ and $(4,1)$ are given by the expressions

$$\frac{2}{\begin{vmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{vmatrix}}\left(q_{32}\begin{vmatrix} r_{11} & q_{13} \\ r_{21} & q_{23} \end{vmatrix} + q_{31}\begin{vmatrix} q_{13} & r_{12} \\ q_{23} & r_{22} \end{vmatrix}\right)$$

and

$$\frac{2}{\begin{vmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{vmatrix}}\left(q_{42}\begin{vmatrix} r_{11} & q_{14} \\ r_{21} & q_{24} \end{vmatrix} + q_{41}\begin{vmatrix} q_{14} & r_{12} \\ q_{24} & r_{22} \end{vmatrix}\right),$$

respectively. We now list the entries of the remaining 2×2 nonzero block in P_{IH}:

$$(P_{IH})_{1,3} = \frac{2\left(q_{13}^2(r_{24}+q_{24}) - q_{13}(q_{14}r_{23}+r_{14}q_{23}) + (r_{13}-q_{13})q_{14}q_{23}\right)}{e+f+g+h},$$

$$(P_{IH})_{1,4} = \frac{-2\left(q_{14}^2(r_{23}+q_{23}) - q_{14}(q_{24}r_{13}+r_{24}q_{13}) + (r_{14}-q_{14})q_{13}q_{24}\right)}{e+f+g+h},$$

$$(P_{IH})_{2,3} = \frac{-2\left(q_{23}^2(r_{14}+q_{14}) - q_{23}(q_{13}r_{24}+r_{13}q_{24}) + (r_{23}-q_{23})q_{13}q_{24}\right)}{e+f+g+h},$$

$$(P_{IH})_{2,4} = \frac{2\left(q_{24}^2(r_{13}+q_{13}) - q_{24}(q_{14}r_{23}+r_{14}q_{23}) + (r_{24}-q_{24})q_{14}q_{23}\right)}{e+f+g+h}.$$

We finally come to the matrix P_{IO}, whose entries are determined completely in terms of the data. First, we list the $(3,3)$ and $(4,4)$ entries of this matrix, which are, respectively,

$$\frac{1}{\begin{vmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{vmatrix}} \left(\begin{vmatrix} q_{33} & r_{13} & r_{23} \\ q_{31} & r_{11} & r_{21} \\ q_{32} & r_{12} & r_{22} \end{vmatrix} + q_{23} \begin{vmatrix} r_{11} & q_{31} \\ r_{12} & q_{32} \end{vmatrix} + q_{13} \begin{vmatrix} r_{22} & q_{32} \\ r_{21} & q_{31} \end{vmatrix} \right)$$

and

$$\frac{1}{\begin{vmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{vmatrix}} \left(\begin{vmatrix} q_{44} & r_{14} & r_{24} \\ q_{41} & r_{11} & r_{21} \\ q_{42} & r_{12} & r_{22} \end{vmatrix} + q_{24} \begin{vmatrix} r_{11} & q_{41} \\ r_{12} & q_{42} \end{vmatrix} + q_{14} \begin{vmatrix} r_{22} & q_{42} \\ r_{21} & q_{41} \end{vmatrix} \right).$$

The entries in the remaining 2×2 block are listed next:

$$(P_{IO})_{1,1} = \frac{\begin{vmatrix} q_{11} & r_{11} & r_{21} \\ q_{13} & r_{13} & r_{23} \\ q_{14} & r_{14} & r_{24} \end{vmatrix} + r_{11}\phi + q_{11}\psi}{e+f+g+h},$$

$$(P_{IO})_{1,2} = \frac{\begin{vmatrix} q_{12} & r_{12} & r_{22} \\ q_{13} & r_{13} & r_{23} \\ q_{14} & r_{14} & r_{24} \end{vmatrix} + r_{12}\phi + q_{12}\psi}{e+f+g+h},$$

$$(P_{IO})_{2,1} = \frac{\begin{vmatrix} q_{21} & r_{11} & r_{21} \\ q_{23} & r_{13} & r_{23} \\ q_{24} & r_{14} & r_{24} \end{vmatrix} + r_{21}\phi + q_{21}\psi}{e+f+g+h},$$

$$(P_{IO})_{2,2} = \frac{\begin{vmatrix} q_{22} & r_{12} & r_{22} \\ q_{23} & r_{13} & r_{23} \\ q_{24} & r_{14} & r_{24} \end{vmatrix} + r_{22}\phi + q_{22}\psi}{e+f+g+h},$$

where

$$\phi = \begin{vmatrix} q_{13} & q_{14} \\ q_{23} & q_{24} \end{vmatrix}; \quad \psi = \begin{vmatrix} q_{14} & q_{13} \\ q_{24} & q_{23} \end{vmatrix} + \begin{vmatrix} r_{14} & r_{13} \\ q_{24} & q_{23} \end{vmatrix} + \begin{vmatrix} q_{14} & r_{24} \\ q_{13} & r_{23} \end{vmatrix}.$$

In our general procedure for constructing multiterminal networks from graphs, the more complicated the graph, the more complicated the network. For instance, for the three-point star and the square in Figure 11, we obtain the networks in Figures 12 and 13, respectively.

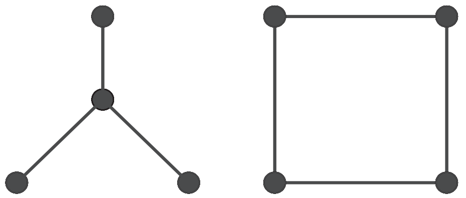

Figure 11. A three-point star and a square.

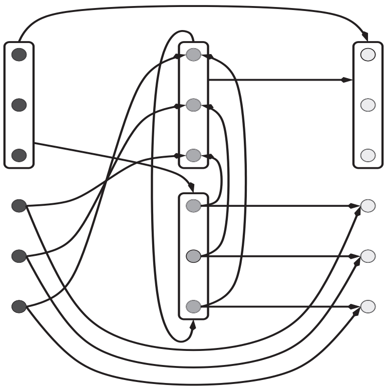

Figure 12. The network arising from a three-point star.

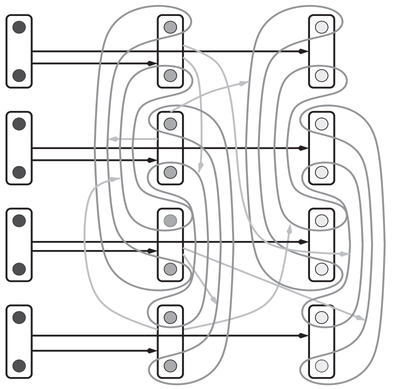

Figure 13. The network arising from a square.

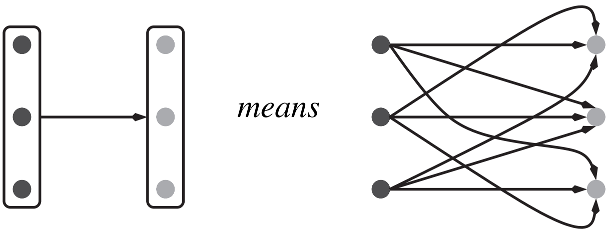

Figure 14. A convention for simplifying figures.

We used the convention depicted in Figure 14 to make Figures 12 and 13 less cluttered.

In all these examples, the same results hold; that is, we can provide explicit expressions for all the unknowns in terms of as many free parameters as there are intermediate terminals. However, the formulas are not as simple as the ones that we have shown here.

7. Limitations of the Model

It is not hard to point out shortcomings in our model as a caricature of the Internet or other realistic networks. Maybe the most obvious ones are related to the following scenario: in practice, messages carry an address, and the traffic of any message is affected by that of the others. We expect to address these issues in the future.

A technical limitation of our method is that we are restricted to multiterminal networks with a *common value* for the number of incoming, intermediate, and outgoing nodes.

8. Summary and Conclusions

The problem of determining the traffic matrix of a network from input-output measurements leads to an apparently *unsolvable* inverse problem as soon as we allow for loops.

Here we have considered a highly simplified model of a realistic multiterminal network with plenty of loops. We give a general method for producing networks of arbitrary size (including those based on a hypercube) for which we can give a complete discussion of the underlying nonlinear inverse problem. The results are illustrated by giving explicit inversion formulas (depending upon some free parameters) in the case of a small but highly nontrivial network.

In our view, the main point of the paper is that the resulting highly nonlinear equations can be *solved explicitly.* We hope to be able to extend these results to more realistic networks, and we welcome suggestions as to the kind of networks that might be of practical interest in regard to this general inversion problem.

Acknowledgements. We are grateful to Professors Zhi-Li-Zhang and Elwyn Berlekamp as well as a very careful referee for useful suggestions on ways to improve the manuscript. The first author was partially supported by NSF Grant DMS 0204682 and AFSOR Contract F41 624-02-1-7000. The second author was partially supported by an NSF Postdoctoral Research Fellowship.

References

[Arridge 99] S. Arridge. "Optical Tomography in Medical Imaging." *Inverse Problems* 15 (1999), R41–R93.

[Dorn 98] O. Dorn. "A Transport-Backtransport Method for Optical Tomography." *Inverse Problems* 14 (1998), 1107–1130.

[Grünbaum 90] F. A. Grünbaum. "Tomography with Diffusion." In *Inverse Problems in Action*, edited by P. C. Sabatier, pp. 16–21. Berlin: Springer-Verlag, 1990.

[Grünbaum 92] F. A. Grünbaum. "Diffuse Tomography: The Isotropic Case." *Inverse Problems* 8 (1992), 409–419.

[Grünbaum 01a] F. A. Grünbaum. "Diffuse Tomography: Using Time-of-Flight Information in a Two-Dimensional Model." *International J. of Imaging Technology* 11 (2001), 283–286.

[Grünbaum 01b] F. A. Grünbaum. "A Nonlinear Inverse Problem Inspired by Three-Dimensional Diffuse Tomography." *Inverse Problems* 17 (2001), 1907–1922.

[Singer et al. 90] J. Singer, F. A. Grünbaum, P. Kohn, and J. Zubelli. "Image Reconstruction of the Interior of Bodies that Diffuse Radiation." *Science* 248 (1990), 990–993.

[Grünbaum and Patch 92a] F. A. Grünbaum and S. Patch. "The Use of Grassmann Identities for Inversion of a General Model in Diffuse Tomography." In *Inverse Problems in Mathematical Physics: Proceedings of the Lapland Conference on Inverse Problems held at Saariselkä, Finland, 14–20 June 1992,* pp. 29–43, Lecture Notes in Physics 422. Berlin: Springer-Verlag, 1993.

[Grünbaum and Patch 92b] F. A. Grünbaum and S. Patch. "Simplification of a General Model in Diffuse Tomography." In *Inverse Problems in Scattering and Imaging*, edited by M. A. Fiddy, pp. 744–754. Proceedings of SPIE 1767. Bellingham, WA: SPIE, 1992.

[Grünbaum and Matusevich 02a] F. A. Grünbaum and L. F. Matusevich. "Explicit Inversion Formulas for a Model in Diffuse Tomography." *Advances in Applied Mathematics* 29 (2002), 172–183.

[Grünbaum and Matusevich 02b] F. A. Grünbaum and L. F. Matusevich. "A Nonlinear Inverse Problem Inspired by 3-Dimensional Diffuse Tomography." *Int. J. Imaging Technology* 12 (2002), 198–203.

[Grünbaum and Matusevich 04] F. A. Grünbaum and L. F. Matusevich. "A Network Tomography Problem Related to the Hypercube." *Contemporary Mathematics* 362 (2004), 189–197.

[Natterer and Wubbeling 01] F. Natterer and F. Wubbeling. *Mathematical Methods in Image Reconstruction.* Monographs on Mathematical Modeling and Computation 5. Philadelphia: SIAM, 2001.

[Patch 94] S. Patch. "Consistency Conditions in Diffuse Tomography." *Inverse Problems* 10:1 (1994), 199–212.

[Patch 95] S. Patch. "Recursive Recovery of a Family of Markov Transition Probabilities from Boundary Value Data." *J. Math. Phys.* 36:7 (1995), 3395–3412.

[Patch 99] S. Patch. "A Recursive Algorithm for Diffuse Planar Tomography." In *Discrete Tomography: Foundations, Algorithms, and Applications*, edited by G. Herman and A. Kuba, Chapter 20. Boston: Birkhauser, 1999.

[Vardi 96] Y. Vardi. "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data." *Journal of the Statistical Association* 91 (1996), 365–377.

F. Alberto Grünbaum, Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720-3840 (grunbaum@math.berkeley.edu)

Laura Felicia Matusevich, Department of Mathematics, Texas A & M University, College Station, TX 77843-3368 (laura@math.tamu.edu)

Received December 14, 2004; accepted January 24, 2006.