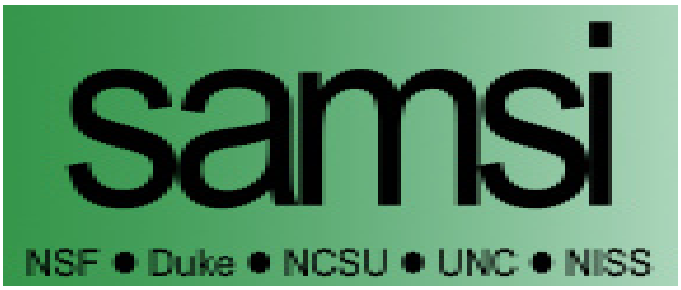

"SEQUENTIAL IMPORTANCE SAMPLING OF BINARY SEQUENCES"

Ian Dinwoodie

Technical Report #2009-4
July 26, 2010

This material was based upon work partially supported by the National Science Foundation under Grant DMS-0635449 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Statistical and Applied Mathematical Sciences Institute
PO Box 14006
Research Triangle Park, NC 27709-4006
www.samsi.info

SEQUENTIAL IMPORTANCE SAMPLING OF BINARY SEQUENCES

IAN H. DINWOODIE
DUKE UNIVERSITY AND SAMSI

Abstract. Two sequential methods are described for sampling constrained binary sequences from partial solutions. The backward method computes elimination ideals over finite fields and constructs partial solutions that extend. The forward method uses numerical global optimization to determine which partial solutions extend. The methods are applied to restricted orderings, binary dynamics, and random graphs.

1. Introduction

Sampling constrained binary sequences is useful for applications in combinatorics (Huber, 2006), statistics of social networks (Chen, 2007, and Snijders, 1991), and regulatory networks (Dinwoodie, 2008). When the sequences are 0 - 1 tables with margin constraints, sequential importance sampling (SIS) has been useful (Blitzstein and Diaconis, 2006, and Chen, Diaconis, Holmes, and Liu, 2005). In this paper we discuss general constraints, which may be called nonlinear. Examples of binary sequences with nonlinear constraints are random graphs with a fixed number of edges and triangles, constrained permutations, or any set of roots of polynomial equations of degree 2 or higher.

We give foundations for two SIS methods that are designed to solve binary sampling problems with difficult constraints. The algebraic method is based on elimination ideals (or the lexicographic Groebner basis if desired) with coefficients either a finite field or the complex numbers. The numerical method is based on global optimization and is very memory efficient for larger computations.

2. Backward Sampling

Consider a set Ω of constrained binary sequences. If the constraints are written as polynomials, then there are algebraic algorithms to eliminate the first variable in the equations defining the constraints, and derive equations in the remaining variables. The derived equations will be called the elimination ideal. Roots to the original system will solve the derived equations. Solving the system by substitution

Date: May 2009, revised July 2010.

is the method where one first solves the derived equations in one fewer variable, then "extends" the solution to the other variable. This method sometimes does not work in nonlinear systems but it will work in our binary variable setting.

Elimination ideals (p. 113, Cox, Little, O'Shea, 1997), which are the precise version of the concept of equations derived in a subset of variables, are useful for sequential sampling, because they give a step-by-step procedure to obtain a feasible sequence that only uses *past* sampled coordinate values in the sequence. This method will be called backward sequential importance sampling. This term distinguishes it from the forward numerical method of the next section, which seeks to extend partial solutions numerically at every step by determining future viability. For the algebraic backward method, the technical details of extending solutions in finite fields (which serve as values for the variables) is described in this section.

Let us consider a simple example to illustrate the two approaches before continuing with technical details. The goal is to sample uniformly from the set of permutations on three characters 1, 2, 3, with no fixed points. We will represent permutations as 3×3 0 - 1 matrices and then write the constraints as polynomials on the entries. The permutation $1 \to 2, 2 \to 3, 3 \to 1$ and the general permutation are represented with the matrices below:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{pmatrix}.$$

The matrices will represent valid constrained permutations if six equations are satisfied for row and column sums of 1, if nine other equations are satisfied to force entries to be binary ($x_1^2 - x_1 = 0$ for example, and similarly for the other variables), and if three further equations forcing the diagonal entries to 0 are included, like $x(1) = 0$. So the constraints can be written with 18 equations (or fewer with some work but it will be important to always include the ones that force entries to be binary, even if they seem redundant). Now the elimination ideals are the polynomial analog of Gaussian elimination for linear equations, a type of triangularization. Below they are given from computations in Singular in eight lines. The first line is the set of equations that involve only the last variable entry x_9 – there is only the expression "x_9" and the corresponding equation is "$x_9 = 0$". Then the next line involves the equations for both x_9 and x_8 (again, set the expressions to 0). Each line has all the equations needed after eliminating the initial variables.

$$x_9$$
$$x_9, x_8^2 - x_8$$
$$x_9, x_7 + x_8 - 1, x_8^2 - x_8$$
$$x_9, x_7 + x_8 - 1, x_6 - x_7, x_8^2 - x_8$$
$$x_9, x_7 + x_8 - 1, x_6 - x_7, x_5, x_8^2 - x_8$$
$$x_9, x_7 + x_8 - 1, x_6 - x_7, x_5, x_4 + x_7 - 1, x_8^2 - x_8$$
$$x_9, x_7 + x_8 - 1, x_6 - x_7, x_5, x_4 + x_7 - 1, x_3 + x_6 - 1, x_8^2 - x_8$$
$$x_9, x_7 + x_8 - 1, x_6 - x_7, x_5, x_4 + x_7 - 1, x_3 + x_6 - 1, x_2 + x_3 - 1, x_8^2 - x_8$$

We can solve these equations in sequence: $x_9 = 1$, and the proposal probability is 1. The next line of equations says x_8 can be 0 or 1, so we flip a coin for the value of x_8, say $x_8 = 1$, and the proposal probability is $1/2$. The next ideal $x_9, x_7+x_8-1, x_8^2-x_8$ is interesting, it forces $x_7 = 0$ with the previous choices. Then continuing through the lines, $x_6 = 0, x_5 = 0, x_4 = 1, x_3 = 1, x_2 = 0$, and finally using all the equations we get $x_1 = 0$. The product of the conditional probabilities is $1/2$, and the sampling "weight" w is then 2 (w_1 in later notation for the first sampled object).

The main theorem below says that the elimination process guarantees a valid solution, it will not get stuck. Note that the work in Chen and Liu (2007) can be applied to this type of example, but their methods do not guarantee a valid table in the end in all examples. On the other hand, their proposal probabilities are relatively sophisticated and would improve upon the naive coin flip that we use, and furthermore there are examples where the algebraic computations that we propose cannot be completed and then may not be helpful.

Now let us compare the backward sampling above with the forward sampling method developed later in Section 3. Here we convert all the equations into an objective function which is repeatedly minimized to test for solutions:

$$F(x_1, \cdots, x_9) = \sum_{i=1}^{9} (x_i^2 - x_i)^2 + x_1^2 + x_4^2 + x_9^2 + (x_1 + x_2 + x_3 - 1)^2 + \cdots$$

Now, is there any permutation with $x_9 = 1$? Let us study

$$F(x_1, \cdots, x_8, 1) = \sum_{i=1}^{8} (x_i^2 - x_i)^2 + x_1^2 + x_4^2 + 1^2 + (x_1 + x_2 + x_3 - 1)^2 + \cdots$$

This function will not have minimum 0, therefore we know that $x_9 = 1$ is not going to extend to a valid solution. The minimization is a numerical test of feasibility,

essentially an inefficient but general substitute for particular methods like the Gale-Ryser and Erdos-Gallai theorems that have been used for this purpose for sampling tables. This way of checking feasibility is not a panacea, as some constraints lead to very awkard and inefficient minimization problems, like the contingency table that we discuss in Example 2.3, but the method can work on some problems with methods of global optimization.

Now we proceed with general and more precise notation. Let $H^d = \{0,1\}^d$. The target distribution π will be uniform on the set

$$\Omega := \{\mathbf{x} \in H^d : f_j(\mathbf{x}) = 0, j = 1, \dots c\},$$

where each f_j is a polynomial. For any function f on Ω, the expectation $E_\Omega(f(X))$ will be with respect to the uniform distribution:

$$E_\Omega(f(X)) = \sum_{\mathbf{x}\in\Omega} f(\mathbf{x}) \frac{1}{|\Omega|}.$$

Other distributions are possible but all our examples will use the uniform distribution.

The set of all polynomials is written $k[s_1, \dots, s_d] = k[\mathbf{s}]$, the ring (that is, collection closed under addition and multiplication) of polynomials with coefficients in the field k and indeterminates $s_1, \dots, s_d$. The indeterminates are logically distinct from actual points (d-tuples of elements in the field k) that may be substituted into the polynomial for evaluation.

Number fields like $F_p = \{0, 1, \dots, p-1\}$ for the coefficient field k (the integers modulo a prime number p) are of practical interest because computations are faster and less demanding of machine memory than computations over the complex numbers $\mathbb{C}$, and sometimes the difference can change a computation from difficult to easy. An example below involving orderings of 12 DNA subsequences is done in F_{13}. The field F_p is not algebraically closed (for example $x^2 + x + 1$ has no roots in $F_2 = \{0, 1\}$), which brings up technical issues in the application of the Extension Theorem (p.161, Cox, Little, O'Shea, 1997), the fundamental tool for feasible sequential sampling. Theorem 1 below is essentially about checking that the coefficient field F_p works with the Extension Theorem in a specific setting for binary sequences. Now more precisely,

$$\Omega := \{\mathbf{x} \in H^d : f_j(\mathbf{x}) = 0, i = 1, \dots c\}, \quad f_j \in F_p[\mathbf{s}],\ j = 1, \dots, c.$$

The polynomials that define our domain Ω as roots generate an *ideal* – make the collection of linear combinations with polynomial coefficients to get the ideal generated by the defining polynomials. Each point in the domain Ω is also a root of everything in the ideal.

The lexicographic Groebner basis (lex basis) is a special way to write an ideal that relates to triangularizing a set of equations for back substitution, and also to a division procedure for knowing whether some polynomial vanishes on the domain Ω or not. First one defines the lexicographic order (lex order) on monomials (p. 54, Cox, Little, O'Shea, 1997), then one may proceed to compute a Groebner basis with respect to lex order for a set of polynomials (a better set of polynomials that is equivalent in terms of roots but better for the division algorithm) with an algorithm such as the Buchberger algorithm (p. 86, Cox, Little, O'Shea, 1997).

Define ideals I_F and I_{01} in $F_p[s_1, \ldots, s_d]$ by

$$I_F = \langle f_1, \ldots, f_c \rangle \tag{1}$$

$$I_{01} = \langle s_i^2 - s_i, i = 1, \ldots, d \rangle. \tag{2}$$

(The notation $\langle f, g, h \rangle$ means the ideal generated by polynomials f, g, h, which is all the linear combinations with polynomial coefficients.) The set Ω may be considered a set of constrained 0 - 1 tables, with nonlinear polynomial constraints generalizing margin constraints.

The proposal distribution, from which we generate an i.i.d. sample $(X_i, i = 1, \ldots, n)$ in Ω, will be close to uniform. The proposal distribution q will be expressed as a product of successive conditional distributions

$$q(\mathbf{x}) = q_d(x_d) \cdot q_{d-1}(x_{d-1}|x_d) \cdot q_{d-2}(x_{d-2}|x_d, x_{d-1}) \cdots q_1(x_1|x_d, \ldots, x_2)$$

just as a random point $(X_{i,1}, X_{i,2}, \ldots, X_{i,d}) \in \Omega$ will be generated sequentially: $X_{i,d}, X_{i,d-1}, \ldots, X_{i,2}, X_{i,1}$.

The unnormalized weights w_i are defined by $w_i = 1/q(X_i)$. The SIS Monte Carlo estimate for $E_\Omega(f(X))$ is given by

$$\hat{E}_\Omega(f(X)) := \frac{1}{n} \sum_{i=1}^{n} f(X_i) \frac{w_i}{\bar{w}}. \tag{3}$$

The law of large numbers says that

$$\bar{w} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{q(X_i)} \to \sum_{\mathbf{x}} q(\mathbf{x})/q(\mathbf{x}) = |\Omega| \tag{4}$$

almost surely as $n \to \infty$. Thus SIS can be used for approximate counting, which we pursue in Example 2.2 to estimate the size of a basin of attraction in nonlinear dynamics.

The SIS procedure for sampling on Ω with an initial Groebner basis computation is described next.

Backward Sequential Importance Sampling (BSIS) on Ω:

(0) Compute a reduced lexicographic Groebner basis for $I_\Omega := I_F + I_{01}$ with variable order $s_1 > s_2 > \cdots > s_d$ in $F_p[\mathbf{s}]$. If $I_\Omega = \langle 1 \rangle$, then Ω is empty and stop. Otherwise Ω is not empty and continue.

(1) For sample size n, let the index i run from 1 to n:

(a) Using the polynomials from the lex basis that only involve s_d, determine which of $\{0, 1\}$ solve the system and let $n_d \in \{1, 2\}$ be the number of values in $\{0, 1\}$ that solve the equations. Then uniformly sample $X_{i,d}$ from the set of roots, and let $q_d(X_{i,d}) = 1/n_d$.

(b) Continue for coordinate indices $d-1, d-2, \ldots, 1$ to count (by substitution of 0 and 1) the number of solutions n_{d-j} to the equations (modulo p) in the lex basis that involve variables $\{s_{d-j}, \ldots, s_d\}$, with $s_{d-j+1} = X_{i,d-j+1}, \ldots, s_d = X_{i,d}$. Choose $X_{i,d-j}$ uniformly from the n_{d-j} solutions, and set

$$q_{d-j}(X_{i,d-j}|X_{i,d-j+1}, \ldots, X_{i,d}) = 1/n_{d-j}.$$

(c) Complete $(X_{i,1}, \ldots, X_{i,d}) \in \Omega$ when $X_{i,1}$ is chosen and $q_1(X_{i,1}|X_{i,d}, \ldots, X_{i,2})$ is computed.

(d) Set $X_i = (X_{i,1}, \ldots, X_{i,d})$ and $l_i = -\log(q_d(X_d)) - \cdots - \log(q_1(X_1|X_d, \ldots, X_2))$.

Recall that $I_F + I_{01}$ is simply the ideal of combined generators from both I_F and I_{01}. Also, note that in our setup there is always at least one polynomial that involves only s_d as stated in 1(a), because $s_d^2 - s_d \in I_\Omega \cap k[x_d]$, the elimination ideal. The following result justifies the method above and uses the explicit form of I_{01} to prove the sequential solution.

Theorem 1. *BSIS always produces an element $X_i \in \Omega$ if $I_\Omega \neq \langle 1 \rangle$, and the importance sampling weights w_i are*

$$w_i = e^{l_i}$$

where the values l_i are defined in step 1 (d) of BSIS.

Proof. If $I_\Omega = \langle 1 \rangle$, then there are no solutions to the polynomials in (1) and (2), so $\Omega = \emptyset$. Conversely, if $I_\Omega \neq \langle 1 \rangle$, then a reduced Groebner basis in $\bar{F}_p[\mathbf{s}]$ (polynomials with coefficients in the algebraic closure of F_p) is also not $\langle 1 \rangle$ (see Lemma 2.4.16, Kreuzer and Robbiano, 2000 for the notion of "field of definition" and Groebner basis computations in field extensions). Then Hilbert's Nulstellensatz (p. 168, Cox, Little, O'Shea, 1997) applied to I_Ω as a subset of $\bar{F}_p[\mathbf{s}]$ says that there exist roots to the polynomials in expressions (1) and (2), with values in $\bar{F}_p$. Now equations (2) force the solution to be 0 or 1, hence solutions are in F_p. Thus $I_\Omega = \langle 1 \rangle$ if and only if Ω is empty.

Assume Ω is not empty. Write $I_\Omega = \langle g_1, \ldots, g_G \rangle$ and assume these polynomials are a lexicographic Groebner basis, which always exists. Suppose we have a partial

solution $(X_{d-k+1}, \ldots, X_d)$ – that is, a solution in F_p^k for some $k = 1, \ldots, d-1$ to the equations in $I_\Omega \cap F_p[s_{d-k+1}, \ldots, s_d]$. This can always be extended another dimension to a solution $(a_{d-k}, X_{d-k+1}, \ldots, X_d) \in \bar{F}_p^{k+1}$, because the polynomial $s_{d-k}^2 - s_{d-k} \in I_\Omega \cap F_p[s_{d-k}, \ldots, s_d]$ and the Extension Theorem (Theorem 3, p. 115, Cox, Little, O'Shea, 1997) says the extension is possible with the presence of such a univariate polynomial. Now the new coordinate a_{d-k} satisfies all equations that involve indeterminates $s_{d-k}, \ldots, s_d$, so in particular it satisfies $s_{d-k}^2 - s_{d-k}$, and thus its value is 0 or 1. So it belongs to F_p and the extended solution is in F_p^{k+1} and the algebraic closure is not needed. The actual choice X_{d-k} is made by choosing from the possible values of a_{d-k}.

The weights w_i are defined by $w_i = 1/q(X_i)$ where q is the proposal distribution. The sequential construction of X_i shows that

$$q(X_i) = q_d(X_{i,d}) \cdot q_{d-1}(X_{i,d-1}|X_{i,d}) \cdots q_1(X_{i,1}|X_{i,d}, \ldots, X_{i,2}).$$

This gives weight

$$\begin{aligned} w_i &= e^{-\sum_{k=0}^{d-1} \log q_{d-k}(X_{i,d-k}|X_{i,d},\ldots,X_{i,d-k+1})} \\ &= e^{l_i}. \end{aligned}$$

This completes the proof. □

Rather than compute the lex Groebner basis, one can recursively compute the elimination ideals $I_0 := I_\Omega$, and $I_i = I_{i-1} \cap k[s_{i+1}, \ldots, s_d]$ – the polynomials that do not involve indeterminates $s_1, \ldots, s_i$ – and use this collection in step 1 b) above. This is computationally more efficient than computing the complete lex basis since other term orders can be used, and most importantly the required extension property of partial solutions still holds.

A diagnostic measure of efficiency of importance sampling is the quantity cv^2 given by the sample variance of the normalized weights: $cv^2 = \text{var}(\{w_i/\bar{w}\})$. It has been argued that $1/(1 + cv^2)$ can be used as a measure of efficiency of the SIS estimator relative to i.i.d. sampling from π (p. 35, Liu, 2001). However this heuristic is not completely reliable and correct variance estimates are useful (see p. 17, Blitzstein and Diaconis, 2006 for a discussion of variance and SIS examples).

Statistical problems with restricted permutations arise in many applications, including permutation tests (Diaconis, Graham, Holmes, 2001), sampling or counting linear extensions (Matthews, 1991, and Huber, 2006), and topological ordering (p. 673, Roberts and Tesman, 2005). Restricted permutations on C characters can be represented as $C \times C$ 0 - 1 matrices that are roots of polynomials in $F_p[s_1, \ldots, s_d]$ ($d = C \times C$ and the prime p is chosen to satisfy $p > C$) that make row and column sums equal to 1 and force other restrictions. We illustrate the method on a problem of sequencing by hybridization modified from Roberts and Tesman (2005) and from

Pevzner (2000). The algebraic approach, while not optimal in terms of computational complexity because of the variable elimination step, applies in theory to any problem of restricted permutations.

Example 2.1. The problem of sequence hybridization is described in Roberts and Tesman (p. 674, 2005). It is about finding possible sequences of DNA from partial subsequences and reduces to one of finding Hamiltonian chains. We will show how the BSIS method applies for counting and sampling restricted permutations.

We modify slightly the original problem to include missing data in order to make the problem harder and to show the flexibility of the algebraic approach (which also works on the original data). We are given twelve subsequences:

$$*A*, CAC, ACG, CGC, GCA, CAA, AAC, ACT, CTT, TTA, TAA, AAA.$$

The goal is to order them so that the last two characters of the subsequence in position 1 are the same as the first two characters of the subsequence in position 2, and likewise for all positions. For example, the transition CAC to ACG is possible, but AAA to TAA is not. Above, * is a single character wildcard that indicates missing data. One can make a directed graph with vertices labelled by the strings, and edges E corresponding to possible transitions. Then we seek an ordering of the 12 sequences $S_1 = *A*, S_2 = CAC, \ldots, S_{12} = AAA$, which is a one-to-one map from $\{1, 2, \ldots, 12\}$ onto $\{S_1, \ldots, S_{12}\}$ where transitions are only permitted along edges. The adjacency matrix A_E for the graph is $A_E =$

	$*C*$	CAC	ACG	CGC	GCA	CAA	AAC	ACT	CTT	TTA	TAA	AAA
$*C*$	0	1	0	1	0	1	0	0	1	0	0	0
CAC	1	0	1	0	0	0	0	1	0	0	0	0
ACG	0	0	0	1	0	0	0	0	0	0	0	0
CGC	1	0	0	0	1	0	0	0	0	0	0	0
GCA	0	1	0	0	0	1	0	0	0	0	0	0
CAA	0	0	0	0	0	0	1	0	0	0	0	1
AAC	1	0	1	0	0	0	0	1	0	0	0	0
ACT	0	0	0	0	0	0	0	0	1	0	0	0
CTT	0	0	0	0	0	0	0	0	0	1	0	0
TTA	0	0	0	0	0	0	0	0	0	0	1	0
TAA	0	0	0	0	0	0	1	0	0	0	0	1
AAA	0	0	0	0	0	0	1	0	0	0	0	0

We code a map $f : \{1, \ldots, 12\} \to \{S_1, \ldots, S_{12}\}$ as a 12×12 0 - 1 matrix, with row i indicating the value of $f(i)$ with a 1 in position j for $f(i) = S_j$. The bijective property holds if row sums and column sums are 1, mod $p = 13$. Compatibility of transitions with the adjacency matrix A_E is obtained by making each pair $(f(1), f(2)), (f(2), f(3)), \ldots, (f(11), f(12))$ an edge, or more precisely by excluding all other transitions that are not on edges.

Let $(s_{i,j})$ be a 12×12 matrix of 144 indeterminates, where $s_{i,j}$ will indicate whether $f(i) = j$ or not, and let $s_{i,\star}$ be the i^{th} row. Call the ideal of row sums

generated by 12 linear polynomials I_r (which includes twelve polynomials like $s_{1,1}+\cdots s_{1,12}-1$ for the first row), and the ideal of column sums I_c (which includes twelve polynomials like $s_{1,1}+\cdots+s_{12,1}-1$). Form the monomial ideals $I_{1,2}, I_{2,3}, \ldots, I_{11,12}$ by setting

$$I_{1,2} = \langle s_{1,\star} \otimes s_{2,\star} \cdot (\mathbf{1} - A_E) \rangle$$
$$\cdots$$
$$I_{11,12} = \langle s_{11,\star} \otimes s_{12,\star} \cdot (\mathbf{1} - A_E) \rangle$$

where $\mathbf{1}$ is the 12×12 matrix with all entries 1, whose purpose is to flip the edge matrix A_E. That is, $s_{1,\star} \otimes s_{2,\star}$ is a 12×12 matrix of products like $s_{1,i}s_{2,j}$, and the "$\cdot$" operation is an entrywise product, giving a final 12×12 matrix of 0s and monomials $s_{1,i}s_{2,j}, (i,j) \neq E$. Then $I_{1,2}$ is the ideal of these entries in the final matrix. Then roots $\mathbf{x}$ to the polynomials in

$$I := \langle s_{i,j}^2 - s_{i,j} \rangle + I_r + I_c + I_{1,2} + \cdots + I_{11,12},$$

which is the collection of all the polynomials from each term in the sum, are all the orderings that we seek.

Note that in some examples there can be cyclical symmetry. With subsequences AAA, AAC, ACA, CAA for example, the algebra will correctly count four orderings corresponding to the four starting points of a cycle.

The elimination ideals for I can be computed (Singular writes a file of elimination ideals of 400 kbytes). Sequential sampling was done for five samples of size 100 each on a desktop computer with 3.33 GHz processor, and each sample required a few seconds of actual time. The average of the estimates was 21.92, with standard deviation 0.253 of the five values. The cv^2 values were around 0.13 on all runs. The algebra is quite demanding but subsequent sampling is very fast. The equations in I can be solved with the numerical solver in Singular (with the library solve.lib) to confirm 22 solutions, but this requires a longer computing time of several minutes (which however is shorter than the time required to get the elimination ideals). An algorithm for finding *all* solutions rather than just one is not given in Roberts and Tesman (2005).

The BSIS procedure can be used to estimate the size of basins of attraction for fixed points in nonlinear dynamics, such as the one in Example 2.2 below, a method which we now describe. Let $F : H^d \to H^d$ be a map, and let $\mathbf{w} \in H^d$ satisfy $F(\mathbf{w}) = \mathbf{w}$. Such a point $\mathbf{w}$ is called a steady-state or fixed point for F. In regulatory network applications, certain fixed points are biologically realistic, while others are not. The set of points that eventually hit a realistic fixed point $\mathbf{w}$,

called basin of attraction, is of special interest, and it is defined precisely as the set $F^{-\infty}(\mathbf{w}) := \{\mathbf{x} : F^{\infty}(\mathbf{x}) = \mathbf{w}\} = \cup_{k=1}^{\infty}\{\mathbf{x} : F^k(\mathbf{x}) = \mathbf{w}\}$. In some cases the basin of attraction can be identified explicitly, as in Mendoza and Alvarez-Buylla (1998), leading to precise size computations. Other methods for estimating the size are in Albert and Othmer (2003). The size may be a small fraction of the total size of H^d (a fraction around 10^{-4} for certain wild-type steady states $\mathbf{w}$ in the *Drosophila* network of Albert and Othmer, 2003), and this makes it hard to access by direct sampling in H^d and rejecting points not in the basin of attraction.

Let $\mathbf{w}$ be a fixed point for the map F and for $k = 1, 2, \ldots$ let $\sigma_k := |\{\mathbf{x} \in H^d : F^k(\mathbf{x}) = \mathbf{w}, F^{k-1} \neq \mathbf{w}\}|$. We set $\sigma_0 = 1$, corresponding to the point $\mathbf{w}$ that hits $\mathbf{w}$ in 0 iterations. Now

$$|\{\mathbf{x} : F^k(\mathbf{x}) = \mathbf{w}\}| = \sigma_0 + \sigma_1 + \ldots + \sigma_k$$

and we next give an estimate for each σ_k using importance sampling.

Suppose the map $F = (f_1, \ldots, f_d)$ is in the form where each coordinate map $f_j : H^d \to \{0, 1\}$ is a polynomial in indeterminates $s_1, \ldots, s_d$. This form can always be achieved. With indeterminates $z, s_1, \ldots, s_d, t_1, \ldots, t_d$ in a polynomial ring with complex coefficients $\mathbb{C}$ and fixed point $\mathbf{w} = (w_1, \ldots, w_d) \in H^d$, form the ideal I:

$$\begin{aligned} I =& \langle f_1(\mathbf{s}) - t_1, \ldots, f_d(\mathbf{s}) - t_d, z \cdot (s_1 - w_1 + \cdots + s_d - w_d) - 1\rangle \\ &+ \langle s_1^2 - s_1, \ldots, s_d^2 - s_d, t_1^2 - t_1, \ldots, t_d^2 - t_d\rangle. \end{aligned}$$

This ideal relates to solutions of $F(\mathbf{s}) = \mathbf{t}$ which are 0 - 1 valued and not equal to the fixed point $\mathbf{w}$. Let $I_1, \ldots, I_d$ be the ideals that eliminate z, then z, s_1, and so on until I_d only involves variables $s_d, t_1, \ldots, t_d$. The variables $t_1, \ldots, t_d$ will represent the target state in sampling.

For $i = 1, \ldots, n$, use BSIS to get a backward sequence of up to k states, and call the sequence S_i. Each S_i is a matrix with $k + 1$ columns and d rows that hold elements of H^d, and are of the form

$$S_i = \begin{cases} (\mathbf{x}_k, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_1, \mathbf{w}) & L(S_i) = k \\ (\cdots, \emptyset, \mathbf{x}_m, \cdots, \mathbf{x}_1, \mathbf{w}) & L(S_i) = m < k \end{cases}$$

with $\mathbf{x}_j \neq \mathbf{w}, \mathbf{x}_{j+1} \in F^{-1}(\mathbf{x}_j)$. That is, the rightmost column in S_i is the fixed point $\mathbf{w}$. The second column from the right is a point $\mathbf{x}_1 \in F^{-1}(\mathbf{w})$ ($\mathbf{x}_1$ depends on the sample index i which we have suppressed for simplicity), obtained using BSIS after setting $\mathbf{t} = \mathbf{w}$ in the polynomial equations. Note that $\mathbf{x}_1 \neq \mathbf{w}$, since we have excluded this solution with the equation that involves the z indeterminate. The probability of $\mathbf{x}_1$ is $q_{i,1}$, from the BSIS algorithm. Now use BSIS to find a point $\mathbf{x}_2 \in F^{-1}(\mathbf{x}_1)$, by setting $\mathbf{t} = \mathbf{x}_1$ and following the BSIS procedure through

the d coordinates, with final probability $q_{i,2}$. If no such point exists, set $\mathbf{x}_2 = \emptyset$ and $q_{i,2} = 1$. Continue until the matrix S_i has $k+1$ columns, $\mathbf{x}_1 \in F^{-1}(\mathbf{w})$, $\mathbf{x}_{j+1} \in F^{-1}(\mathbf{x}_j)$ if $F^{-1}(\mathbf{x}_j) \neq \emptyset$ for $j = 1, \ldots, k-1$. Then set

$$q_k(S_i) = \prod_{j=1}^{L(S_i)} q_{i,j} \tag{5}$$

where $L(S_i)$ is the length of the backward orbit before termination at the empty set. That is, if $F^{-1}(\mathbf{x}_1) = \emptyset$, then $L(S_i) = 1$, and always $0 \leq L(S_i) \leq k$. Then $q_k(S_i)$ is the probability of sampling the backward path S_i, of length at most k. Below I_k is the indicator function for the integer k.

Proposition 1. *Let $k > 0$, and let $q_k(S_i)$ be the probabilities in expression (5) and let $L(S_i)$ be the backward sequence length of S_i. Then*

$$\frac{1}{n} \sum_{i=1}^{n} \frac{I_k(L(S_i))}{q_k(S_i)} \to \sigma_k \tag{6}$$

as $n \to \infty$.

Proof. A matrix S_i with $L(S_i) = k$ is a full matrix of the form like $(\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_1, \mathbf{w})$, with $F(\mathbf{x}_{j+1}) = \mathbf{x}_j, j = 1, \ldots, k-1, F(\mathbf{x}_1) = \mathbf{w}$. Such S_i are in one-to-one correspondence with the last value $\mathbf{x}_k \in F^{-k}(\mathbf{w})$: if two sequences S_i, S_i' have $\mathbf{x}_k = \mathbf{x}_k'$, then the entire sequences will be the same clearly and thus $S_i = S_i'$. Also, $\mathbf{x}_k \in F^{-k}(\mathbf{w}) \setminus F^{-(k-1)}(\mathbf{w})$, since if $F^{k-1}(\mathbf{x}_k) = \mathbf{w}$, then $\mathbf{x}_1 = F^{k-1}(\mathbf{x}_k) = \mathbf{w}$, a contradiction with the construction. Therefore, elements S_i with $L(S_i) = k$ are in one-to-one correspondence with the set $F^{-k}(\mathbf{w}) \setminus F^{-(k-1)}(\mathbf{w})$.

Let

$$\mathcal{S}_k = \{S = (\mathbf{x}_k, \ldots, \mathbf{x}_1, \mathbf{w}) : F(\mathbf{x}_{j+1}) = \mathbf{x}_j, \mathbf{x}_j \neq \mathbf{w}\}$$

which we have just seen is in one-to-one correspondence with $F^{-k}(\mathbf{w}) \setminus F^{-(k-1)}(\mathbf{w})$.

By the law of large numbers,

$$\frac{1}{n} \sum_{i=1}^{n} \frac{I_k(L(S_i))}{q_k(S_i)} \to \sum_{S \in \mathcal{S}_k} \frac{1}{q_k(S)} q_k(S) = |F^{-k}(\mathbf{w}) \setminus F^{-(k-1)}(\mathbf{w})| = \sigma_k.$$

This completes the proof. □

The SIS approach in formula (6) not only estimates the size of the basin attraction, it also generates a sample that may be used for further analysis conditional on the basin of attraction. However, the expression (6) can be unstable, presumably because sometimes very small values of $q_k(S_i)$ arise and give very large incorrect numbers. Such outliers can be recognized by very high cv^2 values. Modifications of the estimator to limit the effect of small denominators would be interesting. To use the result to estimate $|F^{-k}(\mathbf{w})|$, one must do k simulations to get estimates

$\hat{\sigma}_1, \ldots, \hat{\sigma}_k$ and add them with $\sigma_0 = 1$. We remark finally that since the quantities $|F^{-k}(\mathbf{w})|$ are increasing in k and converge, one may estimate the size of the basin of attraction $|F^{-\infty}(\mathbf{w})|$ by increasing k experimentally to achieve a good approximation. The estimates for σ_k can have more variability for larger k because of the longer random product $q_k(S_i)$ in the denominator.

Example 2.2. Aracena (p. 1406, 2008) presents an example of network dynamics with a large number of fixed points. Setting $n = 21$ (n being his notation for number of nodes), we have 21 binary maps given by

```
f1=x(2)
f2=x(21)*x(1)
f3=x(4)
f4=x(21)*x(3)
f5=x(6)
f6=x(21)*x(5)
f7=x(8)
f8=x(21)*x(7)
f9=x(10)
f10=x(21)*x(9)
f11=x(12)
f12=x(21)*(x(11))
f13=x(14)
f14=x(21)*(x(13))
f15=x(16)
f16=x(21)*x(15)
f17=x(18)
f18=x(21)*x(17)
f19=x(20)
f20=x(21)*x(19)
f21=1-((1-x(2))*(1-x(4))*(1-x(6))*(1-x(8))*(1-x(10))*(1-x(12))*(1-x(14))*
    (1-x(16))*(1-x(18))*(1-x(20)))
```

We have found $1023 = 2^{(21-1)/2} - 1$ fixed points by numerical solution of the fixed point equation in Singular, not the 1024 that seem to be predicted in Aracena. Our problem is to measure the size of the basin of attraction for one of the 1023 points, namely the fixed point $\mathbf{0}$. Using Proposition 2.1, we use BSIS with a sample size of $n = 1000$ to get estimates of 1010.7 and 1047.5 for $k = 1$ (with cv^2 values of 1.03 and 0.95). A larger sample size of $n = 5000$ required over 10 minutes in R on a 3.33 GHz desktop, and gave an estimate of 1020.5 with $cv^2 = 1.01$ for example. These estimates show a reasonable standard error. For $k = 2$, the estimates for σ_2 are all 0. Then we would estimate $|F^{-\infty}(\mathbf{0})| \approx 1 + 1010$. Calculations by hand show 1025 points exactly (which is only possible to do because of the special simple form of the equations in this particular example), and this number stabilizes at $k = 1$.

Practical networks such as Albert and Othmer (2003) will not be possible to do by hand but such lightly coupled networks can be analyzed with computer algebra.

Example 2.3. Here we consider the Darwin finch data, one of the motivating examples of Chen, Diaconis, Holmes and Liu (2005). The 12×17 table with 0 - 1 entries has row sums 14, 13, 14, 10, 12, 2, 10, 1, 10, 11, 6, 2, and column sums 3, 3, 10, 9, 9, 7, 8, 9, 7, 8, 2, 9, 3, 6, 8, 2, 2, with a "1" entry indicating presence of finch variety (row variable) on a particular island (column variable). The goal is to count and/or sample uniformly tables with the same row and column sums.

The BSIS method applies using any prime $p > 17$ for the number field F_p of coefficients. We have only been able to compute the elimination ideals modulo the prime $p = 2$ at this time, which corresponds to finding tables with the same margins up to parity. Here the problem is easy and reduces to a certain hypercube of size 9.6×10^{52}, a number which is found by SIS with sample sizes as small as 500, and in fact can be deduced by examining the reduced lexicographic basis (which has only 204 elements). The modulo 2 version of the problem is related to the "satisfiability" problem of Boolean expressions (Motwani and Raghavan, 1995). Rubinstein (2010) has Monte Carlo methods for related problems.

3. Forward Sampling

The forward sampling method uses global optimization as a tool to look forward to see which possible current states 0 or 1 will lead to a feasible full sequence. The global minimum is the substitute for the elimination ideal of backward sampling. The backward algebraic sampling method requires large amounts of memory in one initial hard computation, whereas the forward numerical method uses little computer memory but does many minimizations in the course of the sampling. The global minimization steps are done numerically with a certain tolerance, resulting in a method that runs on large problems with little memory use, but one which may give samples with some variability in quality. The forward numerical method can be distributed over many processors very simply, unlike the backward algebraic method.

Global optimization methods attempt to minimize a real-valued function over $\mathbb{R}^d$ from any initial point and without convexity assumptions. Nonmonotone line search methods often play a key role (Grippo, Lampariello, and Lucidi, 1986), but other methods are also possible (Breiman and Cutler, 1993). The one we used in numerical examples is based on the method of LaCruz, Martinez, and Raydan (2006), a development of the Barzilai-Borwein spectral method, which is refined and implemented in the R package BB (Varadhan and Gilbert, 2008). This particular implementation does not require a differentiable objective function.

Recall that we are seeking to sample uniformly from the binary solutions to equations $f_1(\mathbf{x}), \ldots, f_c(\mathbf{x}) = 0, \mathbf{x} \in H^d$, solutions which make the set Ω. Define the function

$$F(\mathbf{x}) = \sum_{i=1}^{d} (x_i^2 - x_i)^2 + \sum_{j=1}^{c} (f_j(\mathbf{x}))^2, \ \mathbf{x} \in \mathbb{R}^d.$$

Forward Sequential Importance Sampling (FSIS) on Ω:

(0) Identify one point $\mathbf{x}_0 \in \Omega$.

(1) For sample size n, let the index i run from 1 to n:

(a) Let $F_0(\mathbf{y}) = F(y_1, \ldots, y_{d-1}, 0)$, and let $F_1(\mathbf{y}) = F(y_1, \ldots, y_{d-1}, 1)$. Let $m_0 = \min_{\mathbf{y} \in \mathbb{R}^{d-1}} F_0, m_1 = \min_{\mathbf{y} \in \mathbb{R}^{d-1}} F_1$. Let $n_d \in \{1, 2\}$ count how many of $\{m_0, m_1\}$ are 0. Then uniformly sample $X_{i,d}$ from the subset of $\{0, 1\}$ that includes 0 if $m_0 = 0$, and 1 if $m_1 = 0$. Let $q_d(X_{i,d}) = 1/n_d$.

(b) For $j = 1, \ldots, d-1$, set

$$F_0(\mathbf{y}) = F(y_1, \ldots, y_{d-j-1}, 0, X_{i,d-j+1}, \ldots, X_{i,d})$$
$$F_1(\mathbf{y}) = F(y_1, \ldots, y_{d-j-1}, 1, X_{i,d-j+1}, \ldots, X_{i,d})$$

Let $m_0 = \min_{\mathbf{y} \in \mathbf{R}^{d-j-1}} F_0, m_1 = \min_{\mathbf{y} \in \mathbf{R}^{d-j-1}} F_1$. Let $S = \{k : m_k = 0\}$, and set $n_{d-j} = |S| \in \{1, 2\}$. Choose $X_{i,d-j}$ uniformly from S. Set

$$q_{d-j}(X_{i,d-j} | X_{i,d-j+1}, \ldots, X_{i,d}) = 1/n_{d-j}.$$

(c) When $j = d - 1$, F_0 and F_1 do not involve any y variables. After checking which of F_0 or F_1 vanishes to get possible states S, we sample $X_{i,1}$ from S and form the complete sequence $(X_{i,1}, \ldots, X_{i,d})$.

(d) Set $X_i = (X_{i,1}, \ldots, X_{i,d})$, and $l_i = -\log(q_d(X_d)) - \cdots - \log(q_1(X_1 | X_d, \ldots, X_2))$.

Note that if there is a point $\mathbf{x}_0 \in \Omega$, the number of root choices n_d (and later n_{d-j}) is always positive in theory, either 1 or 2. In practice, there may be some errors in the numerical minimization routine that cause difficulties. Other weighted sums, possibly using box constraints $[0, 1]$ on the variables, with minimizers in Ω could be used rather than F, some of which may improve the performance of minimizing algorithms.

Theorem 2. *FSIS always produces an element $X_i \in \Omega$ if Ω is not empty, and the importance sampling weights w_i are*

$$w_i = e^{l_i}.$$

Proof. We are assuming that there is a point $\mathbf{x}_0 \in \Omega$, so at the first step with $F_0(\mathbf{y}) = F(y_1, \ldots, y_{d-1}, 0)$, and $F_1(\mathbf{y}) = F(y_1, \ldots, y_{d-1}, 1)$, at least one of $\{m_0, m_1\}$ is 0, so $n_d > 0$. Thus we will get a partial solution X_d after sampling from the n_d possibilities.

Suppose we have a partial solution $(X_{d-k+1}, \ldots, X_d)$ – that is, a solution in $\{0,1\}^k$ for some $k = 1, \ldots, d-1$ to $\min F(y_1, \ldots, y_{d-k}, X_{d-k+1}, \ldots, X_d) = 0$. This can always be extended another dimension to a solution $(a_{d-k}, X_{d-k+1}, \ldots, X_d) \in \{0,1\}^{k+1}$, because

$$0 = \min_y F(y_1, \ldots, y_d) = \min_y F(y_1, \ldots, y_{d-k}, X_{d-k+1}, \ldots, X_d)$$

and the minimizing value(s) satisfies $y_{d-k}^2 - y_{d-k} = 0$, and thus its value is 0 or 1.

The weights w_i are defined by $w_i = 1/q(X_i)$ where q is the proposal distribution. The sequential construction of X_i shows that

$$q(X_i) = q_d(X_{i,d}) \cdot q_{d-1}(X_{i,d-1}|X_{i,d}) \cdots q_1(X_{i,1}|X_{i,d}, \ldots, X_{i,2}).$$

This gives weight

$$\begin{aligned} w_i &= e^{-\sum_{k=0}^{d-1} \log q_{d-k}(X_{i,k}|X_{i,d},\ldots,X_{i,d-k+1})} \\ &= e^{l_i}. \end{aligned}$$

This is the proof.

□

Note that the algorithm is only sure to work if Ω is assumed to be nonempty, whereas BSIS can determine algebraically whether Ω is nonempty. Step 1 b) is hard, and Theorem 3.1 assumes that it is done perfectly. In practice, there will be some numerical error in the minimization. By allowing more numerical error in the computation, the speed of the method can be improved, at the cost of getting points X_i that only satisfy the constraints on Ω approximately. (For example, in the `BBoptim` function from the R package BB (Varadhan and Gilbert, 2010), one can adjust the `gtol` parameter in the control list that is the threshold to conclude gradient 0, which by default is 10^{-5}, and one may also increase the maximum number of iterations from the default of 1500. Also, for difficult problems one may repeat a minimization at different starting points for verification.)

Example 3.1. Let us consider the network of mutually "known" researcher in the EIES.1 data set from the SIENA research project webpage (http://stat.gamma.rug.nl/siena.html) (this relationship is coded as a "2", and is the mutual presence of the condition "heard about the other, did not meet him/her"). In this example we present a notion of conditional parameter significance. The conditional method will be applicable in other examples where existing computational methods fail. A network where the ergm software (Handcock, Hunter *et al.*, 2009) has difficulty, and thus one where the conditional approach we describe below may be essential, is the network of mutual friends in the same data set EIES.1, for which we were not able to get fitted parameter values.

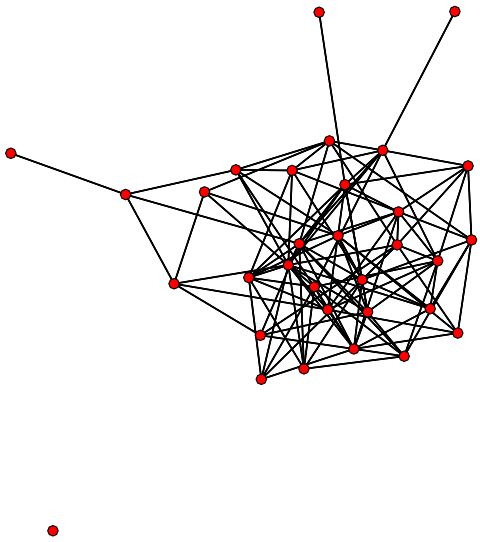

FIGURE 1. EIES network of 32 mutually known researchers

There are 32 people represented as nodes in the network graph y_0 (constructed with the "network" R package of Butts, Hunter, and Handcock, 2008). Note that the network nodes should be labelled, and the labels are left out of Figure 1.

We will consider a three parameter model that uses edge count, triangle count, and the "alternating k-star" statistic $A(y)$. The alternating k-star statistic is described on page 198 in Robins, Snijders, *et al.* (2006), and our goal will be to judge its significance conditional on the number of edges and triangles in the data.

If y is a symmetric 0 - 1 adjacency matrix with no loops, then y_{ij} is 1 if and only if there is an edge between nodes i and j. Let y_{i+} be the number of edges attached

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1	0	1	0	1	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0
3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	0	0
4	1	0	0	0	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	0
5	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	1	0	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	0	0	1	1	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0
12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1	0	0	0
13	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
14	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0
15	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0
16	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
17	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
18	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
19	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0
20	1	1	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
21	0	0	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0
22	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0
24	1	1	1	1	1	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
25	1	0	0	1	0	0	0	1	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
27	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0
29	1	1	0	1	1	0	0	1	0	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	1	0	1	0	1	1	0
30	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
31	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
32	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 2. EIES matrix of 32 mutually known researchers

to node i and define graph statistics:

$$E(y) = \sum_{1 \le i < j \le 32} y_{ij}$$

$$T(y) = \sum_{1 \le i < j < h \le 32} y_{ij} y_{ih} y_{jh}$$

$$A(y) = \sum_{2 \le k \le 31} (-1/2)^{k-2} \left(\sum_{i=1}^{32} \binom{y_{i+}}{k} \right)$$

A 3-parameter network model is

$$q_{\eta,\tau,\alpha}(y) = e^{\eta E(y) + \tau T(y) + \alpha A(y) - \psi(\eta,\tau,\alpha)}. \tag{7}$$

We are interested in computing the significance of the alternating k-star term $A(y)$. Fitting the model with the ergm function of the ergm R package gives an estimate $\hat{\alpha} = 1.68$, standard error from the estimate of the covariance matrix of

3.49 and reported two-sided p-value of $0.630 = 2 \times P(Z > 1.68/3.49)$. However, these numbers change somewhat from one fitting try to the next. Another problem with the ergm fitting procedure is that two different standard error estimates are given – the one from the covariance estimate above is much different than the one that is presented as "MCMC s.e." in the same table of output. For example, "MCMC s.e." is given as 0.128 in the same fitting, giving p-value of practically 0.0. Therefore the p-value is quite ambiguous from the ergm fitting procedure, and an exact counterpart is useful. Our results support a smaller p-value than 0.630.

To measure the significance of $A(y)$ in the model, we will use the conditional method: obtain a p-value for $A(y)$ using the conditional distribution of $A(y)$ given the observed values of $E(y_0)$ and $T(y_0)$. This is a common method for analyzing sparse discrete data where asymptotics are doubtful (see Agresti, Mehta, Patel, 1990, for a presentation), and may give p-values much larger or smaller than asymptotic methods. The conditional distribution is uniform on networks with the same number of edges (113) and triangles (81), but with importance sampling we would use the weighted estimate

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{A(y_i)\geq A(y_0)\}}\frac{w_i}{\bar{w}} \tag{8}$$

for the one-sided p-value $\mu(\{y : A(y) \geq A(y_0) \mid E(y) = 113, T(y) = 81)\})$, where $y_1, \ldots, y_n$ are the random tables produced with FSIS, and $A(y_0) = 335.5$. The two-sided p-value would be double this quantity.

The sampling can be done with the FSIS algorithm, using two ($c = 2$) polynomial constraints $E(y)-113 = 0, T(y)-81 = 0$ (one linear and one cubic), and $d = \binom{32}{2} = 496$ binary variables. On a run with sample size 200 (which took over 24 hours in R (2009) on a 3.00 GHz desktop computer) we estimated the two-sided p-value at .026. A bootstrap sample drawn with weights w_i (and therefore approximately uniform in distribution) from the simulation sample gave 2.5% and 97.5% percentiles for $A(y)$ of $(330.6, 334.6)$, which support the p-value. Other runs were consistent with a p-value smaller than 0.630.

4. Conclusions

The problem of sampling binary sequences with constraints arises in many applications. In this paper we have given two methods for general nonlinear constraints which can be useful when more specialized efficient methods for contingency tables do not apply. The algebraic method requires an initial algebraic calculation of elimination ideals, and sequential sampling follows with one-variable distributions that are updated from past sampled values (Backward SIS). The algebraic computation does not generally scale well in the number of variables, but how well it works

depends very much on the particular equations. Less coupled equations (systems where each equation depends on only a few variables) are easier for the computer algebra and some examples can be done with over one thousand variables.

An alternative is Forward SIS, where numerical optimization determines which candidate states for the present coordinate will complete to feasible sequences. This method scales better and uses very little computer memory, but numerical tolerances for speed can result in slightly imperfect sample statistics. The minimization algorithms work better with fewer constraints. This method can be used for conditional inference on social network data where the number of variables is in the hundreds. Speed improvements in both methods would be valuable and both methods will work better as algebra and optimization software improve.

Acknowledgments. We thank Giovanni Pistone for discussions on algebra and sampling. This material was based upon work partially supported by the National Science Foundation under Grant DMS-0635449 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Admiraal, R., and Handcock, M. S.: networksis: A Package to Simulate Bipartite Graphs with Fixed Marginals Through Sequential Importance Sampling. *Journal of Statistical Software*, **24**, 1-21 (2008).

Agresti, A., Mehta, C. R., and Patel, N. R.: Exact inference for contingency tables with ordered categories. *Journal of the American Statistical Association*, **85**, 453-458 (1990).

Albert, R., and Othmer, H. G.: The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *Journal of Theoretical Biology*, **223**, 1-18 (2003).

Aracena, J.: Maximum number of fixed points in regulatory Boolean networks. *Bulletin of Mathematical Biology*, **70**, 1398-1409 (2008).

Blitzstein, J., and Diaconis, P. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. Stanford University Technical Report (2006).

Breiman, L., and Cutler, A.: A deterministic algorithm for global optimization. *Mathematical Programming*, **58**, 179-199 (1993).

Butts, C.T., Handcock, M. S., and Hunter, D. R.: network: Classes for Relational Data. R package version 1.4-1. Irvine, CA. http://statnet.org/ (2008).

Chen, Y.: Conditional inference on tables with structural zeros. *Journal of Computational and Graphical Statistics*, **16**, 445-467 (2007).

Chen, Y., and Liu, J. S.: Sequential Monte Carlo methods for permutation tests on truncated data, *Statistica Sinica*, **17**, 857-872 (2007).

Chen, Y., Diaconis, P., Holmes, S. P., and Liu, J. S.: Sequential Monte Carlo Methods for Statistical Analysis of Tables, *Journal of the American Statistical Association*, **100**, 109 - 120 (2005).

Cox, D., Little, J., O'Shea, D.: *Ideal, Varieties, and Algorithms, Second Edition.*, Springer, New York (1997).

Diaconis, P., Graham, R., and Holmes, S. P.: Statistical problems involving permutations with restricted positions. *State of the art in probability and statistics (Leiden, 1999)*, 195-222, IMS Lecture Notes Monogr. Ser., 36, Inst. Math. Statist., Beachwood OH (2001).

Dinwoodie, I. H.: Polynomials for classification trees and applications. SAMSI Technical Report 2008-7 (2008).

Goodreau, S. M.: Advances in exponential random graph (p^*) models applied to a large social network. *Social Networks*, **29**, 231-248 (2007).

Greuel, G.-M., Pfister, G., and Schönemann, H.: SINGULAR 3.0.4. *A Computer Algebra System for Polynomial Computations.* Centre for Computer Algebra, University of Kaiserslautern. URL http://www.singular.uni-kl.de (2007).

Grippo, L., Lampariello, F., and Lucidi, S.: A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis*, **23**, 707-716 (1986).

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau S. M., and Morris, M.: ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks. Version 2.2-1. Project home page at http://statnetproject.org. URL http://CRAN.R-project.org/package=ergm (2009).

Huber, M.: Fast perfect sampling from linear extensions. *Discrete Mathematics*, **306**, 420-428 (2006).

Kreuzer, M., and Robbiano, L.: *Computational Commutative Algebra I.* Springer, New York (2000).

LaCruz, W., Martinez, J. M., and Raydan, M.: Spectral residue method without gradient information for solving large-scale nonlinear systems of equations. *Mathematics of Computation*, **75**, 1429-1448 (2006).

Liu, J. S.: *Monte Carlo Strategies in Scientific Computing.* Springer, New York (2001).

Matthews, P.: Generating a random linear extension of a partial order. *Annals of Probability*, **19**, 1367-1392 (1991).

Mendoza, L., and Alvarez-Buylla, E. R.: Dynamics of the genetic regulatory network for *Arabidopsis thaliana*. *Journal of Theoretical Biology.* **193**, 307-319 (1998).

Motwani, R., and Raghavan, P. *Randomized Algorithms.* Cambridge University Press, NY (1995).

Pevzner, P. A.: *Computational Molecular Biology.* MIT Press, Cambridge MA (2000).

Pistone, G., Riccomagno, E., and Wynn, H.: *Algebraic Statistics: Computational Commutative Algebra in Statistics.* Chapman and Hall, New York (2001).

R Development Core Team: *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org (2009).

Roberts, F. S., and Tesman, B.: *Applied Combinatorics*, Pearson Prentice Hall, Upper Saddle River NJ (2005).

Robins, G., Snijders, T., Wang, P., Handcock, M., and Pattison, P.: Recent developments in exponential random graph (p^*) models for social networks. *Social Networks*, **29**, 192-215 (2007).

Rubinstein, R.: Randomized algorithms with splitting: why the classic randomized algorithms do not work and how to make them work, *Methodology and Computing in Applied Probability*, **12**, 1-50 (2010).

Snijders, T. A. B.: Enumeration and Simulation Methods for 0 - 1 Matrices With Given Marginals, *Psychometrika*, **56**, 397-417 (1991).

Varadhan, R., and Gilbert, P. D.: BB: An R Package for Solving a Large System of Nonlinear Equations and for Optimizing a High-Dimensional Nonlinear Objective Function (version 2010.4-1), *Journal of Statistical Software*, **32**, 1-26 (2009).