SIAM REVIEW
Vol. 46, No. 4, pp. 667–689

Fastest Mixing Markov Chain on a Graph*

Stephen Boyd†
Persi Diaconis‡
Lin Xiao§

Abstract. We consider a symmetric random walk on a connected graph, where each edge is labeled with the probability of transition between the two adjacent vertices. The associated Markov chain has a uniform equilibrium distribution; the rate of convergence to this distribution, i.e., the mixing rate of the Markov chain, is determined by the second largest eigenvalue modulus (SLEM) of the transition probability matrix. In this paper we address the problem of assigning probabilities to the edges of the graph in such a way as to minimize the SLEM, i.e., the problem of finding the fastest mixing Markov chain on the graph.

We show that this problem can be formulated as a convex optimization problem, which can in turn be expressed as a semidefinite program (SDP). This allows us to easily compute the (globally) fastest mixing Markov chain for any graph with a modest number of edges (say, 1000) using standard numerical methods for SDPs. Larger problems can be solved by exploiting various types of symmetry and structure in the problem, and far larger problems (say, 100,000 edges) can be solved using a subgradient method we describe. We compare the fastest mixing Markov chain to those obtained using two commonly used heuristics: the maximum-degree method, and the Metropolis–Hastings algorithm. For many of the examples considered, the fastest mixing Markov chain is substantially faster than those obtained using these heuristic methods.

We derive the Lagrange dual of the fastest mixing Markov chain problem, which gives a sophisticated method for obtaining (arbitrarily good) bounds on the optimal mixing rate, as well as the optimality conditions. Finally, we describe various extensions of the method, including a solution of the problem of finding the fastest mixing reversible Markov chain, on a fixed graph, with a given equilibrium distribution.

Key words. Markov chains, second largest eigenvalue modulus, fast mixing, semidefinite programming, subgradient method

AMS subject classifications. 60J22, 60J27, 65F15, 65K05, 90C22, 90C46

DOI. 10.1137/S0036144503423264

1. Introduction.

1.1. Fastest Mixing Markov Chain Problem.

1.1.1. Markov Chain on an Undirected Graph. We consider a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = \{1, \ldots, n\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, with $(i, j) \in \mathcal{E} \Leftrightarrow$

*Received by the editors February 24, 2003; accepted for publication (in revised form) April 7, 2004; published electronically October 29, 2004. This research was sponsored in part by NSF grant ECE-0140700, AFOSR grant F49620-01-1-0365, and DARPA contract F33615-99-C-3014.
http://www.siam.org/journals/sirev/46-4/42326.html

†Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305-4065 (boyd@stanford.edu). Authors listed alphabetically.
‡Department of Statistics and Department of Mathematics, Stanford University, Stanford, CA 94305-4065.
§Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9510 (lxiao@stanford.edu).

$(j,i) \in \mathcal{E}$. We assume that each vertex has a self-loop, i.e., an edge from itself to itself: $(i,i) \in \mathcal{E}$ for $i = 1, \ldots, n$.

We define a discrete-time Markov chain on the vertices of the graph as follows. The state at time t will be denoted $X(t) \in \mathcal{V}$ for $t = 0, 1, \ldots$. Each edge in the graph is associated with a transition probability, with which X makes a transition between the two adjacent vertices. These edge probabilities must be nonnegative, and the sum of the probabilities of edges connected to each vertex (including the self-loop) must be 1. Note that the probability associated with self-loop (i,i) is the probability that $X(t)$ stays at vertex i.

We can describe this Markov chain via its transition probability matrix $P \in \mathbf{R}^{n\times n}$, where

$$P_{ij} = \mathbf{Prob}(X(t+1) = j \mid X(t) = i), \quad i,\ j = 1, \ldots, n.$$

This matrix must satisfy

$$P \geq 0, \qquad P\mathbf{1} = \mathbf{1}, \qquad P = P^T, \tag{1}$$

where the inequality $P \geq 0$ means elementwise, i.e., $P_{ij} \geq 0$ for $i, j = 1, \ldots, n$, and $\mathbf{1}$ denotes the vector of all ones. The condition (1) is simply that P must be symmetric and doubly stochastic; it must also satisfy

$$P_{ij} = 0, \quad (i,j) \notin \mathcal{E}, \tag{2}$$

which states that transitions are allowed only between vertices that are linked by an edge.

Let $\pi(t) \in \mathbf{R}^n$ be the probability distribution of the state at time t: $\pi_i(t) = \mathbf{Prob}(X(t) = i)$. The state distribution satisfies the recursion $\pi(t+1)^T = \pi(t)^T P$, so the distribution at time t is given by

$$\pi(t)^T = \pi(0)^T P^t.$$

Since P is symmetric and $P\mathbf{1} = \mathbf{1}$, we conclude that $\mathbf{1}^T P = \mathbf{1}^T$, so the uniform distribution $(1/n)\mathbf{1}$ is an equilibrium distribution for the Markov chain. If the chain is irreducible and aperiodic (the case we will focus on in this paper), then the distribution $\pi(t)$ converges to the unique equilibrium distribution $(1/n)\mathbf{1}$ as t increases.

1.1.2. SLEM, Mixing Rate, and Mixing Time. We are concerned with the rate of convergence of $\pi(t)$ to the uniform distribution, which is determined by the eigenstructure of the probability transition matrix P. The eigenvalues of P are real (since it is symmetric), and by Perron–Frobenius theory, no more than 1 in magnitude. We will denote them in nonincreasing order:

$$1 = \lambda_1(P) \geq \lambda_2(P) \geq \cdots \geq \lambda_n(P) \geq -1.$$

The asymptotic rate of convergence of the Markov chain to the uniform equilibrium distribution is determined by the second largest eigenvalue modulus (SLEM) of P:

$$\mu(P) = \max_{i=2,\ldots,n} |\lambda_i(P)| = \max\{\lambda_2(P),\ -\lambda_n(P)\}.$$

The smaller the SLEM, the faster the Markov chain converges to its equilibrium distribution.

There are several well-known specific bounds on the convergence of the state distribution to uniform. One of these is given in terms of the *total variation* distance between two distributions ν and $\tilde{\nu}$ on $\mathcal{V}$, defined as the maximum difference in probability assigned to any subset:

$$\|\nu - \tilde{\nu}\|_{\text{tv}} = \max_{\mathcal{S} \subseteq \mathcal{V}} \left| \mathbf{Prob}_{\nu}(S) - \mathbf{Prob}_{\tilde{\nu}}(S) \right| = \frac{1}{2} \sum_i |\nu_i - \tilde{\nu}_i|$$

(see, e.g., [13, section 4.1.1]). We have the following bound on the total variation distance between $\pi(t)$ and the uniform distribution [20, Prop. 3]:

$$\sup_{\pi(0)} \left\| \pi(t) - \frac{1}{n}\mathbf{1} \right\|_{\text{tv}} = \frac{1}{2} \max_i \sum_j \left| P_{ij}^t - \frac{1}{n} \right| \leq \frac{1}{2}\sqrt{n}\mu^t.$$

If the Markov chain is irreducible and aperiodic, then $\mu(P) < 1$ and the distribution converges to uniform asymptotically as μ^t. We call the quantity $\log(1/\mu)$ the *mixing rate*, and $\tau = 1/\log(1/\mu)$ the *mixing time*. The mixing time τ gives an asymptotic measure of the number of steps required for the total variation distance of the distribution from uniform to be reduced by the factor e. If the SLEM is very close to 1, the mixing rate $\log(1/\mu)$ is approximately $1 - \mu$, which is often referred to as the *spectral gap* in the literature.

The mixing rate, mixing time, and the spectral gap can all serve as the measure for fast mixing. Since they are all monotone in the SLEM, we will focus on the SLEM in this paper. For background on Markov chains, eigenvalues, and fast mixing, see, e.g., [13].

1.1.3. Fastest Mixing Markov Chain Problem. In this paper we consider the following problem: Find the edge transition probabilities that give the fastest mixing Markov chain, i.e., that minimize the SLEM $\mu(P)$. This can be posed as the following optimization problem:

$$\begin{array}{lll} \text{minimize} & \mu(P) & \\ \text{subject to} & P \geq 0, \quad P\mathbf{1} = \mathbf{1}, \quad P = P^T, \\ & P_{ij} = 0, \quad (i,j) \notin \mathcal{E}. \end{array} \tag{3}$$

Here P is the optimization variable, and the graph is the problem data. We call this problem the *fastest mixing Markov chain* (FMMC) problem.

We denote the optimal SLEM (which is a function of the graph) as $\mu^\star$:

$$\mu^\star = \inf\{\mu(P) \mid P \geq 0,\ P\mathbf{1} = \mathbf{1},\ P = P^T,\ P_{ij} = 0,\ (i,j) \notin \mathcal{E}\}.$$

Since μ is continuous and the set of possible transition matrices is compact, there is at least one optimal transition matrix $P^\star$, i.e., one for which $\mu(P^\star) = \mu^\star$.

There are several other equivalent ways to formulate the FMMC problem. For example, we can take the edge probabilities as optimization variables, impose the constraints that they be nonnegative, and sum to no more than 1 at each vertex (see section 5).

1.2. Two Simple Heuristic Methods. Several simple methods have been proposed to obtain transition probabilities that give (it is hoped) fast mixing, if not the fastest possible.

1.2.1. The Maximum-Degree Chain. Let d_i be the degree of vertex i, not counting the self-loop; i.e., d_i is the number of neighbor vertices of vertex i, not counting i itself. Let $d_{\max} = \max_{i\in\mathcal{V}} d_i$ denote the maximum degree of the graph. The *maximum-degree chain* is obtained by assigning probability $1/d_{\max}$ to every edge except the self-loops, and choosing the self-loop probabilities to ensure that the sum of the probabilities at each vertex is 1. The maximum-degree transition probability matrix P^{md} is given by

$$P_{ij}^{\mathrm{md}} = \begin{cases} 1/d_{\max} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j, \\ 1 - d_i/d_{\max} & \text{if } i = j, \\ 0 & \text{if } (i,j) \notin \mathcal{E}. \end{cases}$$

For regular bipartite graphs, this construction just gives the usual random walk which is periodic and has -1 as an eigenvalue.

1.2.2. The Metropolis–Hastings Chain. A slightly more sophisticated heuristic can be constructed by applying the *Metropolis–Hastings algorithm* [36, 24] to a random walk on a graph. The transition probabilities of the simple random walk on a graph are given by

$$P_{ij}^{\mathrm{rw}} = \begin{cases} 1/d_i & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

This Markov chain (the base chain) is in general not symmetric; its equilibrium distribution is proportional to the degree.

To obtain a reversible Markov chain with a given equilibrium distribution $\pi = (\pi_1, \ldots, \pi_n)$ based on this random walk, we start by setting $R_{ij} = (\pi_j P_{ji}^{\mathrm{rw}})/(\pi_i P_{ij}^{\mathrm{rw}})$. The Metropolis–Hastings algorithm modifies the transition probabilities as

$$P_{ij}^{\mathrm{mh}} = \begin{cases} P_{ij}^{\mathrm{rw}} \min\{1, R_{ij}\} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j, \\ P_{ii}^{\mathrm{rw}} + \sum_{(i,k)\in\mathcal{E}} P_{ik}^{\mathrm{rw}} \left(1 - \min\{1, R_{ik}\}\right) & \text{if } i = j. \end{cases}$$

(See [8] for a nice geometric interpretation of the Metropolis–Hastings algorithm.) If π is the uniform distribution, then the transition probability matrix P^{mh} is symmetric and can be simplified as

$$P_{ij}^{\mathrm{mh}} = \begin{cases} \min\{1/d_i, 1/d_j\} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j, \\ \sum_{(i,k)\in\mathcal{E}} \max\{0,\ 1/d_i - 1/d_k\} & \text{if } i = j, \\ 0 & \text{if } (i,j) \notin \mathcal{E}. \end{cases}$$

In other words, the transition probability between two distinct, connected vertices is the reciprocal of the larger degree, and the self-loop probabilities are chosen to ensure that the sum of the probabilities at each vertex is 1.

An interesting property of the Metropolis–Hastings chain is that the transition probability on an edge only depends on local information, i.e., the degrees of its two adjacent vertices.

1.3. Applications and Previous Work. Determining or bounding the SLEM of Markov chains is very important in Markov chain Monte Carlo simulation, which is a powerful algorithmic paradigm in statistics, physics, chemistry, biology, computer science, and many other fields (see, e.g., [35, 13, 49]). The chief application of Markov

chain simulation is to the random sampling of a huge state space (often with combinatorial structure) with a specified probability distribution. The basic idea is to construct a Markov chain that converges asymptotically to the specified equilibrium distribution. Then, starting from an arbitrary state, simulate the chain until it is close to equilibrium, and the distribution of the states will be close to the desired distribution.

The efficiency of such algorithms depends on how fast the constructed Markov chain converges to the equilibrium, i.e., how fast the chain mixes. An efficient algorithm can result only if the number of simulation steps is reasonably small, which usually means dramatically less than the size of the state space itself. For example, a Markov chain is called *rapidly mixing* if the size of state space is exponential in some input data size, whereas the mixing time is bounded by a polynomial.

Most previous work focused on bounding the SLEM (or the spectral gap) of a Markov chain with various techniques and developing some heuristics to assign transition probabilities to obtain faster mixing Markov chains. Some well-known analytic approaches for bounding the SLEM are: coupling methods and strong stationary times [1, 16], conductance [29], geometric bounds [20], and multicommodity flows [48, 30]. Diaconis and Saloff-Coste [19] surveyed what is rigorously known about running times of the Metropolis–Hastings algorithm—the most celebrated algorithm for constructing Markov chains in Monte Carlo methods. Kannan [31] surveyed the use of rapidly mixing Markov chains in randomized polynomial time algorithms to approximately solve certain counting problems. More background on fast mixing Markov chains can be found in, e.g., [13, 4, 44] and references therein.

In this paper, we show that the fastest mixing Markov chain on a given graph can be computed *exactly* by a polynomial time optimization algorithm. In practice, this is feasible at least for graphs with a modest number of edges, such as 1000. We also give a subgradient method that can solve large problems with up to 100,000 edges. Although these sizes are still far smaller than the sizes that arise in practical Monte Carlo simulations, the convex optimization formulation and associated duality theory offer the potential of deriving improved bounds for the SLEM. We also hope that the FMMC solution for small size graphs can give insight into how to improve the efficiency of practical Markov chain Monte Carlo simulations.

On the other hand, many practical applications have a very rich combinatorial structure that can be exploited to greatly reduce the solution complexity of the FMMC problem. The paper [10] and the work in progress [42] study in detail the FMMC problem on graphs with rich symmetry properties.

1.4. Outline. In section 2, we show that the FMMC problem can be cast as a convex optimization problem, and even more specifically, as a semidefinite program (SDP). In section 3, we give some numerical examples of the FMMC problem and show that the fastest mixing chain is often substantially faster than the maximum-degree chain and the Metropolis–Hastings chain. In section 4, we describe the Lagrange dual of the FMMC problem and give the (necessary and sufficient) optimality conditions. In section 5, we describe a subgradient method that can be used to solve larger FMMC problems. In section 6, we generalize the FMMC problem to reversible Markov chains. In section 7, we discuss some extensions of the FMMC problem. In particular, we briefly discuss how to exploit graph symmetry to simplify computation, give some bounds relating the spectral gaps of different Markov chains on the same graph, and show that the log-Sobolev constant, another measure of fast mixing, is concave in the transition probability matrix.

2. Convex Optimization and SDP Formulation of FMMC.

2.1. Convexity of SLEM. We first show that the SLEM μ is a convex function of P, on the set of doubly stochastic symmetric matrices. There are several ways to establish this. Our first proof uses the variational characterization of eigenvalues (see, e.g., [27, section 4.2]). Since $\lambda_1(P) = 1$, with associated eigenvector $\mathbf{1}$, we can express the second largest eigenvalue as

$$\lambda_2(P) = \sup\{u^T P u \mid \|u\|_2 \leq 1,\ \mathbf{1}^T u = 0\}.$$

This shows that $\lambda_2(P)$ is the pointwise supremum of a family of linear functions of P (i.e., $u^T P u$), and so is a convex function of P (see, e.g., [43, section 5] and [12, section 3.2]). Similarly, the negative of the smallest eigenvalue,

$$-\lambda_n(P) = \sup\{-u^T P u \mid \|u\|_2 \leq 1\},$$

is also convex. Therefore, the SLEM $\mu(P) = \max\{\lambda_2(P), -\lambda_n(P)\}$, which is the pointwise maximum of two convex functions, is convex.

We can also derive convexity of μ from known results for eigenvalues of symmetric matrices. The sum of any number of the largest eigenvalues of a symmetric matrix is a convex function of the matrix (see, e.g., [15, 40]). In particular, the function $\lambda_1 + \lambda_2$ is convex for general symmetric matrices. Since our matrices are stochastic, we have $\lambda_1 = 1$, so we conclude that $\lambda_2 = (\lambda_1 + \lambda_2) - 1$ is a convex function. This has also been observed in [22].

We can also show convexity of μ by expressing it as the spectral norm of P restricted to the subspace $\mathbf{1}^\perp = \{u \in \mathbf{R}^n \mid \mathbf{1}^T u = 0\}$:

$$\mu(P) = \|(I - (1/n)\mathbf{1}\mathbf{1}^T)P(I - (1/n)\mathbf{1}\mathbf{1}^T)\|_2 = \|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2. \tag{4}$$

Here the matrix $I - (1/n)\mathbf{1}\mathbf{1}^T$ gives the orthogonal projection on $\mathbf{1}^\perp$, and $\|\cdot\|_2$ denotes the spectral norm, or maximum singular value. (In this case, since the matrices are symmetric, $\|\cdot\|_2$ is the largest eigenvalue magnitude.) The formula (4) gives $\mu(P)$ as the norm of an affine function of P, and so is a convex function (see, e.g., [12, section 3.2]).

2.2. Convex Optimization Formulation. The FMMC problem (3) is evidently a convex optimization problem, since the constraints are all linear equalities or inequalities, and the objective function is convex. Using the expression (4) we can formulate the FMMC problem as

$$\begin{array}{ll} \text{minimize} & \mu(P) = \|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2 \\ \text{subject to} & P \geq 0, \quad P\mathbf{1} = \mathbf{1}, \quad P = P^T, \\ & P_{ij} = 0, \quad (i,j) \notin \mathcal{E}, \end{array} \tag{5}$$

i.e., a norm minimization problem over a set of symmetric stochastic matrices.

We can add any convex constraints to the FMMC problem and still have a convex problem that is efficiently solvable. One interesting case is the *local degree* FMMC problem, where we require that the transition probability on an edge must depend on the degrees of its two end vertices. This problem can be viewed as a generalization of the Metropolis–Hastings chain, in which the edge probability is the minimum of the inverses of the two degrees. To formulate the local degree FMMC problem, we simply add the (linear equality) constraints that require the probabilities to be equal for any two edges that have identical degrees at adjacent vertices.

2.3. Semidefinite Programming Formulation. We can express the FMMC problem (5) as an SDP by introducing a scalar variable s to bound the norm of $P - (1/n)\mathbf{1}\mathbf{1}^T$:

$$\begin{array}{ll} \text{minimize} & s \\ \text{subject to} & -sI \preceq P - (1/n)\mathbf{1}\mathbf{1}^T \preceq sI, \\ & P \geq 0, \quad P\mathbf{1} = \mathbf{1}, \quad P = P^T, \\ & P_{ij} = 0, \quad (i,j) \notin \mathcal{E}. \end{array} \tag{6}$$

Here the variables are the matrix P and the scalar s. The symbol $\preceq$ denotes matrix inequality; i.e., $X \preceq Y$ means $Y - X$ is positive semidefinite. (The symbol $\leq$ is used to denote elementwise inequality.)

The problem (6) is not in one of the standard forms for SDP, but is readily transformed to a standard form, in which a linear function is minimized, subject to a linear matrix inequality (the constraint that an affine symmetric matrix-valued function be positive semidefinite) and linear equality constraints. The inequalities in (6) can be expressed as a single linear matrix inequality (LMI),

$$\operatorname{diag}\left(P - (1/n)\mathbf{1}\mathbf{1}^T + sI,\ sI - P + (1/n)\mathbf{1}\mathbf{1}^T,\ \mathbf{vec}(P)\right) \succeq 0.$$

Here, $\operatorname{diag}(\cdot)$ forms a block diagonal matrix from its arguments, and $\mathbf{vec}(P)$ is a vector containing the $n(n+1)/2$ different coefficients in P. Since a block diagonal matrix is positive semidefinite if and only if its blocks are positive semidefinite, this single LMI is equivalent to the inequalities in (6). (See [54, 12] for more details of such transformations.)

The FMMC problem can also be expressed in several equivalent SDP forms, using the characterization of the sum of the largest eigenvalues given in Overton and Womersley [40]; see also Alizadeh [2, section 4].

The SDP formulation (6) has several important ramifications. First, it means that the FMMC problem can be solved efficiently (and globally) using standard SDP solvers, at least for small or medium size problems (with number of edges up to a thousand or so). A custom-designed SDP solver for the FMMC problem, which exploits the special structure of the problem (e.g., sparsity of P), would be able to solve even larger problems. Detailed accounts of interior-point algorithms for SDP, as well as its many applications, can be found in, for example, Nesterov and Nemirovskii [38, section 6.4], Alizadeh [2], Vandenberghe and Boyd [54], Ye [57], Wolkowicz, Saigal, and Vandenberghe [55], Todd [52], and Ben-Tal and Nemirovski [5, section 4]. Benson, Ye, and Zhang [6] exploited the structure and sparsity of some large-scale SDPs arising in combinatorial optimization with a dual scaling method.

Another consequence of the SDP formulation is that we can easily obtain the dual of the FMMC problem, via standard SDP duality theory, as well as a set of necessary and sufficient optimality conditions. (These appear in section 4.)

3. Examples. In this section we give several numerical examples, comparing the fastest mixing chain (obtained via SDP) to the maximum-degree and Metropolis–Hastings chains.

3.1. Some Small Examples. We first consider the four simple small graphs shown in Figure 1. For each graph, we consider the maximum-degree chain, the Metropolis–Hastings chain, and the optimal (fastest mixing) chain, obtained by solving an SDP (for these examples, exactly). Table 1 shows the SLEMs of the three Markov chains for each graph, as well as the transition probability matrices of the

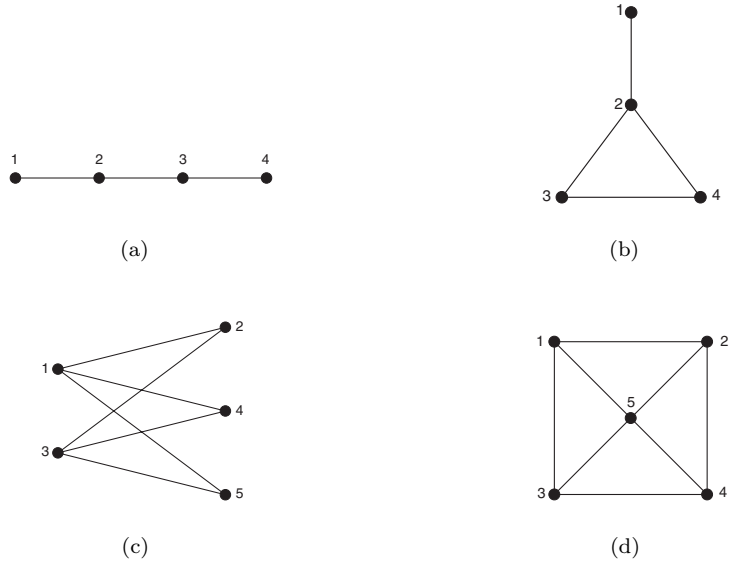

Fig. 1 *Four small graphs.*

Table 1 *SLEMs and optimal transition matrices for the small graphs.*

Graph	μ^{md}	μ^{mh}	$\mu^\star$	Optimal transition matrix $P^\star$
(a)	$\sqrt{2}/2$	$\sqrt{2}/2$	$\sqrt{2}/2$	$\begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$
(b)	2/3	2/3	7/11	$\begin{bmatrix} 6/11 & 5/11 & 0 & 0 \\ 5/11 & 0 & 3/11 & 3/11 \\ 0 & 3/11 & 4/11 & 4/11 \\ 0 & 3/11 & 4/11 & 4/11 \end{bmatrix}$
(c)	2/3	2/3	3/7	$\begin{bmatrix} 1/7 & 2/7 & 0 & 2/7 & 2/7 \\ 2/7 & 3/7 & 2/7 & 0 & 0 \\ 0 & 2/7 & 1/7 & 2/7 & 2/7 \\ 2/7 & 0 & 2/7 & 3/7 & 0 \\ 2/7 & 0 & 2/7 & 0 & 3/7 \end{bmatrix}$
(d)	1/4	7/12	1/4	$\begin{bmatrix} 1/4 & 1/4 & 1/4 & 0 & 1/4 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 \\ 1/4 & 0 & 1/4 & 1/4 & 1/4 \\ 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 \end{bmatrix}$

fastest mixing chains. Note that for graphs (b) and (c), neither the maximum-degree nor the Metropolis–Hastings chain gives the smallest SLEM.

We should note that the fastest mixing chain, i.e., the solution to the FMMC problem, need not be unique. For example, another optimal solution for graph (b)

can be obtained by changing the lower two by two diagonal block to

$$\begin{bmatrix} 3/11 & 5/11 \\ 5/11 & 3/11 \end{bmatrix}.$$

Any convex combination of these two sub-blocks would also provide an optimal chain.

3.2. Random Walk on Contingency Tables. We consider the set of all n_r by n_c matrices with nonnegative integer entries and fixed row and column sums, i.e., the set

$$\mathcal{X} = \{X \in \mathbf{Z}_+^{n_r \times n_c} \mid X\mathbf{1} = r, \quad X^T\mathbf{1} = c\},$$

where $r \in \mathbf{Z}_+^{n_r}$ is the vector of fixed row sums and $c \in \mathbf{Z}_+^{n_c}$ is the vector of fixed column sums. We construct a graph with $\mathcal{X}$ as the vertex set. We say that two tables (matrices) X and $\tilde{X}$ (both in $\mathcal{X}$) are adjacent (connected by an edge) if

$$X - \tilde{X} = (e_i - e_j)(e_k - e_l)^T \tag{7}$$

for some

$$1 \le i, j \le n_r, \quad 1 \le k, l \le n_c, \quad i \ne j, \quad k \ne l,$$

where e_i denotes the ith standard unit vector. (The first two vectors, e_i and e_j, have dimension n_r; the second two, e_k and e_l, have dimension n_c.)

Equation (7) can be explained by a random walk on the contingency tables. Given any $X \in \mathcal{X}$, we randomly pick a pair of rows and a pair of columns, and modify the four intersecting entries by

$$\begin{matrix} +1 & -1 \\ -1 & +1 \end{matrix} \quad \text{or} \quad \begin{matrix} -1 & +1 \\ +1 & -1 \end{matrix}$$

with probability 1/2 each. This modification doesn't change the row and column sums. If the modification results in negative entries, we discard it (and stay at the current table); otherwise, we accept the modification (jump to an adjacent table). We then repeat the process, by selecting new pairs of rows and columns. This describes a Markov chain on $\mathcal{X}$, and it can be shown that this chain (graph) is connected. Actually, this is precisely the maximum-degree Markov chain on the graph of contingency tables, and it generates uniform sampling of the tables in the steady state. See [17] for a review of the background and applications of contingency tables.

While the maximum-degree chain seems to be the only practical method that can be implemented to carry out uniform sampling on a set of contingency tables, it is of academic interest to compare its mixing rate with that of the Metropolis–Hastings chain and the FMMC. Even this comparison can only be done for very small tables with small row and column sums, due to the rapid growth of the number of tables and transitions when the table size, or the row and column sums, are increased.

As an example, consider matrices in $\mathbf{Z}_+^{3\times 3}$ that have fixed row sums $r = (3, 4, 5)$ and column sums $c = (3, 3, 6)$. There are 79 such matrices (nodes of the graph; one example given in Table 2) and 359 allowed transitions (edges not counting self-loops). The maximum degree of the graph is $d_{\max} = 18$ and the minimum degree is $d_{\min} = 4$. We found that the SLEMs of the three Markov chains are $\mu^{\mathrm{md}} = 0.931$, $\mu^{\mathrm{mh}} = 0.880$, and $\mu^\star = 0.796$, respectively.

Table 2 *A small contingency table and its row sums and column sums.*

1	1	1	3
1	1	2	4
1	1	3	5
3	3	6	

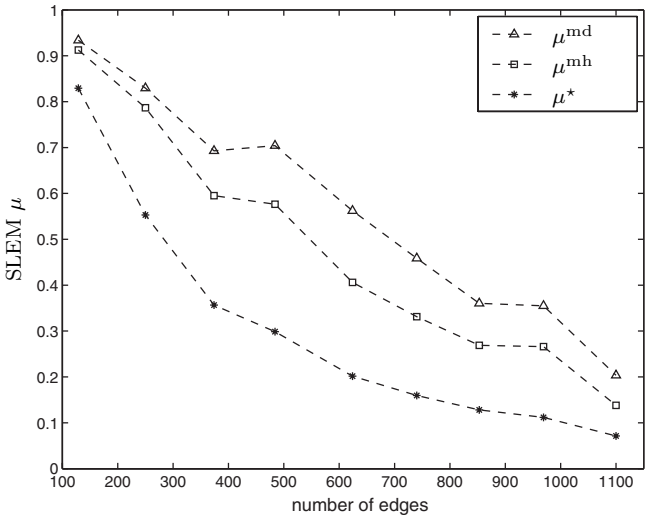

Fig. 2 *SLEMs of three Markov chains on a family of nine randomly generated graphs.*

Random walks on contingency tables is a special case of a class of problems studied by Gröbner basis methods in [21]. They give graph structures for higher dimensional tables, logistic regression problems, and much else. In each case, they use the maximum-degree heuristic to get a uniform distribution. We hope to apply our methods to these problems to get faster algorithms.

3.3. A Random Family. We generate a family of graphs, all with 50 vertices, as follows. First we generate a symmetric matrix $R \in \mathbf{R}^{50\times 50}$, whose entries R_{ij}, for $i \leq j$, are independent and uniformly distributed on the interval $[0, 1]$. For each threshold value $c \in [0, 1]$ we construct a graph by placing an edge between vertices i and j for $i \neq j$ if $R_{ij} \leq c$. We always add every self-loop to the graph.

By increasing c from 0 to 1, we obtain a monotone family of graphs; i.e., the graph associated with a larger value of c contains all the edges of the graph associated with a smaller value of c. We only consider values of c above the smallest value that gives a connected graph. For each graph, we compute the SLEMs of the maximum-degree chain, the Metropolis–Hastings chain, and the fastest mixing chain (using SDP). Figure 2 shows the SLEMs of the three Markov chains for the graphs obtained for $c = 0.1,\ 0.2, \ldots, 0.9$ (all of which were connected). For these nine examples, the Metropolis chain is faster than the maximum-degree chain, but the optimal chain is always substantially faster than the Metropolis–Hastings chain.

Figure 3 shows the eigenvalue distribution of the three Markov chains for the particular graph with $c = 0.2$. Each of the distributions has a single eigenvalue at 1.

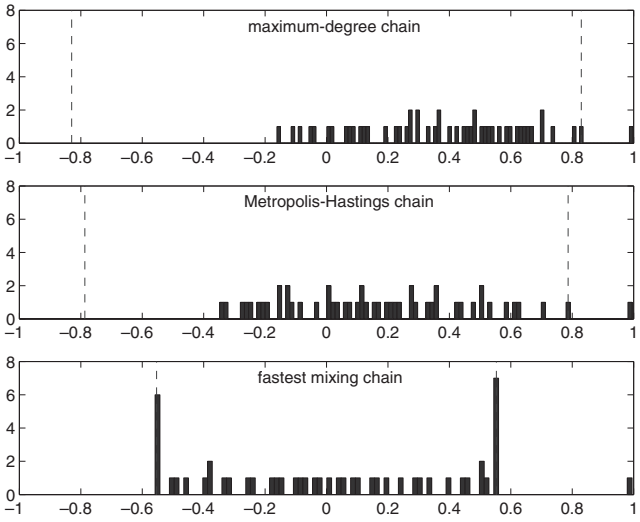

Fig. 3 *Eigenvalue distributions of the three transition probability matrices on the graph with $c = 0.2$. The dashed lines indicate the values $\pm\mu$ for each chain.*

The mixing of the maximum-degree chain is determined by its second eigenvalue, although it has some negative eigenvalues. For the Metropolis–Hastings chain, the second eigenvalue is smaller, and the smallest eigenvalue is more negative, but still not enough to affect the SLEM. For the fastest mixing chain, the eigenvalues (other than 1) have an approximately symmetric distribution, with many at or near the two critical values $\pm\mu^\star$.

4. The Dual Problem and Optimality Conditions.

4.1. The Dual Problem. Just as in linear programming, (Lagrange) duality plays an important role in convex optimization and semidefinite programming. The dual of the FMMC problem can be expressed as

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T z \\ \text{subject to} & Y\mathbf{1} = 0, \quad Y = Y^T, \quad \|Y\|_* \leq 1, \\ & (z_i + z_j)/2 \leq Y_{ij}, \quad (i,j) \in \mathcal{E}, \end{array} \tag{8}$$

with variables $z \in \mathbf{R}^n$ and $Y \in \mathbf{R}^{n\times n}$. Here $\|Y\|_* = \sum_{i=1}^n |\lambda_i(Y)|$, the sum of the singular values of Y. The sum of the singular values of a symmetric matrix is a norm; indeed, it is the dual norm of the spectral norm, so we denote it by $\|\cdot\|_*$. The dual FMMC problem is convex, since the objective, which is maximized, is linear, hence concave, and the constraints are all convex.

This dual problem can be derived in several ways (and expressed in several equivalent forms). It can be derived directly via the standard SDP dual of the SDP formulation (6). The dual problem satisfies the following:

- *Weak duality.* If Y, z are feasible for the dual problem (8), then we have $\mathbf{1}^T z \leq \mu^\star$. In other words, dual feasible points yield lower bounds on the optimal SLEM.
- *Strong duality.* There exist $Y^\star$, $z^\star$ that are optimal for the dual problem and satisfy $\mathbf{1}^T z^\star = \mu^\star$. This means that optimal values of the primal and dual

problems are the same and that the dual problem yields a sharp lower bound on the optimal SLEM.

Both of these conclusions follow from general results for convex optimization problems (see, e.g., [43, 7, 12]). We can conclude strong duality using (a refined form of) Slater's condition (see, e.g., [7, section 3.3] and [12, section 5.2]), since the constraints are all linear equalities and inequalities.

4.1.1. Derivation of Weak Duality. In this section we give a self-contained derivation of weak duality. We will show that if P is primal feasible and Y, z are dual feasible, then

$$\mathbf{1}^T z \le \mu(P). \tag{9}$$

We prove this by bounding $\mathbf{Tr}\, Y(P - (1/n)\mathbf{1}\mathbf{1}^T)$ from above and below. By the definition of dual norm, we have

$$\begin{aligned}\mathbf{Tr}\, Y\left(P - (1/n)\mathbf{1}\mathbf{1}^T\right) &\le \|Y\|_*\|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2 \\ &\le \|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2 \\ &= \mu(P).\end{aligned}$$

The second inequality uses the fact that Y is dual feasible (hence $\|Y\|_* \le 1$), and the equality uses the definition of the SLEM. On the other hand, we have

$$\begin{aligned}\mathbf{Tr}\, Y\left(P - (1/n)\mathbf{1}\mathbf{1}^T\right) &= \mathbf{Tr}\, YP = \sum_{i,j} Y_{ij}P_{ij} \\ &\ge \sum_{i,j}(1/2)(z_i + z_j)P_{ij} \\ &= (1/2)(z^T P\mathbf{1} + \mathbf{1}^T Pz) \\ &= \mathbf{1}^T z.\end{aligned}$$

The first equality uses the fact that $Y\mathbf{1} = 0$, and the inequality comes from primal and dual feasibility: $P_{ij} = 0$ for $(i,j) \notin \mathcal{E}$ and $(1/2)(z_i + z_j) \le Y_{ij}$ for $(i,j) \in \mathcal{E}$. Combining the upper and lower bounds gives the desired result (9).

As an example, consider the FMMC problem associated with the graph shown in Figure 1(b). The dual variables

$$z = \frac{1}{44}\begin{bmatrix} 25 \\ -15 \\ 9 \\ 9 \end{bmatrix}, \qquad Y = \frac{1}{44}\begin{bmatrix} 25 & 5 & -15 & -15 \\ 5 & 1 & -3 & -3 \\ -15 & -3 & 9 & 9 \\ -15 & -3 & 9 & 9 \end{bmatrix} \tag{10}$$

are easily verified to satisfy $Y\mathbf{1} = 0$, $Y = Y^T$, and $(z_i + z_j)/2 \le Y_{ij}$ for $(i,j) \in \mathcal{E}$. Moreover, Y can be expressed as

$$Y = vv^T, \qquad v = \frac{1}{\sqrt{44}}\begin{bmatrix} 5 \\ 1 \\ -3 \\ -3 \end{bmatrix},$$

so its eigenvalues are $0, 0, 0$, and $\|v\|_2 = 1$. Therefore, we have $\|Y\|_* = 1$. Thus, z and Y given in (10) are feasible dual variables.

Therefore, weak duality tells us that the corresponding dual objective, $\mathbf{1}^T z = 7/11$, is a lower bound on $\mu^\star$. Since the matrix P given in Table 1 yields $\mu(P) = 7/11$, we can conclude that it is optimal.

4.2. Optimality Conditions. From strong duality (or general convex analysis) we can develop necessary and sufficient conditions for P to be optimal for the FMMC problem. The primal variable $P^\star$ is optimal if and only if there exist dual variables $z^\star$ and $Y^\star$ that satisfy the following set of (Karush–Kuhn–Tucker) conditions:

- *Primal feasibility.*

$$P^\star \geq 0, \quad P^\star \mathbf{1} = \mathbf{1}, \quad P^\star = P^{\star T},$$
$$P^\star_{ij} = 0, \quad (i,j) \notin \mathcal{E}.$$

- *Dual feasibility.*

$$Y^\star \mathbf{1} = 0, \quad Y^\star = Y^{\star T}, \quad \|Y^\star\|_* \leq 1,$$
$$(z^\star_i + z^\star_j)/2 \leq Y^\star_{ij}, \quad (i,j) \in \mathcal{E}.$$

- *Complementary slackness.*

$$\left((z^\star_i + z^\star_j)/2 - Y^\star_{ij}\right) P^\star_{ij} = 0,$$
$$Y^\star = Y^\star_+ - Y^\star_-, \quad Y^\star_+ = Y^{\star T}_+ \succeq 0, \quad Y^\star_- = Y^{\star T}_- \succeq 0,$$
$$P^\star Y^\star_+ = \mu(P^\star) Y^\star_+, \quad P^\star Y^\star_- = -\mu(P^\star) Y^\star_-, \quad \mathbf{Tr}\, Y^\star_+ + \mathbf{Tr}\, Y^\star_- = 1.$$

The matrices $Y^\star_+$ and $Y^\star_-$ can be viewed as the positive semidefinite and negative semidefinite parts of $Y^\star$, respectively. In section 5 we will see a nice interpretation of the complementary slackness conditions in terms of the subdifferential of μ at $P^\star$. These optimality conditions can be derived with similar arguments as given in [40].

As an example, consider again the FMMC problem for the graph in Figure 1(b). It is easy to verify that the matrix P given in Table 1 and z and Y given in (10) satisfy the above necessary and sufficient conditions.

The results of this section have been used to prove that the fastest mixing Markov chain on a path with n vertices results from putting loops at the two end vertices and assigning transition probability $1/2$ on all edges and the two loops; see [11].

5. A Subgradient Method. Standard primal-dual interior-point algorithms for solving SDPs work well for problems with up to a thousand or so edges. There are many popular SDP solvers available, such as SDPSOL [56], SDPT3 [53], and SeDuMi [51]. A list of current SDP solvers can be found at the SDP website maintained by Helmberg [46]. The particular structure of the SDPs encountered in FMMC problems can be exploited for some gain in efficiency, but problems with 10,000 or more edges are probably beyond the capabilities of interior-point SDP solvers.

In this section we give a simple subgradient method that can solve the FMMC problem on very large-scale graphs, with 100,000 or more edges. The disadvantage, compared to a primal-dual interior-point method, is that the algorithm is relatively slow (in terms of number of iterations) and has no simple stopping criterion that can guarantee a certain level of suboptimality.

5.1. Subdifferential of the SLEM. A *subgradient* of μ at P is a symmetric matrix G that satisfies the inequality

$$\mu(\tilde{P}) \geq \mu(P) + \mathbf{Tr}\, G(\tilde{P} - P) = \mu(P) + \sum_{i,j} G_{ij}(\tilde{P}_{ij} - P_{ij}) \tag{11}$$

for any symmetric stochastic matrix $\tilde{P}$. Subgradients play a key role in convex analysis and are used in several algorithms for convex optimization.

We can compute a subgradient of μ at P as follows. Suppose $\mu(P) = \lambda_2(P)$ and v is a unit eigenvector associated with $\lambda_2(P)$. Then the matrix $G = vv^T$ is a subgradient of $\mu(P)$. To see this, we first note that $v^T\mathbf{1} = 0$. By the variational characterization of the second eigenvalue of P and $\tilde{P}$, we have

$$\begin{aligned}\mu(P) = \lambda_2(P) &= v^T P v,\\ \mu(\tilde{P}) \geq \lambda_2(\tilde{P}) &\geq v^T \tilde{P} v.\end{aligned}$$

Subtracting the two sides of the above equality from that of the inequality, we have the desired inequality

$$\mu(\tilde{P}) \geq \mu(P) + v^T(\tilde{P} - P)v = \mu(P) + \sum_{i,j} v_i v_j (\tilde{P}_{ij} - P_{ij}).$$

Similarly, if $\mu(P) = -\lambda_n(P)$ and v is a unit eigenvector associated with $\lambda_n(P)$, then the matrix $-vv^T$ is a subgradient of $\mu(P)$.

The subgradient method we describe requires only the computation of one subgradient, as described above. But in fact we can characterize the *subdifferential* of μ at P, defined as the set of all subgradients of μ at P, and denoted $\partial\mu(P)$. It can be shown that

$$\begin{aligned}\partial\mu(P) &= \mathbf{Co}\left(\{vv^T \mid Pv = \mu(P)v,\ \|v\|_2 = 1\} \cup \{-vv^T \mid Pv = -\mu(P)v,\ \|v\|_2 = 1\}\right)\\ &= \{Y \mid Y = Y_+ - Y_-,\ Y_+ = Y_+^T \succeq 0,\ Y_- = Y_-^T \succeq 0,\\ &\qquad PY_+ = \mu(P)Y_+,\ PY_- = -\mu(P)Y_-,\ \ \mathbf{Tr}\, Y_+ + \mathbf{Tr}\, Y_- = 1\},\end{aligned}$$

where $\mathbf{Co}(\cdot)$ denotes the convex hull. For the derivation of this result and detailed nonsmooth analysis of spectral functions, see [40, 32, 33]. More general background on nonsmooth analysis and optimization can be found in, e.g., [14, 26, 9].

5.2. Projected Subgradient Method. Let p denote the vector of transition probabilities on the non-self-loop edges (edges that connect two different vertices). For convenience, we label these edges by integers $l = 1, 2, \ldots, m$. Because the staying probabilities at the vertices can be eliminated using the equality $P\mathbf{1} = \mathbf{1}$, we can express the transition probability matrix as an affine function of p,

$$P(p) = I + \sum_{l=1}^{m} p_l E^{(l)}.$$

Suppose that edge l connects two vertices i and j ($i \neq j$), which we denote by $l \sim (i, j)$; then $E^{(l)}_{ij} = E^{(l)}_{ji} = +1$, $E^{(l)}_{ii} = E^{(l)}_{jj} = -1$, and all other entries of $E^{(l)}$ are zero. The diagonal of $P(p)$, i.e., the vector of staying probabilities at the vertices, can be written as $\mathbf{1} - Bp$, where the n by m matrix B is the vertex-edge incidence matrix defined as

$$B_{il} = \begin{cases} 1 & \text{if edge } l \text{ incident to vertex } i,\\ 0 & \text{otherwise.}\end{cases}$$

Notice that each column of B has exactly two nonzero entries indicating the two end vertices of the edge. The entry $(Bp)_i$ is the sum of transition probabilities on the incident edges at vertex i (excluding the self-loop). For p to be feasible, it must satisfy

$$p \geq 0, \qquad Bp \leq \mathbf{1}.$$

Now we can write the FMMC problem in terms of the optimization variable p:

$$\begin{array}{ll} \text{minimize} & \mu(P(p)) \\ \text{subject to} & p \geq 0, \quad Bp \leq \mathbf{1}. \end{array} \tag{12}$$

In the subgradient method, we need to compute a subgradient of the objective function $\mu(P(p))$ for a given feasible p. If $\mu(P(p)) = \lambda_2(P(p))$ and u is a unit eigenvector associated with $\lambda_2(P(p))$, then a subgradient $g(p)$ is given by

$$g(p) = \left(u^T E^{(1)} u, \ldots, u^T E^{(m)} u\right),$$

with components

$$g_l(p) = u^T E^{(l)} u = -(u_i - u_j)^2, \quad l \sim (i,j), \quad l = 1, \ldots, m.$$

Similarly, if $\mu(P(p)) = -\lambda_n(P(p))$ and v is a unit eigenvector associated with the eigenvalue $\lambda_n(P(p))$, then

$$g(p) = \left(-v^T E^{(1)} v, \ldots, -v^T E^{(m)} v\right)$$

is a subgradient, with components

$$g_l(p) = -v^T E^{(l)} v = (v_i - v_j)^2, \quad l \sim (i,j), \quad l = 1, \ldots, m.$$

For large sparse symmetric matrices, we can compute a few extreme eigenvalues and their corresponding eigenvectors very efficiently using Lanczos methods; see, e.g., Parlett [41] and Saad [45].

Now we give a simple subgradient method with approximate projection at each step k:

given a feasible p (e.g., the max-degree chain or Metropolis–Hastings chain)
$k := 1$
repeat

1. *Subgradient step.* Compute a subgradient $g^{(k)}$ and let
$$p := p - \alpha_k g^{(k)} / \|g^{(k)}\|_2$$
2. *Sequential projection step.*
   (a) $p_l := \max\{p_l,\ 0\}, \quad l = 1, \ldots, m$
   (b) *for* node $i = 1, \ldots, n$, $\mathcal{I}(i) = \{l \mid \text{edge } l \text{ incident to node } i\}$
   while $\sum_{l \in \mathcal{I}(i)} p_l > 1$
   $$\mathcal{I}(i) := \{l \mid l \in \mathcal{I}(i),\ p_l > 0\} \tag{13}$$
   $$\delta = \min\left\{ \min_{l \in \mathcal{I}(i)} p_l,\ \left(\textstyle\sum_{l \in \mathcal{I}(i)} p_l - 1\right) / |\mathcal{I}(i)| \right\} \tag{14}$$
   $$p_l := p_l - \delta, \quad l \in \mathcal{I}(i) \tag{15}$$
   end while
   end for
   (c) $k := k + 1$

In this algorithm, step 1 moves p in the direction of the subgradient with stepsize α_k, which satisfies the diminishing stepsize rule:

$$\alpha_k \geq 0, \quad \alpha_k \to 0, \quad \sum_k \alpha_k = \infty. \tag{16}$$

Step 2 approximately projects p onto the feasible set $\{p \mid p \geq 0,\ Bp \leq \mathbf{1}\}$. While the exact projection (minimum distance) can be computed by solving a quadratic program, it is computationally very expensive for very large graphs. Here we use a sequential projection method: Step 2(a) projects p onto the nonnegative orthant; in step 2(b), we project p onto one half-space at a time, and each projection is very easy to compute.

During each execution of the inner loop (the while loop), (13) updates $\mathcal{I}(i)$ as the set of incident edges to node i with strictly positive transition probabilities, and $|\mathcal{I}(i)|$ is its cardinality. If $p_{\mathrm{sum}} = \sum_{l\in\mathcal{I}(i)} p_l > 1$, we would like to project p onto the half-space $\sum_{l\in\mathcal{I}(i)} p_l \leq 1$, but doing so may cause some components of p to be negative. Instead, we project p onto the half-space

$$\sum_{l\in\mathcal{I}(i)} p_l \leq p_{\mathrm{sum}} - \delta|\mathcal{I}(i)|, \tag{17}$$

where δ is chosen to avoid negative components of the projection; see (14). The projection step (15) is very simple. The right-hand side of (17) is at least 1, and it is easy to verify that the while loop terminates in a finite number of steps, bounded by the degree of the node. Moreover, every half-space of the form (17) contains the feasible set $\{p \mid p \geq 0,\ Bp \leq \mathbf{1}\}$. This implies that the distance from any feasible point is reduced by each projection.

Once the sum probability constraint is satisfied at a node, it will never be destroyed by later projections because the edge probabilities can only decrease in the sequential projection procedure.

Let p denote the probability vector after step 1, and p^+ denote the vector after step 2. It is clear that p^+ produced by the sequential projection method is always feasible, and the distance to any optimal solution is reduced, i.e.,

$$\|p^\star - p^+\|_2 \leq \|p^\star - p\|_2$$

for any optimal solution $p^\star$. This is a critical property that allows us to prove the convergence of this algorithm using standard arguments for projected subgradient methods with the diminishing stepsize rule (16). Such proofs can be found in [47, section 2.2] and [7, section 6.3].

Closely related to the subgradient method is the spectral bundle method for solving large-scale SDPs; see, e.g., [26, 25, 37]. Other methods for solving large eigenvalue optimization problems can be found in, e.g., [39, 34].

5.3. Example. To demonstrate the projected subgradient algorithm, we apply it to a large-scale graph with 10,000 vertices and 100,000 edges, generated using the same method described in section 3.3. We use a simple Lanczos method to compute λ_1, λ_2, and λ_n, which exploits sparsity of P and the rank-1 property of $\mathbf{1}\mathbf{1}^T$ for efficiency. We use step length $\alpha_k = 1/\sqrt{k}$ and start the transition matrix at the Metropolis–Hastings chain.

The progress of the algorithm is plotted in Figure 4, which shows the magnitude of the two extreme eigenvalues λ_2 and λ_n, versus iteration number. After 500 iterations,

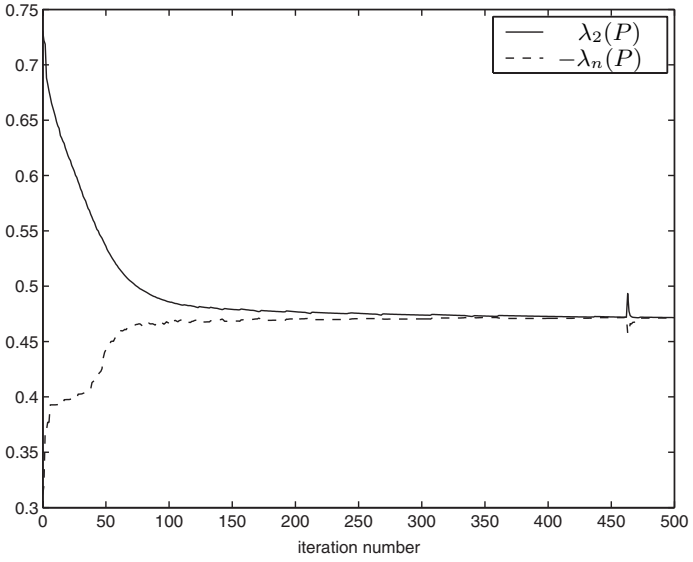

Fig. 4 *Progress of the subgradient method for the FMMC problem with a graph with* 10,000 *vertices and* 100,000 *edges.*

the algorithm gives $\mu = 0.472$, which is a significant reduction compared with the Metropolis–Hastings chain, which has SLEM $\mu(P^{\mathrm{mh}}) = 0.730$ (at $k = 0$). The spike at $k = 463$ happened when λ_n had a larger magnitude than λ_2, and the subgradient computation used the eigenvector associated with λ_n. At all other iterations, λ_2 had a larger magnitude than λ_n. Figure 5 shows the distribution of the 101 largest and 100 smallest eigenvalues of the two Markov chains at $k = 0$ and $k = 500$.

6. Fastest Mixing Reversible Markov Chain. We can generalize the FMMC problem to reversible Markov chains or, equivalently, to random walks on a graph with weighted edges. In this case, each edge $(i, j) \in \mathcal{E}$ is associated with two transition probabilities P_{ij} and P_{ji}. The transition matrix is not required to be symmetric, but it must satisfy the detailed balance condition

$$\pi_i P_{ij} = \pi_j P_{ji}, \qquad i,\ j = 1, \ldots, n, \tag{18}$$

where $\pi = (\pi_1, \ldots, \pi_n)$ is the equilibrium distribution of the Markov chain.

The two heuristics—the Metropolis and max-degree construction—can easily be adapted to the nonuniform case. For Metropolis, take the proposal distribution to be nearest neighbor random walk on the underlying graph with uniform weights. A move from vertex i to j is accepted with probability $\min(1, (\pi_j d_i)/(\pi_i d_j))$. Otherwise the walk stays at i. For max-degree, suppose we are given positive weights w_i at each vertex. Choose $w^\star \geq \max_i \sum_{(i,j)\in\mathcal{E}} w_j$. Then set $P_{ij} = w_j/w^\star$ for $i \neq j$ with $(i, j) \in \mathcal{E}$ and choose P_{ii} to satisfy $P\mathbf{1} = \mathbf{1}$. Both constructions give a reversible Markov chain with stationary distribution proportional to w_i.

In the fastest mixing reversible Markov chain problem, we are given a fixed equilibrium distribution π, and the goal is to find a reversible transition probability matrix with smallest SLEM. Let $\Pi = \mathrm{diag}(\pi)$, so the detailed balance condition (18) can be written as $\Pi P = P^T \Pi$, which means that the matrix $\Pi^{1/2} P \Pi^{-1/2}$ is symmetric (and,

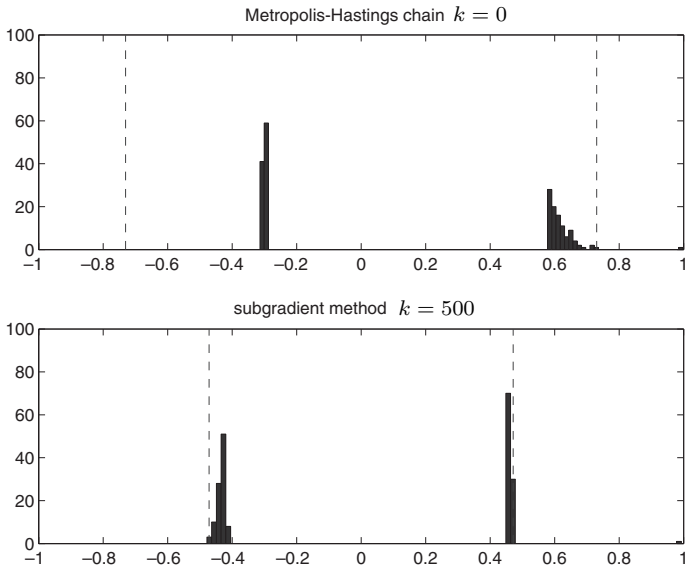

Fig. 5 *Distribution of the* 101 *largest and* 100 *smallest eigenvalues. The dashed lines indicate the values* $\pm\mu$ *for each Markov chain.*

of course, has the same eigenvalues as P). The eigenvector of $\Pi^{1/2}P\Pi^{-1/2}$ associated with the maximum eigenvalue 1 is $q = (\sqrt{\pi_1}, \ldots, \sqrt{\pi_n})$. The SLEM $\mu(P)$ equals the second largest (in magnitude) eigenvalue of $\Pi^{1/2}P\Pi^{-1/2}$ or, equivalently, its spectral norm restricted to the subspace $q^{\perp}$. This can be written as

$$\mu(P) = \|(I - qq^T)\Pi^{1/2}P\Pi^{-1/2}(I - qq^T)\|_2 = \|\Pi^{1/2}P\Pi^{-1/2} - qq^T\|_2.$$

Thus the fastest mixing reversible Markov chain problem can be formulated as

$$\begin{array}{ll} \text{minimize} & \mu(P) = \|\Pi^{1/2}P\Pi^{-1/2} - qq^T\|_2 \\ \text{subject to} & P \geq 0, \quad P\mathbf{1} = \mathbf{1}, \quad \Pi P = P^T\Pi, \\ & P_{ij} = 0, \quad (i,j) \notin \mathcal{E}, \end{array} \tag{19}$$

which is a convex optimization problem. We can derive an SDP formulation of this problem by introducing a scalar variable s to bound the norm of $\Pi^{1/2}P\Pi^{-1/2} - qq^T$, as in (6).

7. Extensions.

7.1. Exploiting Symmetry. In many cases, the graphs of interest have large symmetry groups, and this can be exploited to substantially increase the efficiency of solution methods or even, in some cases, to solve the FMMC problem analytically. This is explored in far more detail in [10, 42]; here we describe a very simple case to illustrate the basic idea.

We first observe that if a graph is symmetric, then we can assume without loss of generality that the optimal transition matrix $P^\star$ is also symmetric. To see this, let $P^\star$ be any optimal transition matrix. If g is a permutation of $\mathcal{V}$ that preserves the graph, then $gP^\star$ (which is $P^\star$ with its rows and columns permuted by g) is also feasible. Let $\overline{P}$ denote the average over the orbit of P under the symmetry group.

This matrix is feasible (since the feasible set is convex, and each $gP^\star$ is feasible), and moreover, using convexity of μ, we have $\mu(\overline{P}) \leq \mu(P^\star)$. It follows that $\mu(\overline{P}) = \mu^\star$, i.e., $\overline{P}$ is optimal. It is also, by construction, invariant under the symmetry group.

Now we consider a specific example, a complete bipartite graph (illustrated in Figure 1(c)), where the two parts have m and n nodes, respectively. Without loss of generality, we can assume $n \geq m \geq 2$. Here the symmetry group is $S_n \times S_m$: we can arbitrarily permute each of the parts. By symmetry, we can assume that every (non-self-loop) edge has the same transition probability p, with $0 < p < 1/n$. The transition probability matrix is thus

$$P(p) = \begin{bmatrix} (1-np)I_m & p\mathbf{1}_m\mathbf{1}_n^T \\ p\mathbf{1}_n\mathbf{1}_m^T & (1-mp)I_n \end{bmatrix};$$

i.e., we have only one scalar variable, p, the edge transition probability. This matrix has at most four different eigenvalues, which are

$$1, \qquad 1-mp, \qquad 1-np, \qquad 1-(m+n)p,$$

so the FMMC problem reduces to minimizing the maximum absolute value of the last three expressions, over the choice of p. It is easy to verify that the optimal transition probability is

$$p^\star = \min\left\{\frac{1}{n}, \frac{2}{n+2m}\right\},$$

and the associated smallest SLEM is

$$\mu(P^\star) = \max\left\{\frac{n-m}{n}, \frac{n}{n+2m}\right\}.$$

In addition to reducing the number of variables, exploiting symmetry can often lead to a change of basis that makes the matrix block diagonal, which reduces the size of matrices in the numerical SDP solution. For more details, see [42].

7.2. Some Bounds. The spectral gap $(1-\mu)$ for the max-degree, Metropolis, and fastest mixing Markov chains can be quite different. Jerrum [28] has suggested the following example showing that the optimal chain can improve over the max-degree chain by unbounded amounts: Let K_n denote the complete graph on n vertices. Let K_n–K_n denote two disjoint copies of K_n joined by a single edge. Here $d_{\max} = n$. Analysis presented in [10] shows that $1-\mu(P^{\mathrm{md}}) = 2n^{-2} + O(n^{-3})$, while the fastest mixing chain has $1-\mu^\star \geq (1/3)n^{-1} + O(n^{-2})$.

On the other hand, the fastest mixing Markov chain can only improve the spectral gap to the second eigenvalue, i.e., $(1-\lambda_2)$, by a factor of $d_{\max}$, the maximum degree. Let P be any symmetric Markov chain on $\mathcal{G}$ and P^{md} be the maximum degree chain defined in section 1.2.1. Then we have the following inequality:

$$1-\lambda_2(P) \leq d_{\max}\left(1-\lambda_2(P^{\mathrm{md}})\right). \tag{20}$$

Since this holds for any Markov chain on $\mathcal{G}$, it holds for any optimal one in particular.

To show this, we use the variational characterization of eigenvalues (see, e.g., [27, p. 176] and [13, p. 205]),

$$1-\lambda_2(P) = \inf\left\{\frac{\mathcal{E}_{P,\pi}(f,f)}{\mathrm{Var}_\pi(f)} \;\middle|\; \mathrm{Var}_\pi(f) \neq 0\right\},$$

where the Dirichlet form $\mathcal{E}_{P,\pi}(f,f)$ and variance $\mathrm{Var}_\pi(f)$ are defined as

$$\mathcal{E}_{P,\pi}(f,f) = \frac{1}{2}\sum_{i,j}(f_i - f_j)^2\pi_i P_{ij}, \qquad \mathrm{Var}_\pi(f) = \frac{1}{2}\sum_{i,j}(f_i - f_j)^2\pi_i\pi_j.$$

Here the equilibrium distribution $\pi = \mathbf{1}/n$. We use $P_{ij} \le 1$, and compare the Dirichlet forms of P and P^{md}:

$$\begin{aligned}\mathcal{E}_{P,\pi}(f,f) &= \frac{1}{2}\sum_{i,j}(f_i - f_j)^2 P_{ij}\frac{1}{n}\\ &\le d_{\max}\frac{1}{2}\sum_{i,j}(f_i - f_j)^2\frac{1}{d_{\max}n}\\ &= d_{\max}\mathcal{E}_{P^{\mathrm{md}},\pi}(f,f).\end{aligned}$$

This implies the inequality (20).

Similar arguments can be used to show that

$$1 - \lambda_2(P^{\mathrm{md}}) \le 1 - \lambda_2(P^{\mathrm{mh}}) \le \frac{d_{\max}}{d_{\mathrm{mh}}}\left(1 - \lambda_2(P^{\mathrm{md}})\right),$$

where

$$d_{\mathrm{mh}} = \min_{(i,j)\in\mathcal{E}} \max\{d_i, d_j\}.$$

Thus the two heuristics, max-degree and Metropolis–Hastings, are roughly comparable.

Of course, the negative eigenvalues can change the story for the spectral gap, in particular if λ_n is very close to -1, since the spectral gap $1-\mu = \min\{1-\lambda_2,\ 1+\lambda_n\}$. One thing to notice is that the two extreme eigenvalues, λ_2 and λ_n, are both monotone decreasing in the off-diagonal entries of P. In particular, since $P_{ij}^{\mathrm{mh}} \ge P_{ij}^{\mathrm{md}}$ for all $i \ne j$, we always have

$$\lambda_2(P^{\mathrm{mh}}) \le \lambda_2(P^{\mathrm{md}}), \qquad \lambda_n(P^{\mathrm{mh}}) \le \lambda_n(P^{\mathrm{md}}).$$

This can be shown by similar arguments as above.

We mention here one more related result. Consider the following modification of the max-degree chain. Let all the edge transition probabilities be equal to $1/(d_{\max}+1)$ and denote the corresponding transition probability matrix by $P^{\mathrm{md}+}$. Then it can be shown, using arguments similar to the ones above, that

$$1 - \mu(P) \le (d_{\max} + 1)\left(1 - \mu(P^{\mathrm{md}+})\right)$$

for any symmetric transition probability matrix P defined on the graph. Thus, the spectral gap of the optimal chain is no more than a factor $d_{\max} + 1$ larger than the spectral gap of the modified max-degree chain.

7.3. Optimizing log-Sobolev Constants. We have used the spectral gap as a measure of rapid mixing. The log-Sobolev constant (see, e.g., [23, 50, 18]) is another important alternative measure of mixing rate. If (P,π) is a reversible Markov chain on $\{1,2,\ldots,n\}$, the log-Sobolev constant α is defined by

$$\alpha = \inf\left\{\frac{\mathcal{E}_{P,\pi}(f,f)}{\mathcal{L}_\pi(f)} \;\middle|\; \mathcal{L}_\pi(f) \ne 0\right\}$$

with the denominator defined as

$$\mathcal{L}_\pi(f) = \sum_i |f_i|^2 \log\left(\frac{|f_i|^2}{\|f\|^2}\right)\pi_i.$$

Note that $\mathcal{L}_\pi(f)$ is nonnegative, with the definition $\|f\|^2 = \sum_i |f_i|^2\pi_i$.

The constant α may be compared with the spectral gap associated with the eigenvalue λ_2,

$$1-\lambda_2 = \inf\left\{\frac{\mathcal{E}_{P,\pi}(f,f)}{\mathrm{Var}_\pi(f)} \;\middle|\; \mathrm{Var}_\pi(f) \neq 0\right\}.$$

As one motivation for considering α, we consider Markov chains running in continuous time. Let $H(t) = e^{-t(I-P)}$, with $H_{ij}(t)$ being the probability of moving from i to j after time $t > 0$. Starting at i, the total variation distance between the distribution at time t and the stationary distribution π is bounded as

$$4\|H_i(t) - \pi\|_{\mathrm{tv}}^2 \leq \frac{1}{\pi_*}e^{-2(1-\lambda_2)t}$$

with $\pi_* = \min \pi_i$. In [18, (1.8)] it is shown that

$$2\|H_i(t) - \pi\|_{\mathrm{tv}}^2 \leq \log\left(\frac{1}{\pi_*}\right)e^{-2\alpha t}.$$

For large state spaces the $1/\pi_*$ factors can be huge and the improvements to understanding convergence can be sizeable.

We can formulate the problem of finding the reversible Markov chain, with fixed equilibrium distribution π, on a graph $\mathcal{G}$, that has maximum log-Sobolev constant as a convex optimization problem. We have already shown that $R(\mathcal{G},\pi)$, the set of π-reversible Markov chains compatible with $\mathcal{G}$, is a convex set; we only need to show that $\alpha : (\mathcal{G},\pi) \to [0,\infty)$ is a concave function on $R(\mathcal{G},\pi)$. To do this, we note that since $\mathcal{E}_{P,\pi}$ and $\mathcal{L}_\pi$ are homogeneous of degree 2, we can express α as

$$\alpha = \inf\{\mathcal{E}_{P,\pi}(f,f) \mid \mathcal{L}_\pi(f) = 1\}.$$

For fixed f, the Dirichlet form $\mathcal{E}_{P,\pi}(f,f)$ is an affine function of P. Thus α is a pointwise minimum of affine functions and so is concave.

We hope to compare the chains obtained by optimizing α with those optimizing μ in future work.

Acknowledgments. The authors are grateful to Pablo Parrilo for helpful discussions of the FMMC problem in general and for exploiting graph symmetry in particular. We are grateful to Joe and Jill Oliveira, owners of the BYTES cafe, where this research started.

REFERENCES

[1] D. Aldous, *Random walk on finite groups and rapidly mixing Markov chains*, in Séminaire de Probabilités XVII, Lecture Notes in Math. 986, Springer-Verlag, New York, 1983, pp. 243–297.

[2] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[3] F. Alizadeh, J.-P. A. Haeberly, M. V. Nayakkankuppam, M. L. Overton, and S. Schmieta, *SDPpack: A Package for Semidefinite-Quadratic-Linear Programming*, 1997.
[4] E. Behrends, *Introduction to Markov Chains, with Special Emphasis on Rapid Mixing*, Adv. Lectures Math., Vieweg, Weisbaden, Germany, 2000.
[5] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS/SIAM Ser. Optim. 2, SIAM, Philadelphia, 2001.
[6] S. J. Benson, Y. Ye, and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (2000), pp. 443–461.
[7] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Nashua, NH, 1999.
[8] L. Billera and P. Diaconis, *A geometric interpretation of the Metropolis-Hastings algorithm*, Statist. Sci., 16 (2001), pp. 335–339.
[9] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization. Theory and Examples*, CMS Books Math./Ouvrages Math. SMC, Springer-Verlag, New York, 2000.
[10] S. Boyd, P. Diaconis, P. A. Parrilo, and L. Xiao, *Symmetry analysis of reversible Markov chains*, Internet Math., to appear; available online from http://www.stanford.edu/~boyd/symmetry.html.
[11] S. Boyd, P. Diaconis, J. Sun, and L. Xiao, *Fastest mixing Markov chain on a path*, Amer. Math. Monthly, to appear; available online from http://www.stanford.edu/~boyd/fmmc_path.html.
[12] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004; available online from http://www.stanford.edu/~boyd/cvxbook.html.
[13] P. Brémaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues*, Texts Appl. Math., Springer-Verlag, Berlin, Heidelberg, 1999.
[14] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, 1990.
[15] J. Cullum, W. E. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Stud., 3 (1975), pp. 35–55.
[16] P. Diaconis, *Group Representations in Probability and Statistics*, Institute of Mathematical Statistics, Hayward, CA, 1988.
[17] P. Diaconis and A. Gangolli, *Rectangular arrays with fixed margins*, in Discrete Probability and Algorithms, D. Aldous et al., eds., Springer-Verlag, New York, 1995, pp. 15–42.
[18] P. Diaconis and L. Saloff-Coste, *Logarithmic Sobolev inequalities for finite Markov chains*, Ann. Appl. Probab., 6 (1996), pp. 695–750.
[19] P. Diaconis and L. Saloff-Coste, *What do we know about the Metropolis algorithms?*, J. Comput. System Sci., 57 (1998), pp. 20–36.
[20] P. Diaconis and D. Stroock, *Geometric bounds for eigenvalues of Markov chains*, Ann. Appl. Probab., 1 (1991), pp. 36–61.
[21] P. Diaconis and B. Sturmfels, *Algebraic algorithms for sampling from conditional distributions*, Ann. Statist., 26 (1998), pp. 363–397.
[22] R. Diekmann, S. Muthukrishnan, and M. V. Nayakkankuppam, *Engineering diffusive load balancing algorithms using experiments*, in Solving Irregularly Structured Problems in Parallel, Lecture Notes in Comput. Sci. 1253, Springer-Verlag, Berlin, 1997, pp. 111–122.
[23] L. Gross, *Logarithmic Sobolev inequalities and contractivity properties of semigroups*, in Dirichlet Forms (Varenna, 1992), Lecture Notes in Math. 1563, Springer-Verlag, Berlin, 1993, pp. 54–88.
[24] W. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.
[25] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. Optim., 10 (2000), pp. 673–696.
[26] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.
[27] R. A. Horn and C. A. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[28] M. Jerrum, *Private communication*, 2003.
[29] M. Jerrum and A. Sinclair, *Approximating the permanent*, SIAM J. Comput., 18 (1989), pp. 1149–1178.
[30] N. Kahale, *A semidefinite bound for mixing rates of Markov chains*, in Proceedings of the 5th Integer Programming and Combinatorial Optimization Conference, Lecture Notes in Comput. Sci. 1084, Springer-Verlag, Berlin, 1996, pp. 190–203.
[31] R. Kannan, *Markov chains and polynomial time algorithms*, in Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science, 1994, pp. 656–671.

[32] A. S. Lewis, *Convex analysis on the Hermitian matrices*, SIAM J. Optim., 6 (1996), pp. 164–177.
[33] A. S. Lewis, *Nonsmooth analysis of eigenvalues*, Math. Programming, 84 (1999), pp. 1–24.
[34] A. S. Lewis and M. L. Overton, *Eigenvalue optimization*, Acta Numer., 5 (1996), pp. 149–190.
[35] J. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer Ser. Statist., Springer-Verlag, New York, 2001.
[36] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, *Equations of state calculations by fast computing machines*, J. Chem. Phys., 21 (1953), pp. 1087–1092.
[37] S. A. Miller and R. S. Smith, *A bundle method for efficiently solving large structured linear matrix inequalities*, in Proceedings of the American Control Conference, Chicago, IL, 2000, pp. 1405–1409.
[38] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math., SIAM, Philadelphia, 1994.
[39] M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.
[40] M. L. Overton and R. S. Womersley, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Math. Programming, 62 (1993), pp. 321–357.
[41] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[42] P. A. Parrilo, L. Xiao, S. Boyd, and P. Diaconis, *Fastest Mixing Markov Chain on Graphs with Symmetries*, manuscript in preparation, 2004.
[43] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[44] J. S. Rosenthal, *Convergence rates for Markov chains*, SIAM Rev., 37 (1995), pp. 387–405.
[45] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
[46] *Semidefinite programming page*, http://www-user.tu-chemnitz.de/~helmberg/semidef.html. Website maintained by C. Helmberg.
[47] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*, Springer Ser. Comput. Math., Springer-Verlag, Berlin, 1985.
[48] A. Sinclair, *Improved bounds for mixing rates of Markov chains and multicommodity flow*, Combin. Probab. Comput., 1 (1992), pp. 351–370.
[49] A. Sinclair, *Algorithms for Random Generation and Counting: A Markov Chain Approach*, Birkhäuser Boston, Boston, 1993.
[50] D. Stroock, *Logarithmic Sobolev inequalities for Gibbs states*, in Dirichlet Forms (Varenna, 1992), Lecture Notes in Math. 1563, Springer-Verlag, Berlin, 1993, pp. 194–228.
[51] J. F. Sturm, *Using SeDuMi* 1.02, *a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11–12 (1999), pp. 625–653; special issue on Interior Point Methods (CD supplement with software).
[52] M. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.
[53] K. C. Toh, M. J. Todd, and R. H. Tutuncu, *SDPT3: A Matlab software package for semidefinite programming, version* 2.1, Optim. Methods Softw., 11 (1999), pp. 545–581.
[54] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.
[55] H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., *Handbook of Semidefinite Programming, Theory, Algorithms, and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.
[56] S.-P. Wu and S. Boyd, sdpsol: *A Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure. User's Guide, Version Beta*, Stanford University, Stanford, CA, 1996.
[57] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1997.