

---

# On Doubly Stochastic Graph Optimization

---

## Abstract

In this paper we introduce an approximate optimization framework for solving graphs problems involving doubly stochastic matrices. This is achieved by using a low dimensional formulation of the matrices and the approximate solution is achieved by a simple subgradient method. We also describe one problem that can be solved using our method.

## 1 Introduction

Graph optimization is a class of problems that assigns edge weights or transition probabilities to a given graph which minimize a given criterion usually subject to some connectivity and other constraints. An example is the fastest mixing Markov chain problem [1], where the object is to assign transition probabilities that minimize the mixing rate of a Markov random walk on a given graph. The mixing rate problem has been shown to arise in a class of gossip algorithm problems [10] where the object is to find an averaging algorithm or equivalently a transition matrix such that the averaging time over the graph is minimized. Other related problems that involve optimization over spectral functions of doubly stochastic matrices include

1. Minimizing Effective Resistance on a Graph [2], where the idea is to choose a random walk on the graph that minimizes the average commute time between all nodes.
2. Finding the best doubly stochastic approximation to a given affinity matrix [5]. This arises in the context of spectral clustering in machine learning.

Here we consider the fastest mixing Markov chain problem and present an efficient approximate solution based on using a smaller subset of the space of large doubly stochastic transition matrices. This involves a computationally expensive pre-processing step but needs to be executed only once for a given problem. In the next section we describe the notation and the underlying results used to reduce the dimensionality of the problem.

## 2 Problem Formulation

### 2.1 Fastest Mixing Markov Chain

Consider a symmetric Markov chain on a graph  $G$  with a transition matrix  $P \in R^{n \times n}$ . The stationary distribution in this case is the uniform distribution  $\pi = (1/n)e^T$  and the mixing rate measures how fast an initial distribution converges to the uniform. This rate of convergence is measured by the second largest eigenvalue modulus (SLEM) of  $P$ ,  $\mu(P) = \max(\lambda_2(P), -\lambda_n(P))$ , the smaller it is the faster the Markov chain converges. Here  $\lambda_2(P)$  and  $\lambda_n(P)$  are the second largest and the smallest eigenvalues of  $P$ . The problem of finding the fastest mixing Markov chain was described in [1]. It can be written as

Input- A  $n \times n$  doubly stochastic matrix  $A$

1. for  $l = 1 : n^2 + n - 2$
2. Using Bipartite Matching find a permutation  $\pi_l$  of vertices  $1, \dots, n$  such that each  $A_{i,\pi_i}$  is positive
3.  $\theta_l = \min_i(A_{i,\pi_i})$
4.  $A = A - \theta_l P_{\pi_l}$ , where  $P_{\pi_l}$  is a permutation matrix corresponding to  $\pi_l$ .
5. Exit if all entries of  $A$  are zero

Output  $(\theta_i, P_{\pi_i})$  such that  $A = \sum_{i=1}^M \theta_i P_{\pi_i}$

Table 1: BN Decomposition Algorithm for a Doubly Stochastic Matrix

$$\begin{aligned}
 & \min \mu(P) & (1) \\
 & \text{s.t.} \\
 & Pe = e, P^T = P, P \geq 0 \\
 & P_{ij} = 0 \text{ if } \{i, j\} \notin E
 \end{aligned}$$

The problem can be expressed as an SDP and solved using standard techniques and also for very large graphs with more 100,000 or more edges a subgradient method is presented in [1].

## 2.2 BN Decomposition

Let  $G = (V, E)$  be an undirected graph with  $n$  vertices and  $m$  edges. There is a Markov chain associated with the graph such that the weight of each edge is the transition probability  $p_{ij}$  from the  $i$  to the  $j$ th node. These transition probabilities are described by a  $n \times n$  matrix  $P$  such that

$$P \geq 0, Pe = e, P = P^T, P_{ij} = 0 \text{ if } \{i, j\} \notin E \quad (2)$$

where  $e$  is the vector of all 1's and the inequality indicates element wise positivity for  $P_{ij}$ . The entries are zero only if there is no corresponding edge in the given graph.

We can write the symmetric stochastic matrix  $P$  as a convex combination of Permutation matrices using the following result due to Birkhoff [6]

*Theorem- Any  $n \times n$  matrix  $P$  is doubly stochastic if and only if there are  $M$   $n \times n$  permutation matrices  $P_1, \dots, P_M$  and positive scalars  $\theta_1, \dots, \theta_M$  such that*

$$P = \sum_{i=1}^M \theta_i P_i \text{ and } \sum_{i=1}^M \theta_i = 1 \quad (3)$$

Some bounds on  $M$  exist in the literature [6], but as we shall see it becomes irrelevant for our purposes. Now given a matrix  $P$  an algorithm (table 1) based on a proof given by Dulmage and Halperin is described in [7]. It involves bipartite graph matching and to each such matching corresponds a permutation matrix. These permutation matrices define a basis for a certain subset of the space of  $n \times n$  doubly stochastic matrices.

### 2.2.1 Identifying Basis Subset

If we have a reasonable choice for a Markov chain that mixes fast, for e.g. the Metropolis Hastings chain. We hope to use its BN decomposition to select a permutation basis. Having then identified such a permutation basis we can hope to solve for the fastest mixing chain by optimizing over the

$\theta$ 's, keeping the basis matrices fixed. However the number of basis matrices or equivalently the problem size could be very large. Can we then identify a smaller subset of basis matrices and search over those?

Consider the initial transition matrix  $P^m$ , an input to the decomposition algorithm to obtain a permutation basis for the space of DS matrices to search on. The BN procedure returns a  $\theta = (\theta_1, \dots, \theta_M)$  vector and  $M$  permutation matrices  $(P_1, \dots, P_M)$ . Intuitively, since higher values of  $\theta_i$  contribute more to probability weight on a particular edge, it makes sense to ignore altogether very small  $\theta_i$ 's and hence the corresponding  $P_i$ 's. This is reasonable if we make the assumption that since our initial choice  $P^m$  is a good one any improvement that we hope to find over this one is structurally similar to  $P^m$ . Then we use the following procedure values to filter out insignificant  $P_i$ 's

Table 2: Select Basis

Choose $k \ll M$ Compute $r_i = \ P^m - \theta_i P_i\ _F$ for all $i$ Return the $P_i$ 's corresponding to the $k$ smallest $r_i$ 's.
---

Later in the experiments section we will show the results for different values of  $k$  for a fixed  $M$ . Once we have the  $P_i$ 's we have in essence fixed a subset of Birkhoff polytope for our optimization procedure to search over. The parameter space for the search is then defined by the  $\theta = (\theta_1, \dots, \theta_k) \in R^k$  such that this new  $\theta$  lies in the probability simplex and  $P(\theta) = \sum_{i=1}^k \theta_i P_i$ . Clearly the SLEM is also a nonlinear function of  $\theta$  and will be written as  $\mu(\theta)$ . Note that since  $P(\theta)$  is symmetric,  $P(\theta) = (P(\theta) + P(\theta)^T)/2 = \sum_{i=1}^k \theta_i (P_i + P_i^T)/2$ . Hence our basis matrices are  $(P_i + P_i^T)/2$  for each  $i$  to maintain the symmetry constraint.

In the ensuing sections we show experimental evidence and demonstrate that it is possible to truncate the space considerably and still obtain reasonable results.

### 2.3 Basis Subset Optimization for Fastest Mixing Chain

The most commonly used heuristic for fast mixing is the Metropolis-Hastings random walk. To obtain a Markov chain with the uniform stationary distribution the following transition matrix is constructed [1]

$$P_{ij}^m = \begin{cases} \min(1/d_i, 1/d_j) & \text{if } i \neq j \text{ and } \{i, j\} \in E \\ \sum_k \max(0, 1/d_i - 1/d_k) & \text{if } i=j \text{ and } \{i, k\} \in E \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Using the procedure to identify the basis subset in the previous section. We BN decompose the  $P^m$  matrix to obtain the subset space to search over. Next we use a subgradient method to solve the fastest mixing problem on this smaller subset of DS matrices constrained by the fixed chosen permutation basis.

#### 2.3.1 Subgradient Method

In this parameter space the optimization problem becomes

$$\begin{aligned} \min \mu(\theta) \\ \text{s.t.} \\ \sum_{i=1}^k \theta_i = 1, \theta_i \geq 0 \end{aligned} \quad (5)$$

The SLEM in general is a non-differentiable function of the entries of the matrix and therefore as a function of the  $\theta$  parameter. We use subgradients to solve the optimization problem. It can be

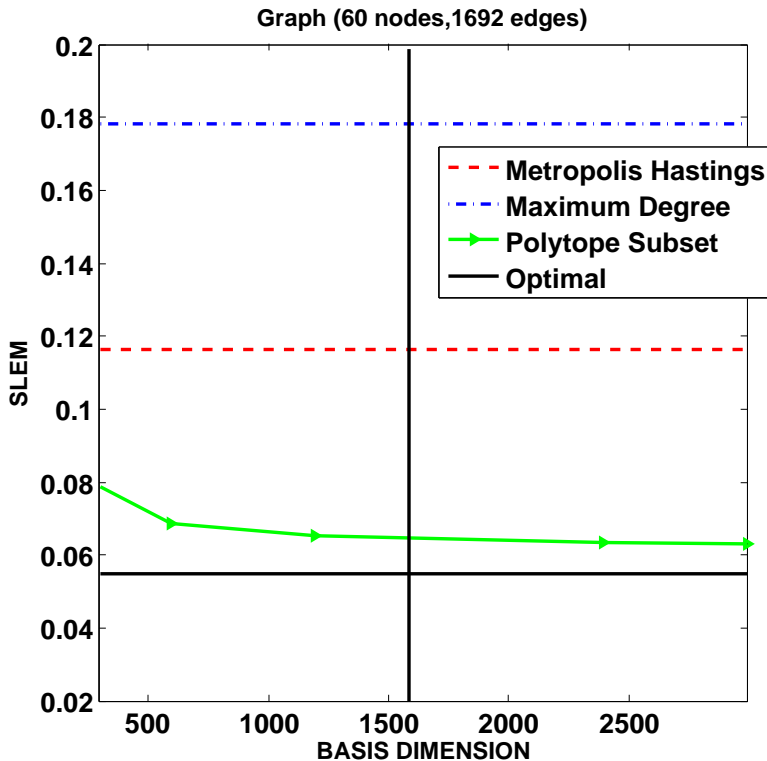


Figure 1: SLEM for a fixed graph with varying basis dimension size for our method. The horizontal axis is the number of basis elements used, i.e. the number of variables being optimized. The vertical line is the number of edges, i.e. the number of variables being optimized in a direct optimization approach. The parallel lines include the corresponding SLEM values for Metropolis Hastings and the Optimal. The SLEM values comes close to the optimal as we increase the basis dimension.

shown that the subgradient  $g$  corresponding to the equation below is dependent on the eigenvector corresponding to the second largest eigenvalue in magnitude

$$\mu(\tilde{\theta}) \geq \mu(\theta) + v(\theta)^T (P(\tilde{\theta}) - P(\theta))v(\theta) = \mu(\theta) + \delta\theta^T g \quad (6)$$

and is given by  $g = (v(\theta)^T P_1 v(\theta), \dots, v(\theta)^T P_k v(\theta))$ . Where  $(P_1, \dots, P_k)$  are the permutation basis and  $v(\theta)$  is the eigenvector corresponding to the second largest eigenvalue in magnitude. Thus at each iteration an eigenvector computation is required.

The projected subgradient method then proceeds as usual on this considerably smaller  $k$ -dimensional space and involves a projection step onto the probability simplex for which we use an efficient algorithm described in [8].

The overall algorithm can be given as

1. Compute the Metropolis Hastings matrix  $P^m$  for a given Graph  $G$ .
2. Compute the BN decomposition of  $P^m$  to obtain a permutation basis such that  $P^m = \sum_{i=1}^M \theta_i^m P_i$  and obtain the truncated basis by ignoring permutation matrices corresponding to the  $(M - k)$   $\theta_i^m$ 's returned by the truncation procedure.
3. Solve the optimization problem (5) using the subgradient method above for parameters  $\theta_1, \dots, \theta_k$ .

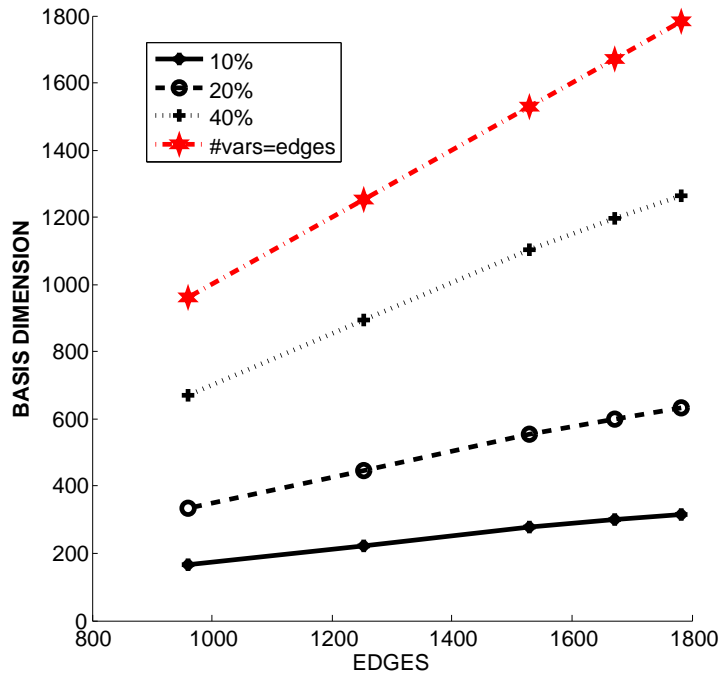
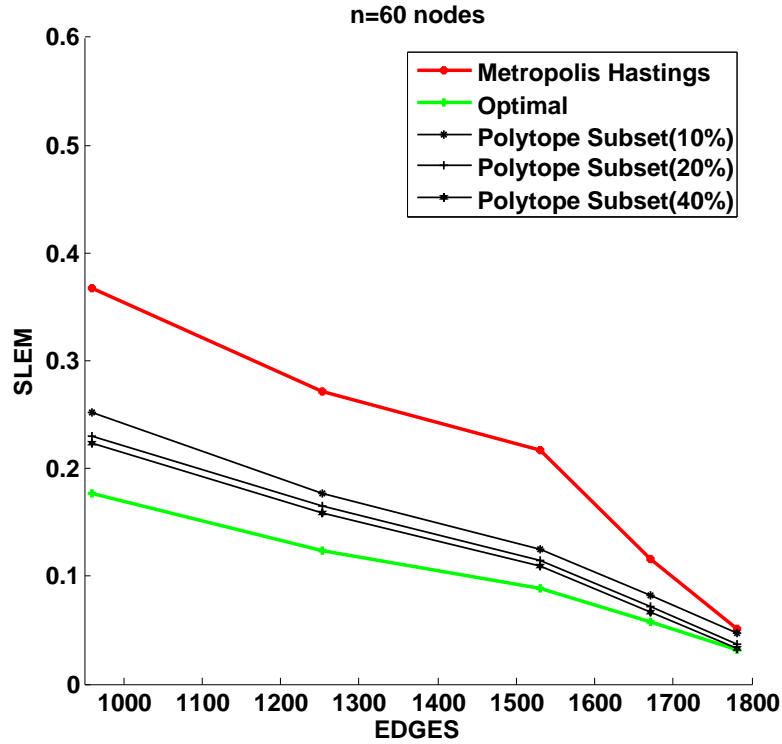


Figure 2: a) SLEM for graphs with varying no. of edges. The % indicates the proportion of total basis matrices used. Even with 10% of the total variables, performance is much better than the Metropolis Hastings chain. b) The plot shows the basis dimension (problem size) (Y) for each of the graphs (edges (X)) in the previous plot for each of the % levels. The number of variables are considerably less than edges in the graph at the 10% level. The top line shows the no. of variables as the number of edges in the direct optimization approach.

## 2.4 Simulation

We first present results on a small graph with 60 nodes generated uniformly as described in fastest mixing paper [1]. Figure 1 shows a graph with  $n = 60$  nodes and the corresponding slem's as we increase the basis dimension or equivalently the number of permutation basis matrices used. The middle line indicates the number of edges in the graph. It can be seen that even with a limited basis dimension we do considerably better than the Metropolis-Hastings chain and stay relatively close to the optimal. Thus the basis dimension can also be used as a knob to control the accuracy vs efficiency tradeoff.

Figure 2(a) shows the SLEM values on graphs with  $n = 60$  nodes and varying number of edges. We can see that even with 10% of permutation basis we stay relatively close to the optimal value and significantly better than the Metropolis Hastings chain. In figure 2(b) we can see in the top line the gain we get if were to solve the optimization problem with the number of variables equal to the number of edges as opposed to a fraction of the total basis dimension.

## 3 Conclusion

In this paper we presented an efficient subgradient algorithm based on the Birkhoff-von Neumann decomposition to get the approximate fastest mixing rate Markov chain on a graph. For future work we hope that there are problems involving doubly stochastic matrices that we can solve using our framework. We also intend to present distributed versions of the algorithm as the subgradient method is very amenable to parallel computation. On the theoretical front the issue of the optimality loss incurred in restricting the search space by using a fixed permutation basis or equivalently the quality of our approximation needs to be understood.

## 4 References

- [1] Boyd S., Diaconis P. & Xiao L., Fastest Mixing Markov Chain on a Graph. , *SIAM Review* Vol.46,No.4,pp. 667-689,2004.
- [2] Ghosh A., Boyd S. & Saberi S., Minimizing Effective Resistance of a Graph., *SIAM Review* , problems and techniques section, 50(1):37-66, February 2008..
- [3] Ghosh A. & Boyd S., Upper Bounds on Algebraic Connectivity using Convex Optimization., *Linear Algebra and its Applications*, 418:693-707, October 2006..
- [4] Xiao L. & Boyd S., Fast linear iterations for distributed averaging., *Systems and Control Letters*, 53:65-78, 2004.
- [5] Zass R. & Shashua A., Doubly Stochastic Normalization for Spectral Clustering., *Proceeding NIPS*, 2006..
- [6] Minc H. & Weiss Y. Nonnegative Matrices *Wiley Intersciences Series in Discrete Mathematics and Optimization*, 1988.
- [7] Marshall W. A. & Olkin I. Inequalities:Theory of Majorization and Applications, *Academic Press* 1979.
- [8] Duchi J., Shwartz S.S., Singer Y. & Chandra T., Efficient projections onto the L1-ball for learning in high dimensions . *Proceedings ICML* 2008.
- [9] Horn R.A. & Johnson C.R.,Matrix Analysis. *Cambridge University Press*,1985.
- [10] Boyd S.,Ghosh A.,Prabhakar B. & Shah D. , Randomized Gossip Algorithms. *IEEE Transactions on Information Theory*, Vol. 52,NO. 6, June 2006.