

Assigning Stationary Distributions to Sparse Stochastic Matrices

Nicolas Gillis*

Paul Van Dooren†

Abstract

The target stationary distribution problem (TSDP) is the following: given an irreducible stochastic matrix G and a target stationary distribution $\hat{\boldsymbol{\mu}}$, construct a minimum norm perturbation, Δ , such that $\hat{G} = G + \Delta$ is also stochastic and has the prescribed target stationary distribution, $\hat{\boldsymbol{\mu}}$. In this paper, we revisit the TSDP under a constraint on the support of Δ , that is, on the set of non-zero entries of Δ . This is particularly meaningful in practice since one cannot typically modify all entries of G . We first show how to construct a feasible solution \hat{G} that has essentially the same support as the matrix G . Then we show how to compute globally optimal and sparse solutions using the component-wise ℓ_1 norm and linear optimization. We propose an efficient implementation that relies on a column-generation approach which allows us to solve sparse problems of size up to $10^5 \times 10^5$ in a few minutes. We illustrate the proposed algorithms with several numerical experiments.

Keywords: stochastic matrix, stationary distribution, support, sparsity, linear optimization, Metropolis-Hastings algorithm

AMS subject classifications: 60J10, 93C73, 65F15, 90C05

1 Introduction

In this paper, we consider discrete-time Markov chains with n nodes. We denote the transition matrix of such a chain as the nonnegative matrix $G \in \mathbb{R}_+^{n \times n}$ where the entry of G at position (i, j) , denoted $G_{i,j}$, is the probability to go to node j when leaving node i . The matrix G is stochastic, that is, its entries are nonnegative and the sum of the entries in each row is equal to one. The matrix G is irreducible if it is not reducible, that is, if there does not exist two sets of indices, \mathcal{I} and \mathcal{J} , such that $\mathcal{I} \cup \mathcal{J} = \{1, 2, \dots, n\}$, $\mathcal{I} \cap \mathcal{J} = \emptyset$ and $G_{i,j} = 0$ for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$. Irreducibility of G implies that there exists a unique positive vector $\boldsymbol{\mu} > 0$ such that $\boldsymbol{\mu}^\top G = \boldsymbol{\mu}^\top$ and $\boldsymbol{\mu}^\top \mathbf{1}_n = 1$; this is a consequence of the Perron-Frobenius theorem; see [6] for more details. This vector $\boldsymbol{\mu}$ is called the stationary distribution of G , and accounts for the “importance” of each node, as the i th entry of $\boldsymbol{\mu}$ is equal to the probability to be in node i after having spent enough time in the chain.

Stochastic matrices G are used to model hyperlink networks used in particular for the PageRank computation [18], social networks [30], or queuing networks [8]. In such Markov chains, the stationary distributions contain important information on the nodes in the network, such as their centrality or other types of rankings [2]. The target stationary distribution then captures some desired properties of

*Department of Mathematics and Operational Research, Université de Mons, Rue de Houdain 9, 7000 Mons, Belgium. Email: nicolas.gillis@umons.ac.be. NG acknowledges the support by the European Union (ERC consolidator, eLinoR, no 101085607).

†UCLouvain, Email: vandooren.p@gmail.com.

the system. For example, consider a road network where each node is a road, and $G_{i,j}$ the probability that a car on road i turns into road j , while $G_{i,i}$ can be chosen to model the different travel times along the different roads [16]. The i th entry of the stationary distribution $\boldsymbol{\mu}$ represents the long-term average probability with which a typical car will be found to drive on road i . In other words, the vector $\boldsymbol{\mu}$ represents the (relative) road congestion; see [5] for more details and other applications.

In this context, the target stationary distribution problem (TSDP) was introduced recently in [5], and is defined as follows. We are given

- G , an $n \times n$ irreducible stochastic matrix with positive stationary distribution $\boldsymbol{\mu} > 0$, that is,

$$G \geq 0, \quad G\mathbf{1}_n = \mathbf{1}_n, \quad \boldsymbol{\mu}^\top G = \boldsymbol{\mu}^\top, \quad \boldsymbol{\mu}^\top \mathbf{1}_n = 1, \quad (1)$$

where $\mathbf{1}_n$ is the n -dimensional vector of all 1's.

- $\hat{\boldsymbol{\mu}} > 0$, a positive target distribution such that $\hat{\boldsymbol{\mu}}^\top \mathbf{1}_n = 1$.

The TSDP requires to find a minimum norm correction, Δ , such that $\hat{G} := G + \Delta$ is still stochastic and has the target $\hat{\boldsymbol{\mu}}$ as its stationary distribution. The set of admissible candidate matrices, \mathcal{D} , is thus described by three conditions:

$$\mathcal{D} := \left\{ \Delta \in \mathbb{R}^{n \times n} \mid \hat{\boldsymbol{\mu}}^\top (G + \Delta) = \hat{\boldsymbol{\mu}}^\top, \quad \Delta \mathbf{1}_n = \mathbf{0}_n, \quad G + \Delta \geq 0 \right\}, \quad (2)$$

where $\mathbf{0}_n$ is the n -dimensional vector of all 0's; see [19] for more details on this feasible set. In practice, one is interested in reaching the target distribution with minimum effort, that is, in finding minimum norm solutions Δ yielding a perturbed model $\hat{G} = G + \Delta$ with the prescribed distribution $\hat{\boldsymbol{\mu}}$. The set \mathcal{D} is convex and the TSDP requires to solve the following convex optimization problem

$$\min_{\Delta \in \mathcal{D}} \|\Delta\|, \quad (\text{TSDP})$$

for a given norm $\|\cdot\|$. For example, in the road network application, one is interested to modify the network at minimum cost (e.g., add a few road segments, and/or remove/add one-way streets) in order to reduce the congestion.

This line of research is quite different from the body of literature on the sensitivity of the stationary distribution of a stochastic matrix G with respect to a perturbation Δ that preserves the stochasticity of $G + \Delta$. This is a field of active research and various sensitivity bounds have been proposed in the literature [1] [14] [15], [22] [23] [26] [27] [29]. Such bounds are of interest in a wide range of application areas, such as mathematical physics, climate modeling, Bayesian statistics, and bio-informatics. But the aim of the present paper is to assign a target distribution, which is an inverse problem rather than a sensitivity problem.

To the best of our knowledge, there are only a few works that consider the TSDP or variants. An early work that considered a similar problem is [7]. They prove that the famous Metropolis-Hastings (M-H) algorithm [25, 20] (see (3) below for the formula) finds the optimal solution of the TSDP where the norm minimized is the weighted component-wise ℓ_1 norm¹, $\sum_{i=1}^n \hat{\mu}_i \|G(i, \cdot) - \hat{G}(i, \cdot)\|_1$, and with the additional constraint that \hat{G} is $\hat{\boldsymbol{\mu}}$ -reversible, that is,

$$\hat{\mu}_i \hat{G}(i, j) = \hat{\mu}_j \hat{G}(j, i), \quad \text{for all } i, j.$$

¹In [7], authors do not take into account the diagonal entries in the objective. However, by the stochastic constraint and the fact that the M-H solution satisfies $\hat{G}(i, j) \leq G(i, j)$ for all $i \neq j$, this is equivalent, since $\sum_{j \neq i} |G(i, j) - \hat{G}(i, j)| = \frac{1}{2} \|G(i, \cdot) - \hat{G}(i, \cdot)\|_1$ for all i .

The M-H solution can be computed in $O(\text{nnz}(G))$ operations, as follows:

$$\hat{G}(i, j) = \min \left(G(i, j), \frac{\hat{\mu}_j}{\hat{\mu}_i} G(j, i) \right) \quad \text{for all } i \neq j, \quad (3)$$

while $\hat{G}(i, i) = 1 - \sum_{j \neq i} \hat{G}(i, j)$ for all i . Note that $\hat{G}(i, j) = 0$ whenever $G(i, j) = 0$ or $G(j, i) = 0$. This implies that the support of G is contained in $\text{supp}(I) \cup (\text{supp}(G) \cap \text{supp}(G^\top))$. If the support of G is not symmetric, that is, $\text{supp}(G) \neq \text{supp}(G^\top)$, then the M-H algorithm might fail to generate an irreducible chain \hat{G} ; see, e.g., the discussion in [17]. For example, take the irreducible cycle

$$G = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

with $\boldsymbol{\mu} = [1/3, 1/3, 1/3]$, and let $\hat{\boldsymbol{\mu}} = [1/2, 1/4, 1/4]$. The solution to the M-H algorithm is the identity matrix, that is, $\hat{G} = I_n$, which is not irreducible, while the optimal solution is given by

$$\Delta^* = \begin{bmatrix} 0.5 & -0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \text{ so that } G + \Delta^* = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Note that this solution will coincide with our closed-form solution proposed in section 2.

Solving the TSDP for large-scale problems directly using commercial solvers might be intractable for large n , with $O(n^2)$ variables and constraints; see the discussion in section 3. This motivated the introduction of a heuristic approach to find an approximate minimizer using a set of rank-1 corrections [5]. An algorithm was given in [5] for constructing a minimum norm rank-1 solution when such a solution exists. However, for sparse matrices, there might not exist feasible rank-1 perturbations. Moreover, rank-one perturbations are typically dense, which is not desirable: in practice, it is typically not possible to modify most of the links in a network, but rather one would like to modify only a few. For example, in a road network [16], it is not possible to add links between distant roads, while one would like to reduce the congestion by modifying as few existing links as possible.

In this paper, we therefore consider the TSDP with a support constraint while trying to promote sparse solutions. The support of a matrix $\Delta \in \mathbb{R}^{n \times n}$ is defined as the set containing the index pairs where Δ is non-zero, that is,

$$\text{supp}(\Delta) = \{(i, j) \mid \Delta_{i,j} \neq 0\}.$$

The support constraint requires that the support of the perturbation, Δ , is contained in a given set $\Omega \subseteq \{(i, j) \mid 1 \leq i, j \leq n\}$. Finally, we focus in this paper on the following TSDP problem:

$$\min_{\Delta \in \mathcal{D}} \|\Delta\|_1 \quad \text{such that} \quad \text{supp}(\Delta) \subset \Omega, \quad (4)$$

where

- $\|\Delta\|_1 = \sum_{i,j} |\Delta_{i,j}|$ is the component-wise ℓ_1 norm which promotes sparsity as it is the convex envelope (that is, the tightest convex lower bound) of the ℓ_0 norm on the ℓ_∞ ball; see, e.g., [3] and the references therein. In fact, we will prove in Theorem 3.1 that if (4) is feasible, there is an optimal solution with less than $\min(|\Omega|, \text{nnz}(G) + 2n)$ non-zero entries, where $|\Omega|$ is the cardinality of Ω and $\text{nnz}(G)$ is the number of non-zero entries in G .

- Ω is a given set of index pairs that correspond the links that can be modified in the Markov chain.

For example, if one can only modify existing links, we have $\Omega = \text{supp}(G)$, which is referred to as a non-structural perturbation [12].

The above problem is related to the feasible set $\{G \mid G \geq 0, G\mathbf{1}_n = \mathbf{1}_n, \text{supp}(G) \subset \Omega\}$ which was studied in [12], [19], and where authors tried to describe all possible stationary vectors of irreducible matrices in this set, for a given Ω . Note that researchers have tackled other optimization problems over Markov chains; for example, how to choose the non-zero entries of G such that the modulus of the second largest eigenvalue is minimized. This is the so-called problem of finding the fastest mixing Markov chain on the graph induced by G [9].

Contribution and outline of the paper In section 2, we first show that the TSDP with support constraint (4) always admits a feasible solution for $\Omega = \text{supp}(G + I)$, and provide a feasible solution that is an optimal diagonal scaling of $I - G$. We then discuss and analyze this feasible solution, in particular its distance to optimality, when it is rank-one and optimal, and how the ordering of $\hat{\boldsymbol{\mu}}$ affects the solution. In section 3, we propose an efficient linear optimization formulation of (4), with $2n$ equalities and $O(|\Omega|)$ variables, that allows us to solve (4) in $O(n^3)$ operations, in the worst case. When Ω is sparse, the run time is empirically observed to be comparable to the time of constructing a sparse $2n \times O(\text{nnz}(G))$ matrix with 2 nonzeros per column, where $\text{nnz}(G)$ is the number of non-zero entries in G , allowing us to solve sparse problems of size $10^4 \times 10^4$ within seconds, and of size $10^5 \times 10^5$ within minutes. When Ω is dense and G is sparse, we propose a column-generation approach that allows us to solve problems of similar size in a comparable computational time, by initializing the solution with the feasible case $\Omega = \text{supp}(G + I)$. We report numerical experiments in section 4.

Notation We use uppercase for matrices, boldface lowercase for vectors and normal lowercase for scalars. The set $\mathbb{R}_+ = \{x \mid x \geq 0\}$ is the nonnegative orthant. We use $\mathbf{0}_n$ and $\mathbf{1}_n$ to denote the n -vectors of all 0's and 1's, respectively, and \mathbf{e}_i to denote the i th basis vector in \mathbb{R}^n . The $n \times n$ matrix of all 1's is denoted by $\mathbf{1}_{n,n}$. We drop the index n when the dimension is clear from the context. The entry of matrix M at position (i, j) is denoted by $M_{i,j}$, and the i th entry of the vector \mathbf{v} by v_i . The i th row (resp. j th column) of M is denoted by $M_{i,:}$ (resp. by $M_{:,j}$). By $\text{supp}(M)$, we mean the set of index pairs for which the matrix M is nonzero, and by $\text{nnz}(M) = |\text{supp}(M)|$ the number of nonzeros in M where $|\cdot|$ denotes the cardinality of a set. By $\text{Diag}(\alpha_1, \dots, \alpha_n)$, we mean the $n \times n$ diagonal matrix with the parameters α_i on its main diagonal, and $(\cdot)^\top$ denotes the transpose. The inequalities $>$ and \geq applied to vectors and matrices are elementwise inequalities. We say that an n -by- n matrix M is stochastic if $M \geq 0$ and $M\mathbf{1}_n = \mathbf{1}_n$.

2 TSDP with $\Omega = \text{supp}(G + I_n)$

In this section, we analyze a special case where we allow the non-zero entries of G and entries on its diagonal to be modified, that is, $\Omega = \text{supp}(G + I_n)$. We first show that the TSDP (4) is always feasible in this case (section 2.1). In fact, we provide an explicit closed-form solution. We then discuss properties of this solution: optimality (section 2.2), sparsity and rank-1 case (section 2.3), and how the ordering of $\boldsymbol{\mu}$ compared to that of $\hat{\boldsymbol{\mu}}$ affects the solution (section 2.4).

2.1 Closed-form feasible solutions for $\Omega = \text{supp}(G + I_n)$

Let us consider the solutions to the TSDP of the form

$$\Delta(\boldsymbol{\alpha}) = D(\boldsymbol{\alpha})(I_n - G),$$

where $D(\boldsymbol{\alpha}) := \text{Diag}(\alpha_1, \dots, \alpha_n)$ and $\mathbf{0}_n \leq \boldsymbol{\alpha} \leq \mathbf{1}_n$ so that $\text{supp}(\Delta) \subseteq \text{supp}(G + I_n)$. This leads to the perturbed matrices

$$G(\boldsymbol{\alpha}) := G + \Delta(\boldsymbol{\alpha}) = (I_n - D(\boldsymbol{\alpha}))G + D(\boldsymbol{\alpha}). \quad (5)$$

The i th row of $G(\boldsymbol{\alpha})$ is a convex combination of the i th row of G with weight $(1 - \alpha_i)$ and the i th row of the identity matrix with weight α_i . This means that the larger α_i , the more important is the self-loop of node i , and hence a larger increase in the corresponding entry of the stationary vector is expected. This is a standard construction in Markov chains. As we will see below (Theorem 2.2), this construction allows us to obtain any stationary distribution.

The following lemma formalizes the fact that $G(\boldsymbol{\alpha})$ is stochastic, and irreducible if $\boldsymbol{\alpha} < \mathbf{1}_n$.

Lemma 2.1. *Let G be an irreducible stochastic matrix. Then the family of matrices*

$$\mathcal{G}_{\boldsymbol{\alpha}} := \{G(\boldsymbol{\alpha}) \mid \mathbf{0}_n \leq \boldsymbol{\alpha} \leq \mathbf{1}_n\}$$

is a closed convex set of stochastic matrices, and the subset $\mathcal{G}_{\boldsymbol{\alpha} < \mathbf{1}_n} := \{G(\boldsymbol{\alpha}) \mid \mathbf{0}_n \leq \boldsymbol{\alpha} < \mathbf{1}_n\}$ is the set of the irreducible stochastic matrices within $\mathcal{G}_{\boldsymbol{\alpha}}$.

Proof. Each row of the matrix $G(\boldsymbol{\alpha})$ is a convex combination of that of G and I , and $\mathcal{G}_{\boldsymbol{\alpha}}$ is a closed convex set of stochastic matrices. When constraining the parameters to $\mathbf{0}_n \leq \boldsymbol{\alpha} < \mathbf{1}_n$, the matrices $G(\boldsymbol{\alpha})$ and G (which is actually equal to $G(\mathbf{0}_n)$) have the same off-diagonal zero pattern, and hence $G(\boldsymbol{\alpha})$ is irreducible if and only if G is irreducible (see, e.g., [10, Chap. 8]). Finally, in order to show that $\mathcal{G}_{\boldsymbol{\alpha} < \mathbf{1}_n}$ contains *all* irreducible matrices of $\mathcal{G}_{\boldsymbol{\alpha}}$, it suffices to see that if any $\alpha_i = 1$ then $\mathbf{e}_i^\top G(\boldsymbol{\alpha}) = \mathbf{e}_i^\top$, which implies that $G(\boldsymbol{\alpha})$ is reducible. \square

The solutions $\Delta(\boldsymbol{\alpha})$ lead to a set of stochastic matrices, $\mathcal{G}_{\boldsymbol{\alpha}}$. It remains to choose the free parameters $\boldsymbol{\alpha}$ to assign $\hat{\boldsymbol{\mu}}$ as the stationary distribution of an irreducible stochastic matrix $G(\boldsymbol{\alpha})$ in $\mathcal{G}_{\boldsymbol{\alpha} < \mathbf{1}_n}$, that is, choose $\boldsymbol{\alpha}$ such that $\hat{\boldsymbol{\mu}}^\top G(\boldsymbol{\alpha}) = \hat{\boldsymbol{\mu}}^\top$.

The following theorem shows that the set $\mathcal{G}_{\boldsymbol{\alpha} < \mathbf{1}_n}$ always contains feasible solutions for arbitrary G and $\hat{\boldsymbol{\mu}}$, and provides the minimum norm solution among these feasible solutions.

Theorem 2.2. *Let G be an irreducible stochastic matrix with stationary distribution $\boldsymbol{\mu} > 0$, and let $\hat{\boldsymbol{\mu}} > 0$, $\hat{\boldsymbol{\mu}}^\top \mathbf{1}_n = 1$, be a given target stationary distribution. Then the set of parameters $\boldsymbol{\alpha}$ so that the matrices $G(\boldsymbol{\alpha}) \in \mathcal{G}_{\boldsymbol{\alpha} < \mathbf{1}_n}$ have the target $\hat{\boldsymbol{\mu}}$ as stationary distribution, that is, $G(\boldsymbol{\alpha})^\top \hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}}$, is given by*

$$\boldsymbol{\alpha}(c) = \mathbf{1}_n - c(\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}), \quad 0 < c \leq 1/\|\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}\|_\infty,$$

where $\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}$ denotes the element-wise division² of the vectors $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$. The parameter vector $\boldsymbol{\alpha}^ = \boldsymbol{\alpha}(c^*)$ of minimum norm is obtained for $c^* = 1/\|\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}\|_\infty$, and it also minimizes the norm of the perturbation matrix $\Delta(\boldsymbol{\alpha}^*) = D(\boldsymbol{\alpha}^*)(I_n - G)$, for any monotone norm $\|\cdot\|$, that is, any norm satisfying the property that $|A| \geq |B| \Rightarrow \|A\| \geq \|B\|$ for any matrix A and B .*

²This is the MATLAB notation.

Proof. If the vector $\hat{\boldsymbol{\mu}}$ is the stationary distribution of $G(\boldsymbol{\alpha})$, it must satisfy $\hat{\boldsymbol{\mu}}^\top (I_n - G(\boldsymbol{\alpha})) = 0$. Since $G(\boldsymbol{\alpha}) = (I_n - D(\boldsymbol{\alpha}))G + D(\boldsymbol{\alpha})I_n$ for some $\boldsymbol{\alpha}$, $\hat{\boldsymbol{\mu}}$ must satisfy

$$\hat{\boldsymbol{\mu}}^\top (I_n - D(\boldsymbol{\alpha}))(I_n - G) = 0.$$

But since G is irreducible, the rank of $(I_n - G)$ is $n - 1$ and any left null vector must be proportional to $\boldsymbol{\mu}^\top$. Therefore, all candidate solutions for $\boldsymbol{\alpha}$ must satisfy, for some non-zero scalar c :

$$\hat{\boldsymbol{\mu}}^\top (I_n - D(\boldsymbol{\alpha})) = c\boldsymbol{\mu}^\top, \quad \mathbf{0}_n \leq \boldsymbol{\alpha} < \mathbf{1}_n,$$

which can be rewritten as a system of linear equations and inequalities :

$$\hat{\mu}_i(1 - \alpha_i) = c\mu_i, \quad 0 \leq \alpha_i < 1, \quad i = 1, \dots, n,$$

in the unknown variables c and α_i , $i = 1, \dots, n$. Using the \circ -notation for the elementwise product of vectors, this is equivalent to

$$\hat{\boldsymbol{\mu}} \circ (\mathbf{1}_n - \boldsymbol{\alpha}) = c\boldsymbol{\mu}, \quad \mathbf{0}_n < (\mathbf{1}_n - \boldsymbol{\alpha}) \leq \mathbf{1}_n.$$

Since $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$ are positive, and the vector $(\mathbf{1}_n - \boldsymbol{\alpha})$ is constrained to be positive, c must be positive as well. The upper bound $(\mathbf{1}_n - \boldsymbol{\alpha}) \leq \mathbf{1}_n$ implies that c must be smaller than or equal to $c^* := 1/\|\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}\|_\infty$. The set of solutions is then the line segment

$$\boldsymbol{\alpha}(c) = \mathbf{1}_n - c\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}, \quad 0 < c \leq c^* := 1/\|\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}\|_\infty,$$

and the parameter vector $\boldsymbol{\alpha}(c)$ of minimum norm is obtained for $c = c^*$ since all entries of $\boldsymbol{\alpha}(c)$ decrease linearly with increasing c in the interval $0 < c \leq c^*$. The perturbation matrix $\Delta(\boldsymbol{\alpha}) = D(\boldsymbol{\alpha})(I_n - G)$ is linear in $\boldsymbol{\alpha}$, and the absolute values of all entries decrease with increasing c , and hence any monotone norm also decreases with increasing c , reaching its minimum for $c = c^*$. \square

The following example illustrates the method.

Example 2.3. Consider the stochastic matrix G of a ring network, in which each node connects to two other nodes, forming a single continuous pathway through each node, with stationary distribution $\boldsymbol{\mu}$:

$$G = \begin{bmatrix} 1/2 & 1/4 & 0 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \\ 1/4 & 0 & 1/4 & 1/2 \end{bmatrix}, \quad \boldsymbol{\mu}^\top = [1/4, 1/4, 1/4, 1/4].$$

Let the target distribution be the vector $\hat{\boldsymbol{\mu}}^\top = [\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}]$. We have $\boldsymbol{\mu}^\top./\hat{\boldsymbol{\mu}}^\top = [2, 2, 1, \frac{1}{2}]$, with optimal parameter $c^* = \frac{1}{2}$, so that $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(c^*) = [0, 0, \frac{1}{2}, \frac{3}{4}]^\top$, leading to

$$\Delta(\boldsymbol{\alpha}^*) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1/8 & 1/4 & -1/8 \\ -3/16 & 0 & -3/16 & 3/8 \end{bmatrix} \text{ and } G(\boldsymbol{\alpha}^*) = \begin{bmatrix} 1/2 & 1/4 & 0 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/8 & 5/8 & 1/4 \\ 1/16 & 0 & 1/8 & 13/16 \end{bmatrix}.$$

Since G is symmetric, the M-H algorithm, see (3), will generate a feasible solution, namely

$$\Delta = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1/8 & 1/8 & 0 \\ -3/16 & 0 & -1/8 & 5/16 \end{bmatrix} \text{ and } \hat{G} = \begin{bmatrix} 1/2 & 1/4 & 0 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/8 & 3/4 & 1/8 \\ 1/16 & 0 & 1/16 & 7/8 \end{bmatrix},$$

with $\|\Delta\|_1 = 7/8 < \|\Delta(\alpha^*)\|_1 = 10/8$. Note that an optimal solution with error $6/8$, computed using linear optimization (see section 3), is

$$\Delta = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1/8 & 0 & 1/8 \\ -3/16 & 0 & -1/16 & 1/4 \end{bmatrix} \text{ and } \hat{G} = \begin{bmatrix} 1/2 & 1/4 & 0 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/8 & 1/2 & 3/8 \\ 1/16 & 0 & 3/16 & 3/4 \end{bmatrix}.$$

In the following sections, we discuss several properties of the solutions $G(\alpha)$. Before doing so, let us introduce some additional notation. For given stationary distributions μ and $\hat{\mu}$, we define the ratio $\mathbf{r} := \mu./\hat{\mu}$ whose entries belong to the interval $0 < r_* \leq r_i \leq r^*$, where

$$r_* := \min_i \mu_i/\hat{\mu}_i > 0, \quad \text{and} \quad r^* := \max_i \mu_i/\hat{\mu}_i > 0. \quad (6)$$

Note that the maximal admissible value for c is $c^* = 1/r^*$. The elements of $\alpha(c^*)$ are bounded by

$$0 \leq \alpha_i = 1 - r_i/r^* \leq 1 - r_*/r^* = \frac{r^* - r_*}{r^*},$$

and

$$|\Delta(\alpha)| = D(\alpha)|I - G| \leq (1 - r_*/r^*)|I - G| = \frac{(r^* - r_*)}{r^*}|I - G|. \quad (7)$$

Hence $\|\alpha(c^*)\|$ and $\|\Delta(\alpha^*)\|$ for mononote norms $\|\cdot\|$ are expected to decrease with a decreasing gap $r^* - r_*$, and are equal to zero if and only if $\mu = \hat{\mu}$, that is, $r^* = r_*$.

2.2 How good is $\Delta(\alpha^*)$?

In this subsection we compare the optimal solution in \mathcal{G}_α with the lower bound for the optimal solution in the larger set \mathcal{D} given in (2). The following lemma gives such a lower bound.

Lemma 2.4. *Any Δ such that $\hat{\mu}^\top(G + \Delta) = \hat{\mu}^\top$, in particular any feasible solution $\Delta \in \mathcal{D}$ of the TSDP, satisfies*

$$\|\Delta\| \geq \frac{\|\hat{\mu}^\top(I - G)\|}{\|\hat{\mu}\|} = \frac{\|(\hat{\mu} - \mu)^\top(I - G)\|}{\|\hat{\mu}\|},$$

for any induced norm³, and this also holds for the component-wise ℓ_1 norm.

Proof. The inequality $\|x^\top\| \cdot \|A\| \geq \|x^\top A\|$ for the vector-matrix product $x^\top A$ holds for any induced norm, and also holds for the component-wise ℓ_1 norm. Applying this to $\hat{\mu}^\top \Delta = \hat{\mu}^\top(I - G)$ then yields the required inequality for $\|\Delta\|$. The equality then follows trivially from $\mu^\top(I - G) = 0$. \square

³Given a vector norm $\|\cdot\|$, the matrix induced norm is given by $\|A\| = \sup_{\mathbf{z} \neq 0} \frac{\|A\mathbf{z}\|}{\|\mathbf{z}\|}$.

For the perturbation $\Delta(\boldsymbol{\alpha}) = D(\boldsymbol{\alpha})(I_n - G)$, the bound $\|\Delta(\boldsymbol{\alpha}^*)\| \leq (1 - r_*)/r^* \|I_n - G\|$ was obtained earlier; see (7). Putting this bound together with Lemma 2.4, we obtain the following interval

$$\|(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^\top (I_n - G)\| / \|\hat{\boldsymbol{\mu}}^\top\| \leq \|\Delta(\boldsymbol{\alpha}^*)\| \leq (1 - r_*)/r^* \|I_n - G\|, \quad (8)$$

for the optimal $\Delta(\boldsymbol{\alpha}^*)$ in $\mathcal{G}_{\boldsymbol{\alpha}}$. This interval goes to zero when $\mathbf{d} := \hat{\boldsymbol{\mu}} - \boldsymbol{\mu}$ goes to zero, which indicates that the optimal solution constrained to $\mathcal{G}_{\boldsymbol{\alpha}}$ is approaching a globally optimal solution. This will be illustrated on numerical examples in section 4. In the next section, we focus on the case of a rank-1 perturbation.

2.3 Support of $\Delta(\boldsymbol{\alpha}^*)$ and optimal rank-1 solutions

Using Theorem 2.2, it is straightforward to describe the support of the feasible solutions of the TSDP of the form $\Delta(\boldsymbol{\alpha}) = D(\boldsymbol{\alpha})(I_n - G)$.

Lemma 2.5. *Let G be an irreducible stochastic matrix with stationary distribution $\boldsymbol{\mu} > 0$, and let us use the same notation as in Theorem 2.2. For all $\boldsymbol{\alpha} = \boldsymbol{\alpha}(c)$ such that $c < c^*$, we have $\text{supp}(\Delta(\boldsymbol{\alpha})) = \text{supp}(G + I_n)$. For $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(c^*)$, the rows of $\Delta(\boldsymbol{\alpha})$ with index in $\{i \mid \frac{\mu_i}{\hat{\mu}_i} = r^* = \frac{1}{c^*}\}$ are all-zeros, the other rows have the same support as that of $\text{supp}(G + I_n)$ (as when $c < c^*$). In other terms, for $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(c^*)$,*

$$\text{supp}(\Delta(\boldsymbol{\alpha}^*)_{i,:}) = \begin{cases} \text{supp}((G + I_n)_{i,:}) & \text{for } \frac{\mu_i}{\hat{\mu}_i} < r^* = \frac{1}{c^*}, \\ \emptyset & \text{for } \frac{\mu_i}{\hat{\mu}_i} = r^* = \frac{1}{c^*}. \end{cases}$$

Proof. We have $\Delta(\boldsymbol{\alpha})_{i,:} = \alpha_i(\mathbf{e}_i^\top - G_{i,:})$. Since G is irreducible $G_{i,:} \neq \mathbf{e}_i^\top$, and hence the supports of the i th rows of $\Delta(\boldsymbol{\alpha})$ and $I - G$ coincide unless $\alpha_i = 0$. For $\boldsymbol{\alpha} = \boldsymbol{\alpha}(c)$, this happens only when $c = c^*$ and $\frac{\mu_i}{\hat{\mu}_i} = r^* = \frac{1}{c^*}$. \square

Lemma 2.5 implies that the solution $\Delta(\boldsymbol{\alpha}^*)$ will have the same support as $G + I_n$, except for the rows that correspond to the maximum entries of $\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}$ which will be equal to zero.

Rank-1 case Lemma 2.5 also implies that $\Delta(\boldsymbol{\alpha}^*)$ has rank-1 if the vector $\boldsymbol{\mu}./\hat{\boldsymbol{\mu}}$ has only two different values, namely r_* and r^* , and if, moreover, r^* is repeated $n - 1$ times, in which case $\Delta(\boldsymbol{\alpha}^*)$ has a single non-zero row with index j such that

$$\mu_j = r_* \hat{\mu}_j \quad \text{and} \quad \mu_i = r^* \hat{\mu}_i \quad \text{for all } i \neq j. \quad (9)$$

Since $\boldsymbol{\mu}^\top \mathbf{1}_n = \hat{\boldsymbol{\mu}}^\top \mathbf{1}_n = 1$, $r_* + \hat{\mu}_j(r_* - r^*) = 1$, we have $\hat{\mu}_j = (r^* - 1)/(r^* - r_*) < 1$. Finally, we obtain

$$r_* < 1 < r^*, \quad \alpha_j = 1 - \frac{r_*}{r^*}, \quad \alpha_i = 0 \text{ for } i \neq j, \quad \Delta = \alpha_j \mathbf{e}_j \mathbf{e}_j^\top (I_n - G). \quad (10)$$

Note that $\mathbf{e}_j \mathbf{e}_j^\top (I_n - G)$ is the matrix whose j th row is $(I_n - G)_{j,:}$ and all the other rows are equal to zero.

What $\hat{\boldsymbol{\mu}}$ can we choose to make $\Delta(\boldsymbol{\alpha}^*)$ rank-1? We can take $\hat{\boldsymbol{\mu}} = \frac{\boldsymbol{\mu} + \lambda \mathbf{e}_j}{(\boldsymbol{\mu} + \lambda \mathbf{e}_j)^\top \mathbf{1}_n} = \frac{1}{1+\lambda}(\boldsymbol{\mu} + \lambda \mathbf{e}_j)$ for any j and $\lambda > 0$. When $\lambda \rightarrow \infty$, the state j tends to become absorbing, that is, $\Delta_{j,:} \rightarrow \mathbf{e}_j^\top - G_{j,:}$, since $\hat{\boldsymbol{\mu}}_j \rightarrow 1$. Intuitively, we increase the probability of the state to return to itself, making it absorbing at the limit. In other words, by modifying the entries in one row of G , increasing the weight to oneself while reducing proportionally the weights to the others, we increase $\hat{\boldsymbol{\mu}}_j$, while decreasing $\hat{\boldsymbol{\mu}}_i$ for $i \neq j$ with the same ratio, $\frac{1}{1+\lambda} = \frac{1}{r^*}$.

Example 2.6. Consider Example 2.3:

$$G = \begin{bmatrix} 1/2 & 1/4 & 0 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \\ 1/4 & 0 & 1/4 & 1/2 \end{bmatrix}, \quad \boldsymbol{\mu}^\top = [1/4, 1/4, 1/4, 1/4].$$

Choosing $\hat{\boldsymbol{\mu}}^\top = [\frac{2}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}]$. We have $\boldsymbol{\mu}^\top ./ \hat{\boldsymbol{\mu}}^\top = [\frac{5}{8}, \frac{5}{4}, \frac{5}{4}, \frac{5}{4}]$, with optimal parameter $c^* = \frac{4}{5}$, so that $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(c^*) = [\frac{1}{2}, 0, 0, 0]^\top$, leading to

$$\Delta(\boldsymbol{\alpha}^*) = \begin{bmatrix} 1/4 & -1/8 & 0 & -1/8 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad G(\boldsymbol{\alpha}^*) = \begin{bmatrix} 3/4 & 1/8 & 0 & 1/8 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \\ 1/4 & 0 & 1/4 & 1/2 \end{bmatrix}.$$

This turns out to be a globally optimal solution of the TSDP that minimizes the component-wise ℓ_1 , as proved in the next paragraph.

Remark 2.7. The subdominant (that is, second largest in modulus) eigenvalue of G is 0.5, while that of $G(\boldsymbol{\alpha}^*)$ is 0.65, indicating an increased mixing time of the associated Markov chain. Intuitively, any perturbation causing a substantial increase in the diagonal entries will slow down time to convergence since a random walk will spend on average more time in each node, hence taking more time to visit all nodes. An interesting direction of further research is to look for perturbation that achieve a target distribution while controlling the subdominant eigenvalue. For example, instead of minimizing the norm of the perturbation, one could try to minimize the modulus of the subdominant eigenvalue, or combine these two objectives.

Optimality of the rank-1 solution $\Delta(\boldsymbol{\alpha}^*)$ As observed in Example 2.6, it turns out the solution $\Delta(\boldsymbol{\alpha}^*)$ is optimal for the TSDP in the rank-1 case, under the condition that $j \in \operatorname{argmax}_k \hat{\boldsymbol{\mu}}_k$, as stated in the following theorem.

Theorem 2.8. Let G be an irreducible stochastic matrix with stationary distribution $\boldsymbol{\mu} > 0$, and let us use the same notation as in Theorem 2.2. Assume $\hat{\boldsymbol{\mu}} = \frac{1}{1+\lambda}(\boldsymbol{\mu} + \lambda \mathbf{e}_j)$ for some $\lambda > 0$ and some j . Assume also that $\lambda \geq \mu_i - \mu_j$ for all $i \neq j$, that is, $j \in \operatorname{argmax}_k \hat{\boldsymbol{\mu}}_k$. Then $\Delta(\boldsymbol{\alpha}^*)$ where $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(c^*)$ is an optimal solution of the TSDP for the component-wise and matrix ℓ_1 norms.

Proof. By [5, Theorem 5.1], any feasible rank-one solution of the TSDP without the nonnegativity constraints has the form

$$\Delta(\mathbf{x}) = \frac{\mathbf{x} \hat{\boldsymbol{\mu}}^\top}{\hat{\boldsymbol{\mu}}^\top \mathbf{x}} (I - G) \quad \text{for } \mathbf{x} \in \mathbb{R}^n, \mathbf{x}^\top \hat{\boldsymbol{\mu}} \neq 0.$$

In fact, if $\Delta = \mathbf{xy}^\top$ for some vectors \mathbf{x} and \mathbf{y} , the constraints $\hat{\boldsymbol{\mu}}^\top \Delta = \hat{\boldsymbol{\mu}}^\top (I - G)$ leads to

$$\hat{\boldsymbol{\mu}}^\top \mathbf{xy}^\top = \hat{\boldsymbol{\mu}}^\top (I - G) \neq 0 \quad \Rightarrow \quad \mathbf{y}^\top = \frac{\hat{\boldsymbol{\mu}}^\top}{\hat{\boldsymbol{\mu}}^\top \mathbf{x}} (I - G), \text{ where } \hat{\boldsymbol{\mu}}^\top \mathbf{x} \neq 0.$$

Moreover, [5, Theorem 5.1] proved that there always exists an optimal rank-one solution to the TSDP without the nonnegativity constraints for any induced norm ; this follows from the fact that there exists a rank-one solution achieving the lower bound given in Lemma 2.4 for any Δ satisfying the constraint $\hat{\boldsymbol{\mu}}^\top (G + \Delta) = \hat{\boldsymbol{\mu}}^\top$ (see below for the formula of the rank-one solution for the case of the induced ℓ_1 norm). For the component-wise ℓ_1 norm, the proof can be found in Appendix B.

Hence solving $\min_{\mathbf{x}} \|\Delta(\mathbf{x})\|$ solves the TSDP under the rank-one constraint (that is, $\text{rank}(\Delta) = 1$), with the constraint $\Delta \mathbf{1}_n = \mathbf{1}_n$ because $\Delta(\mathbf{x})$ satisfies it automatically: $\Delta(\mathbf{x}) \mathbf{1}_n = \mathbf{xy}^\top \mathbf{1}_n = 0$, but possibly without the nonnegativity constraints.

Let us develop the above formula when $\hat{\boldsymbol{\mu}} = \frac{1}{1+\lambda}(\boldsymbol{\mu} + \lambda \mathbf{e}_j)$:

$$\Delta(\mathbf{x}) = \frac{\mathbf{x}(\boldsymbol{\mu} + \lambda \mathbf{e}_j)^\top}{(\boldsymbol{\mu} + \lambda \mathbf{e}_j)^\top \mathbf{x}} (I - G) = \lambda \frac{\mathbf{x} \mathbf{e}_j^\top}{\boldsymbol{\mu}^\top \mathbf{x} + \lambda x_j} (I - G) = \frac{\lambda}{\boldsymbol{\mu}^\top \mathbf{x} + \lambda x_j} \mathbf{x} (I - G)_{j,:},$$

which leads to

$$\lambda \left\| \frac{\mathbf{x} \mathbf{e}_j^\top}{\boldsymbol{\mu}^\top \mathbf{x} + \lambda x_j} (I - G) \right\| = \frac{\lambda}{|\boldsymbol{\mu}^\top \mathbf{x} + \lambda x_j|} \|\mathbf{x} (I - G)_{j,:}\|.$$

Let us start with the matrix ℓ_1 norm, that is, $\|A\|_{\ell_1} = \max_{\mathbf{z}} \frac{\|A\mathbf{z}\|_1}{\|\mathbf{z}\|_1}$. We have

$$\|\mathbf{x} (I - G)_{j,:}\|_{\ell_1} = \max_{\mathbf{z}} \frac{\|\mathbf{x} (I - G)_{j,:} \mathbf{z}\|_1}{\|\mathbf{z}\|_1} = \|\mathbf{x}\|_1 \max_{\mathbf{z}} \frac{\|(I - G)_{j,:} \mathbf{z}\|_1}{\|\mathbf{z}\|_1} = \|\mathbf{x}\|_1 \|(I - G)_{j,:}\|_{\infty}.$$

For the component-wise ℓ_1 norm, we have

$$\|\mathbf{x} (I - G)_{j,:}\|_1 = \|\mathbf{x}\|_1 \|(I - G)_{j,:}\|_1.$$

Since $\Delta(\mathbf{x})$ is independent of the scaling of \mathbf{x} , we can assume w.l.o.g. that $\|\mathbf{x}\|_1 = 1$. Hence $\min_{\mathbf{x}} \|\Delta(\mathbf{x})\|_*$ where $*$ $\in \{1, \ell_1\}$ is equivalent to solving

$$\min_{\mathbf{x}, \|\mathbf{x}\|_1=1} \frac{1}{|\boldsymbol{\mu}^\top \mathbf{x} + \lambda x_j|} = \max_{\mathbf{x}, \|\mathbf{x}\|_1=1} |\boldsymbol{\mu}^\top \mathbf{x} + \lambda x_j|,$$

which is minimized at $\mathbf{x}^* = \mathbf{e}_i$ for $i \in \text{argmax}_k \hat{\mu}_k$. (Note that this was proved in [5, Corollary 5.3] for the matrix ℓ_1 norm.) This gives

$$\Delta(\mathbf{x}^*) = \Delta(\mathbf{e}_i) = \frac{\lambda}{\mu_i + \lambda(\mathbf{e}_i)_j} \mathbf{e}_i (I - G)_{j,:},$$

meaning that the i th row of G corresponding to the largest entry in $\hat{\boldsymbol{\mu}}$ is perturbed in the direction $(I - G)_{j,:}$. Intuitively, the most influential node in G will increase its weight towards the j th node while reducing weights to others, proportionally to the j th node weights. This makes sense since we are trying to increase the value of μ_j to $\hat{\mu}_j$: we use the most influential node to do that.

Finally, this solution is nonnegative if $\text{supp}(I - G)_{j,:} \subset \text{supp}(I - G)_{i,:}$, which will typically not be the case when G is sparse and $i \neq j$. However, if $\lambda \geq \mu_i - \mu_j$ then $\hat{\mu}_j$ is the largest entry in $\hat{\boldsymbol{\mu}}$ so that $\mathbf{x}^* = \mathbf{e}_j$ and hence

$$\Delta(\mathbf{x}^*) = \frac{1}{\mu_j + \lambda} \mathbf{e}_j (\boldsymbol{\mu} + \lambda \mathbf{e}_j)^\top (I - G) = \frac{\lambda}{\mu_j + \lambda} \mathbf{e}_j \mathbf{e}_j^\top (I - G).$$

This implies that $G + \Delta(\mathbf{x}^*) \geq 0$ which is therefore an optimal solution of the TSDP for the component-wise and matrix ℓ_1 norms⁴. Recall that $\Delta(\boldsymbol{\alpha}^*) = \alpha_j \mathbf{e}_j \mathbf{e}_j^\top (I - G)$, see (10), and α_j was minimized to obtain the best possible solution of the TSDP of this form. Since the above rank-one solution is optimal, this means that $\Delta(\mathbf{x}^*) = \Delta(\boldsymbol{\alpha}^*)$. In fact, noting that $r^* = \frac{1}{1+\lambda}$ and $r_* = \frac{\mu_j}{\hat{\mu}_j} = \frac{\mu_j}{\mu_j} = \frac{(1+\lambda)\mu_j}{\mu_j + \lambda}$, we have

$$\alpha_j = 1 - \frac{r_*}{r^*} = 1 - \frac{\frac{(1+\lambda)\mu_j}{\mu_j + \lambda}}{1 + \lambda} = \frac{\lambda}{\mu_j + \lambda},$$

which concludes the proof. \square

Example 2.9. Let us take the final matrix from Example 2.6:

$$G = \begin{bmatrix} 3/4 & 1/8 & 0 & 1/8 \\ 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \\ 1/4 & 0 & 1/4 & 1/2 \end{bmatrix} \quad \text{with} \quad \boldsymbol{\mu} = \left[\frac{2}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5} \right].$$

Let $\hat{\boldsymbol{\mu}} = \frac{1}{1+\lambda}(\boldsymbol{\mu} + \lambda \mathbf{e}_2) = \left[\frac{4}{11}, \frac{3}{11}, \frac{2}{11}, \frac{2}{11} \right]$ for $\lambda = \frac{1}{10}$, so that $\hat{\mu}_1 > \hat{\mu}_2$. We have

$$\Delta(\mathbf{x}^*) = \begin{bmatrix} -11/160 & 11/80 & -11/160 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \neq \Delta(\boldsymbol{\alpha}^*) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1/12 & 1/6 & -1/12 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

while

$$\Delta^* = \underset{\Delta \in \mathcal{D}}{\text{argmin}} \|\Delta\|_1 = \begin{bmatrix} -1/16 & 1/16 & 0 & 0 \\ 0 & 1/12 & -1/12 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

where $\|\Delta(\mathbf{x}^*)\|_1 = \frac{11}{40} < \|\Delta^*\|_1 = \frac{7}{24} < \|\Delta(\boldsymbol{\alpha}^*)\|_1 = \frac{1}{3}$, but $\Delta(\mathbf{x}^*)$ is not admissible as $G + \Delta(\mathbf{x}^*) \not\geq 0$. Note that Δ^* was computed using linear optimization; see section 3.

2.4 Ordering of $\boldsymbol{\mu}$

In this section we show how to reduce the difference $r^* - r_*$ by introducing a permutation on the vector $\hat{\boldsymbol{\mu}}$, and hence the norm of $\Delta(\boldsymbol{\alpha}^*)$ is expected to be smaller; see Equation (7). In other words, since $\hat{\boldsymbol{\mu}}$ is a parameter to be chosen by the user, it is interesting to know in which situation $\Delta(\boldsymbol{\alpha}^*)$ is expected to have a smaller norm to achieve the target $\hat{\boldsymbol{\mu}}$. To do so, we derive a result on the orderings of the stationary distributions $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$ that minimizes $r^* - r_*$.

⁴This case is analyzed in [5, Section 7]. However, authors do not mention the condition $\lambda \geq \max_i \mu_i - \mu_j$ to obtain [5, Equation (7.2)].

Theorem 2.10. *Let P and \hat{P} be $n \times n$ permutation matrices that reorder the elements of $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$, respectively, meaning that the elements of the permuted vectors $P\boldsymbol{\mu}$ and $\hat{P}\hat{\boldsymbol{\mu}}$, each form a non-decreasing series. Then*

$$[\min_i (P\boldsymbol{\mu})_i / (\hat{P}\hat{\boldsymbol{\mu}})_i, \max_i (P\boldsymbol{\mu})_i / (\hat{P}\hat{\boldsymbol{\mu}})_i] \subset [\min_i \mu_i / \hat{\mu}_i, \max_i \mu_i / \hat{\mu}_i].$$

Proof. See Appendix A. □

This theorem shows that when the vectors $P\boldsymbol{\mu}$ and $\hat{P}\hat{\boldsymbol{\mu}}$ have “coherent” orderings, then the corresponding interval $[r_*(P\boldsymbol{\mu}, \hat{P}\hat{\boldsymbol{\mu}}), r^*(P\boldsymbol{\mu}, \hat{P}\hat{\boldsymbol{\mu}})]$ is minimal, and hence the norm $\|\boldsymbol{\alpha}(c^*)\|$ is expected to be smaller. Note that the permutation of $\boldsymbol{\mu}$ can be achieved by a permutation on the rows of the matrix G , but this is *not* a similarity transformation.

In summary, our result indicates that the required perturbation Δ is expected to be smaller when $\hat{\boldsymbol{\mu}}$ is coherently ordered with $\boldsymbol{\mu}$.

3 Formulation with linear optimization

In the previous section, we focused on solutions of the TSDP of the form $\Delta(\boldsymbol{\alpha}) = \text{Diag}(\boldsymbol{\alpha})(I_n - G)$ with support in $\text{supp}(G + I_n)$. In this section, we propose an efficient linear optimization formulation of the TSDP with support constraint and for the component-wise ℓ_1 norm. This allows us to solve large problems efficiently, especially when Ω or G are sparse.

3.1 Linear optimization formulation

The TSDP with support constraint $\text{supp}(\Delta) \subset \Omega$ and for the component-wise ℓ_1 norm can be formulated as follows:

$$\begin{aligned} \min_{\Delta \in \mathbb{R}^{n \times n}} \|\Delta\|_1 \quad \text{such that} \quad & \Delta \mathbf{1}_n = \mathbf{0}_n, \\ & \hat{\boldsymbol{\mu}}^\top \Delta = \hat{\boldsymbol{\mu}}^\top (I - G), \\ & \Delta + G \geq 0, \\ & \Delta_{i,j} = 0 \text{ for } (i, j) \notin \Omega. \end{aligned} \tag{11}$$

We will refer to this problem as ℓ_1 -TSDP with support constraint. Since the entries of Δ are non-zero only in the set Ω , (11) has only $|\Omega|$ variables (the zero entries of Δ are not variables). In the dense case, that is, when $\Omega = \{(i, j) \mid 1 \leq i, j \leq n\}$, a naive way to formulate (11) is to introduce n^2 variables, say $Z \in \mathbb{R}_+^{n \times n}$, one for each entry of Δ , impose $Z \geq \Delta$ and $Z \geq -\Delta$ and minimize $\sum_{i,j} Z_{i,j}$; this is the standard way to linearize the ℓ_1 norm. However, on top of introducing n^2 variables, it introduces $2n^2$ inequalities which is highly ineffective, since solving linear optimization problems requires, roughly speaking, a cubic number of operations in the number of variables or in the number of constraints⁵ (solving the dual); see [11] and the references therein. For example, the problem was formulated in this way in [5, Equation (8.1)], and the formulation of the authors using a commercial solver, Gurobi, could not solve problems larger than $n = 500$ within 10 minutes. The reason is that [5] did not consider the component-wise ℓ_1 norm, but matrix norms for which it is not possible to avoid the introduction of $O(|\Omega|)$ inequalities to model the problem via linear optimization.

⁵More precisely, it requires $\tilde{O}(n^\omega \log(n/\delta))$ time where $O(n^\omega)$ is the time to multiply two n -by- n matrices, δ is the relative accuracy, and \tilde{O} ignores polylog(n) factors.

Let us reformulate the problem in a more cost-effective way. First, it is interesting to note that whenever $G_{i,j} = 0$, we have $\Delta_{i,j} \geq 0$ because of the constraints $G + \Delta \geq 0$, and hence there is no need to introduce an additional variable for $\Delta_{i,j}$ when $G_{i,j} = 0$ to minimize $|\Delta_{i,j}|$. Let us denote the support of G as $\text{supp}(G)$, its complement as $\overline{\text{supp}(G)}$ (that is, the set of index pairs corresponding to zero entries of G), and let us decompose Δ using three nonnegative terms: $\Delta = \Delta^0 + (\Delta^+ - \Delta^-)$, where

$$\text{supp}(\Delta^0) \subset \Omega^0 := \Omega \cap \overline{\text{supp}(G)},$$

and

$$\text{supp}(\Delta^+), \text{supp}(\Delta^-) \subset \Omega^\pm := \Omega \cap \text{supp}(G).$$

Note that Ω^0 and Ω^\pm is a partition of Ω : $\Omega^\pm \cup \Omega^0 = \Omega$ and $\Omega^\pm \cap \Omega^0 = \emptyset$. Hence $\Delta^0(i,j) \geq 0$ can be non-zero when $(i,j) \in \Omega$ and $G_{i,j} = 0$, while $\Delta^+(i,j), \Delta^-(i,j) \geq 0$ can be non-zero when $(i,j) \in \Omega$ and $G_{i,j} > 0$. We will denote the number of non-zero entries of Δ^0 by $p^0 = |\Omega \cap \overline{\text{supp}(G)}| \leq |\Omega|$, and the number of non-zero entries in Δ^+ and Δ^- by $p^\pm = |\Omega \cap \text{supp}(G)|$. With these new variables, the TSDP (11) can be reformulated as follows

$$\begin{aligned} & \min_{\Delta^0, \Delta^+, \Delta^- \in \mathbb{R}_+^{n \times n}} \sum_{i,j} \Delta_{i,j}^0 + \Delta_{i,j}^+ + \Delta_{i,j}^- \\ & \text{such that } (\Delta^0 + \Delta^+ - \Delta^-) \mathbf{1}_n = \mathbf{0}_n, \\ & \hat{\boldsymbol{\mu}}^\top (\Delta^0 + \Delta^+ - \Delta^-) = \mathbf{z} := \hat{\boldsymbol{\mu}}^\top (I - G), \\ & \Delta^0 \leq \mathbf{1}_{n,n}, \Delta^- \leq G, \Delta^+ \leq \mathbf{1}_{n,n} - G, \\ & \text{supp}(\Delta^0) \subset \Omega^0, \text{supp}(\Delta^+), \text{supp}(\Delta^-) \subset \Omega^\pm. \end{aligned} \tag{12}$$

The constraint $\Delta^- \leq G$ ensures that $G + \Delta$ is nonnegative. Since $G + \Delta$ is also stochastic, as $\Delta \mathbf{1}_n = 0$, the constraints $\Delta^0 \leq \mathbf{1}_{n,n}$ and $\Delta^+ \leq \mathbf{1}_{n,n} - G$ are redundant. We have not observed much impact on the solver by removing or using these upper bound constraints. Note that we are minimizing the sum of the entries of Δ^+ and Δ^- , and hence we will always have at optimality that the supports of Δ^+ and Δ^- are disjoint.

In any case, (12) has only $2n$ equalities and less than $2|\Omega|$ bounded variables. The fact that we have only $2n$ equalities allows us to solve this problem in roughly $O(n^3)$ operations via the dual. This is a significant improvement. Moreover, each inequality has at most n variables, and hence this is a very sparse linear optimization problem, even sparser when the set Ω is sparse. This will allow commercial solvers, such as Gurobi, to solve such problems much faster than the worst case $O(n^3)$ operations; for example, we will be able to solve problems with $n = 10^5$ with Ω having a few non-zero entries per row in a few minutes; see section 4 for experiments.

3.2 Sparsity of ℓ_1 norm minimization

The optimal solution of (12) will have some degree of sparsity. In this section, we make this statement precise by providing an explicit upper bound on the number of non-zero entries in an optimal solution; see Theorem 3.1 below.

Let us rewrite (12) in a more standard form, by vectorizing the non-zero entries of Δ^0 in $\mathbf{x}^0 \in \mathbb{R}^{p^0}$, of Δ^+ in $\mathbf{x}^+ \in \mathbb{R}^{p^+}$, and of Δ^- in $\mathbf{x}^- \in \mathbb{R}^{p^-}$, and let $\mathbf{x} = (\mathbf{x}^0, \mathbf{x}^+, \mathbf{x}^-) \in \mathbb{R}^p$:

$$\min_{\mathbf{x} = (\mathbf{x}^0, \mathbf{x}^+, \mathbf{x}^-) \in \mathbb{R}^p} \mathbf{1}_n^\top \mathbf{x} \quad \text{such that} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, \mathbf{x}^- \leq \mathbf{g}, \tag{13}$$

where $p = p^0 + 2p^\pm$, $A \in \mathbb{R}^{2n \times p}$ contains the coefficient of the $2n$ linear equalities and has two non-zero entries per column (since each variable appears in exactly two constraints: $\Delta_{i,j}$ appears in $\Delta(i, :)\mathbf{1}_n = 0$ and $\boldsymbol{\mu}^\top \Delta(:, j) = \mathbf{z}$), $\mathbf{b} = [\mathbf{0}_n; \mathbf{z}]$ is the right-hand side, and $\mathbf{g} \in \mathbb{R}^{p^\pm}$ contains the non-zero entries of G in the set Ω .

Theorem 3.1. *If (13) is feasible, there exists an optimal solution, \mathbf{x}^* , of (13) such that*

$$|\text{supp}(\mathbf{x}^*)| \leq \min(|\Omega|, |\text{supp}(G \cap \Omega)| + 2n).$$

Equivalently, if the ℓ_1 -TSDP with support constraint (11) is feasible, there exists an optimal solution of (11), Δ^ , such that*

$$|\text{supp}(\Delta^*)| \leq \min(|\Omega|, |\text{supp}(G \cap \Omega)| + 2n).$$

Proof. The feasible set of (13), $\mathcal{P} = \{\mathbf{x} = (\mathbf{x}^0, \mathbf{x}^+, \mathbf{x}^-) \mid A\mathbf{x} = \mathbf{b}, x \geq 0, \mathbf{x}^- \leq \mathbf{g}\}$, is bounded, and hence is a polytope. Therefore, by the fundamental theorem of linear optimization, if \mathcal{P} is non-empty, that is, (13) is feasible, there exists an optimal solution which is a vertex of \mathcal{P} , say \mathbf{x}^* .

Let us first show $|\text{supp}(\mathbf{x}^*)| \leq |\Omega|$. For any optimal solution, $\mathbf{x} = (\mathbf{x}^0, \mathbf{x}^+, \mathbf{x}^-)$, we must have $x_i^+ = 0$ or $x_i^- = 0$ for all i , otherwise we can strictly decrease the objective. This is because the entries of $\mathbf{x}^+ - \mathbf{x}^-$ are equal to the entries of Δ in $\text{supp}(G) \cap \Omega$. This means that $|\text{supp}([\mathbf{x}^+, \mathbf{x}^-])| \leq p^\pm = |\text{supp}(G) \cap \Omega|$, and hence $|\text{supp}(\mathbf{x})| \leq |\Omega|$.

It remains to prove that $|\text{supp}(\mathbf{x}^*)| \leq |\text{supp}(G)| + 2n$. Recall that any vertex of a p -dimensional polytope must have p linearly independent active constraints. Since there are (only) $2n$ equalities in (13), there must be $p - 2n$ of the bound constraints active at any vertex of \mathcal{P} . Let $\mathbf{x} = (\mathbf{x}^0, \mathbf{x}^+, \mathbf{x}^-)$ be a vertex of \mathcal{P} . Each entry in $\mathbf{x}^+ \geq 0$ and $0 \leq \mathbf{x}^- \leq \mathbf{g}$ can touch at most one bound constraint (the entries in \mathbf{x}^- cannot touch two of them, since by definition $\mathbf{g} > 0$). This is possible if $x_i^+ = 0 = x_i^-$, or if $x_i^+ = 0$ and $x_i^- = g_i > 0$. This means that x^0 must touch at least $p - 2n - 2p^\pm = p^0 - 2n$ of the lower bound constraints, $x^0 \geq 0$, that is, x^0 must have at least $p^0 - 2n$ entries equal to zero. This implies that $|\text{supp}(\mathbf{x}^0)| \leq 2n$, and gives the result, since $|\text{supp}([\mathbf{x}^+, \mathbf{x}^-])| \leq |\text{supp}(G \cap \Omega)|$ at optimality; see above. □

Theorem 3.1 implies that if G or Ω is sparse, then there is an optimal solution of (11) which is sparse. To attain the upper bound $|\text{supp}(G \cap \Omega)| + 2n$, Δ must be negative in all entries of $\text{supp}(G)$ as it requires $\Delta^- = G$. This is unlikely to happen at optimality since we are minimizing the sum of these variables. In fact, in all cases we have tested, optimal solutions are significantly sparser; see the numerical experiments in section 4.

3.3 Efficient algorithms via a column-generation approach

The number of columns of A in (13) is $O(|\Omega|)$. If Ω is sparse, we can formulate the problem directly and solve it very efficiently via commercial solvers; see section 4 for numerical experiments.

However, if Ω is dense, for example $\Omega = \{(i, j) \mid 1 \leq i, j \leq n\}$, constructing the $O(n^2)$ columns of the constraint matrix A is time consuming, even if A is sparse (2 non-zeros per column). This is particularly wasteful if G is sparse since we know only a few entries in the optimal solution Δ will be non-zero, namely at most $|\text{supp}(G)| + 2n$ (Theorem 3.1).

This calls for a column-generation approach: First solve the problem for a well-chosen sparse Ω . Then progressively add index pairs in Ω that will reduce the objective function.

Choice of initial Ω Given a set Ω , deciding whether one can change the stationary distribution from $\boldsymbol{\mu}$ to $\hat{\boldsymbol{\mu}}$ has been studied in [13, 12]. However, finding a small initial support such that the problem is feasible, while avoiding a horrendous enumeration, appears to be a non-trivial question, to the best of our knowledge. Luckily, we know that $\Omega = \text{supp}(G + I)$ is always feasible (Theorem 2.2), and hence we will initialize Ω with the support of $G + I$. For G dense, it is unclear how to initialize Ω with a sparse matrix to make the problem feasible, and this is probably not as useful as the optimal solution is not necessarily very sparse.

Choice of entries to add to Ω To add index pairs in Ω that will reduce the objective, we can resort to the well-known column-generation approach in linear optimization, see [28] and the references therein. Let us recall the basic idea of column generation. We are trying to solve a linear optimization problem of the form

$$\min_{\mathbf{x}=(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^{p_1 \times p_2}} \mathbf{c}_1^\top \mathbf{x}_1 + \mathbf{c}_2^\top \mathbf{x}_2 \quad \text{such that} \quad [A_1, A_2] \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \mathbf{b}, \mathbf{x} \geq 0,$$

where $p = p_1 + p_2$, $A_i \in \mathbb{R}^{m \times p_i}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c}_i \in \mathbb{R}^{p_i}$ for $i = 1, 2$. The dual is given by

$$\max_{\mathbf{y} \in \mathbb{R}^m} \mathbf{b}^\top \mathbf{y} \quad \text{such that} \quad A_1^\top \mathbf{y} \leq \mathbf{c}_1 \text{ and } A_2^\top \mathbf{y} \leq \mathbf{c}_2.$$

Assume we solve the primal without the variables \mathbf{x}_2 so that the constraint $A_2^\top \mathbf{y} \leq \mathbf{c}_2$ disappears from the dual, and obtain $(\mathbf{x}_1^*, \mathbf{y}^*)$ as an optimal primal-dual solution. If $A_2^\top \mathbf{y}^* \leq \mathbf{c}_2$, that is, the dual solution, \mathbf{y}^* , remains feasible with the additional variables \mathbf{x}_2 being taken into account, then $(x_1^*, 0)$ is an optimal solution to the original problem, by strong duality. If this is not the case, that is, $A_2(:, i)^\top \mathbf{y}^* > c_2(i)$ for some i , then allowing the variable $x_2(i)$ to enter the solution will allow us to reduce the objective (this is the rationale of a pivoting step in the simplex algorithm). The quantity $\mathbf{c}_2 - A_2^\top \mathbf{y}^*$ are the so-called reduced costs of the variables \mathbf{x}_2 which needs to be nonnegative at optimality.

Let us apply this idea to our problem. Let $\mathbf{y}^0 \in \mathbb{R}^n$ be the dual variables associated to the constraints $\Delta \mathbf{1}_n = 0$, and $\mathbf{y}^\mu \in \mathbb{R}^n$ to the constraints $\Delta^\top \hat{\boldsymbol{\mu}} = \mathbf{z}$. Since, at the first step, $\Omega = \text{supp}(G + I)$, we will add entries with index $(i, j) \notin \Omega$ such that $G_{i,j} = 0$. Adding (i, j) in Ω , that is, adding a variable in the primal corresponding to $\Delta_{i,j}^0$, adds a constraint in the dual, namely $y_i^0 + \hat{\mu}_j y_j^\mu \leq 1$. In fact, in the primal, the coefficient of the variable $\Delta_{i,j}^0$ is only non-zero for two constraints: $\Delta_{i,:} \mathbf{1}_n = 0$ and $\Delta_{:,j}^\top \hat{\boldsymbol{\mu}} = z_j$. If this dual constraint is violated, that is, $y_i^0 + \hat{\mu}_j y_j^\mu > 1$, this means that the current primal solution is not optimal for $\Delta_{i,j}^0 = 0$ (the reduced cost are not nonnegative for the current basis). Hence adding variables (i, j) in Ω for which $y_i^0 + \hat{\mu}_j y_j^\mu > 1$ will reduce the objective function value. However, it is not possible for a generic solver to perform such a task without generating explicitly all constraints in the dual (and this takes time when Ω is dense); this is because the solver cannot guess the structure of our problem.

To take advantage of these observations, we adopt the following strategy: We initialize $\Omega^0 = \text{supp}(G + I)$. We solve (11), and then we add to Ω the $|\Omega^0|$ largest entries from the rank-two matrix $\mathbf{y}^0 \mathbf{1}_n^\top + \hat{\boldsymbol{\mu}} (\mathbf{y}^\mu)^\top$ that are larger than 1. We provide the commercial solver, Gurobi, with the previous basis, for a clever initialization. We stop adding new variables when the decrease⁶ in the relative error between two iterations is smaller than some prescribed accuracy δ , that is, $\frac{\|\Delta^{\text{old}}\|_1 - \|\Delta^{\text{new}}\|_1}{\|G\|_1} < \delta$, or if

⁶The error can only decrease between two iterations since the support of the optimal solution of (11) at iteration $i + 1$, Ω^{i+1} , contains the support from the previous iteration, Ω^i ; see step 7 in Algorithm 1.

the CG has converged, that is, $\mathbf{y}^0 \mathbf{1}_n^\top + \hat{\boldsymbol{\mu}}(\mathbf{y}^\mu)^\top \leq \mathbf{1}_{n,n}$. We will use $\delta = 1\%$ and $\delta = 0.01\%$ in the numerical experiments.

Algorithm 1 summarizes our strategy, which is a batch column-generation approach. It is more effective in MATLAB because adding only a single variable at each iteration would lead to many calls of the Gurobi solver and would be ineffective. Ideally, one should implement the strategy within Gurobi, in C, but this is out of the scope of this paper.

Algorithm 1 Column-Generation based algorithm for the TSDP (11)

Input: A stochastic irreducible matrix $G \in \mathbb{R}^{n \times n}$, a positive target distribution $\hat{\boldsymbol{\mu}} \in \mathbb{R}_+^n$, a set Ω of interest (default: $\{(i, j) \mid 1 \leq i, j \leq n\}$), a target relative accuracy $0 \leq \delta < 1$ (default: 10^{-4}).

- 1: $\text{obj}_{-1} = +\infty$,
 - 2: $\Delta^0 = \Delta(\boldsymbol{\alpha}^*)$, $\text{obj}_0 = \|\Delta^0\|_1$
 - 3: Initialize $\Omega^1 = \text{supp}(G + I) \cap \Omega$, $n^+ = |\Omega^1|$, $\Omega^+ = \Omega^1$, $i = 1$.
Note: Ω^1 might not be feasible if Ω does not contain $\text{supp}(G + I)$.
 - 4: **while** $\text{obj}_{i-2} - \text{obj}_{i-1} > \delta \|G\|_1$ and $\Omega^+ \neq \emptyset$ **do**
 - 5: Solve (11) with $\Omega = \Omega^i$ starting from Δ^{i-1} , and obtain the optimal solution Δ^i with optimal dual variables \mathbf{y}^0 and \mathbf{y}^μ , let $\text{obj}_i = \|\Delta^i\|_1$.
 - 6: Construct the matrix $R = \mathbf{y}^0 \mathbf{1}_n^\top + \hat{\boldsymbol{\mu}}(\mathbf{y}^\mu)^\top - \mathbf{1}_{n,n}$, and let Ω^+ be the set of index pairs in R corresponding to the n^+ largest positive entries in R .
Note: R could have less than n^+ positive entries, in which case we only add the positive ones. In fact, if Δ^i is globally optimal, $\Omega^+ = \emptyset$.
 - 7: Set $\Omega^{i+1} = \Omega^i \cup \Omega^+$, $i \leftarrow i + 1$.
 - 8: **end while**
-

Computational cost and key improvement in the selection strategy Solving (11) requires to solve linear optimization problems with $2n$ constraints. This requires roughly $O(n^3)$ operations in the worst case; see section 3.1. However, we will see that that it is empirically much faster as this linear program is very sparse, especially when G is; see section 4.1.2 for numerical experiments.

In terms of memory, the constraint matrix $A \in \mathbb{R}^{2n \times O(|\Omega|)}$ actually only requires $O(|\Omega|)$ memory as there are two non-zeros per column of A . Empirically, we will see that a few iterations of Algorithm 1 (in most cases, less than 10) are enough to obtain good solutions. Hence $|\Omega| = O(\text{nnz}(G))$, and the memory requirement is proportional to that of storing G .

For large n , we noticed that constructing explicitly the rank-two matrix $T = \mathbf{y}^0 \mathbf{1}_n^\top + \hat{\boldsymbol{\mu}}(\mathbf{y}^\mu)^\top$ to extract the index pairs corresponding to the largest entry was a bottleneck in the algorithm (step 6): it requires $O(n^2)$ memory and $O(n^2 \log(n^2))$ operations (to sort the entries of T). This rank-two matrix has positive factors in each rank-one term, namely $\mathbf{1}_n$ and $\hat{\boldsymbol{\mu}}$, and hence the largest entries of T depend directly on the largest entries of \mathbf{y}^0 , \mathbf{y}^μ and $\hat{\boldsymbol{\mu}}$. We have designed a simple heuristic, when n is large, to tackle this problem: for a parameter $m \leq n/2$, extract the m largest indices in \mathbf{y}^0 and the m largest indices in $\hat{\boldsymbol{\mu}}$ and put them in \mathcal{I} (possibly remove duplicates), and extract the $2m$ largest indices in \mathbf{y}^μ and put them in \mathcal{J} . Then perform an exhaustive search on the submatrix $T(\mathcal{I}, \mathcal{J})$ which requires $O(m^2 \log(m^2))$ operations. We used $m = 200$ in our implementation, and use this heuristic as soon as $n > 200$. This means that, when n is large, we add at most $m^2 = 40,000$ entries in Ω at each step. Note that more sophisticated heuristics exist⁷, e.g., [21] and [4]. However, our current heuristic

⁷We actually tried the heuristic from [21], but it does not scale well to extract a large number of index pairs as it

performs extremely well on all the tested cases. For example, we tested against the exhaustive search for the real data set *moreno* (with $n = 2155$; see section 4.2): Algorithm 1 with $\delta = 0$ (that is, an optimal solution is sought) took 41 seconds with an exhaustive search in step 6, and only 5.6 seconds with our heuristic, to obtain the same optimal objective function value.

4 Numerical experiments

We will compare the following solutions:

- *MH*: the Metropolis-Hastings algorithm; see section 1 and in particular equation (3).
- *D*: the closed-form solution $\Delta(\alpha^*)$ described in Theorem 2.2. This solution is computed extremely fast, requiring $O(\text{nnz}(G))$ operations.
- *S*: The optimal solution of (11) with $\Omega = \text{supp}(G + I)$. *S* stands for sparse.
- *GS*: The optimal solution of (11) with $\Omega = \{(i, j) \mid 1 \leq i, j \leq n\}$ computed by solving directly (11). *GS* stands for global solution.
- *CG*(δ): The solution of (11) with $\Omega = \{(i, j) \mid 1 \leq i, j \leq n\}$ computed with Algorithm 1 with stopping criterion δ ; we will use $\delta \in \{10^{-2}, 10^{-4}, 0\}$. *CG* stands for column generation. The case $\delta = 0$ should generate a solution with the same objective as *GS*, but hopefully significantly faster when *G* is sparse. Note that choosing δ sufficiently large generates the solution *S* since Algorithm 1 is initialized with $\Omega = \text{supp}(G + I)$.

By construction, we know in advance that, in terms of objective function values, we have

$$\|D\|_1 \geq \|S\|_1 \geq \|CG(\delta)\|_1 \geq \|CG(0)\|_1 = \|GS\|_1.$$

In fact, *S* is the optimal solution for $\Omega = \text{supp}(G + I)$, *D* is a feasible solution for that support, and *CG*(δ) is initialized with $\Omega = \text{supp}(G + I)$. It will be interesting to compare the gaps between these objectives, and the computational times. Moreover, $\|MH\|_1 \geq \|S\|_1$ since the support of *MH* is contained in that of *G + I*; see section 1.

We do not compare with the algorithms in [5] because they are not competitive. For example, for the 3 real data sets they consider, their fastest approach able to compute a feasible solution in all cases (namely, the rank-1 steps heuristic–R1SH) takes 228, 50 and 75 seconds, respectively; see [5, Table 7]. The solution we will generate with *CG*(0) is computed faster (less than 6 seconds in all cases), is sparse, and is globally optimal; see Tables 3, 4 and 5. Moreover, the solutions in [5] do not minimize the component-wise ℓ_1 norm, but matrix norms, and do not produce sparse solutions; in fact, they produce dense low-rank solutions. To cite their own words,

“The applicability of our heuristics for sparse *G* is hindered. Being able to perturb only one row in a stochastic matrix, it is not hard to imagine that the number of reachable stochastic matrices is limited. In other words, finding a rank-1 perturbation towards a specific stationary distribution (the main focus of this paper) is often infeasible.”

uses a dense n -by- p matrix, where p is the number of index pairs to extract. In our case, p is of the order of n making this heuristic require $O(n^2)$ memory and operations, which is what we need to avoid.

Quality measures We will report the following three quantities for each solution Δ :

- **obj**: the relative objective value $\frac{\|\Delta\|_1}{\|G\|_1}$,
- **spars**: the relative sparsity $\frac{|\text{supp}(\Delta)|}{|\text{supp}(G+I)|}$, and
- **time**: the computational time in seconds.

Softwares and machine All experiments are performed on a laptop with processor 12th Gen Intel(R) Core(TM) i9-12900H 2.50 GHz, 32Go RAM, on MATLAB R2019b. The code is available from <https://gitlab.com/ngillis/TSDP>. To solve the linear optimization problems, we use the commercial solver Gurobi⁸, version 10.00.

4.1 Synthetic data sets

In this section, we mostly focus on the run time of the proposed algorithms depending on the size and sparsity of the input matrix, and the gap in the objective function between the different approaches. This will allow us to estimate, empirically, their computational cost and distance to optimality.

Getting inspiration from queuing matrices, where each node is connected only to its two neighbors, one on the left and one on the right, we generate irreducible matrices as follows: Given a parameter k , each node is connected to its k neighbors on the right and to its k neighbors on the left. Each non-zero entry is randomly generated using the uniform distribution in $[0,1]$, and G is then normalized to make it stochastic. Hence each row of G has $2k$ non-zero entries (except for the first k and last rows k that do not have enough left and right neighbors, respectively). Here is an example of such a matrix generated randomly with parameters $n = 5$ and $k = 2$ (and two digits of accuracy):

$$G = \begin{bmatrix} 0 & 0.32 & 0.68 & 0 & 0 \\ 0.18 & 0 & 0.39 & 0.43 & 0 \\ 0.26 & 0.32 & 0 & 0.30 & 0.12 \\ 0 & 0.33 & 0.32 & 0 & 0.35 \\ 0 & 0 & 0.28 & 0.72 & 0 \end{bmatrix}.$$

We set the vector⁹ $\hat{\boldsymbol{\mu}} = G^\top \frac{\mathbf{1}_n}{n}$, that is, we initialize $\hat{\boldsymbol{\mu}}$ with equal probability for all nodes, then apply once step of the power iteration. This makes $\hat{\boldsymbol{\mu}}$ somewhat closer to $\boldsymbol{\mu}$. Note that the sparsity pattern of G is symmetric, and hence MH is guaranteed to generate a feasible solution.

Remark 4.1 (Choice of $\hat{\boldsymbol{\mu}}$). *There are plenty of other possible choices for $\hat{\boldsymbol{\mu}}$, and ours here is somewhat arbitrary. It balances the importance of the nodes in the graph, as $\hat{\boldsymbol{\mu}}$ is close to $\frac{\mathbf{1}_n}{n}$. For example, in social networks, this corresponds to make everyone have a similar popularity, and in road networks to make all roads equally congested.*

Another example, that we considered in section 2.3, is the choice $\hat{\boldsymbol{\mu}} = \frac{1}{1+\lambda}(\boldsymbol{\mu} + \lambda \mathbf{e}_j)$ for some j and $\lambda > 0$. This is an interesting perturbation since it increases the importance of a targeted node in the chain, e.g., to increase the popularity of a person in a social network, or increase the congestion in a

⁸Although it is a closed-source commercial software, see <https://www.gurobi.com/>, free use is possible for academic users.

⁹We also made experiments with $\hat{\boldsymbol{\mu}} = \frac{\mathbf{1}_n}{n}$. It produces somewhat similar results, but the differences between the solutions $CG(\delta)$ were slightly less pronounced, and hence we chose to report the results for $\hat{\boldsymbol{\mu}} = G^\top \frac{\mathbf{1}_n}{n}$.

road with little traffic. However, this leads to less interesting numerical results since we have proved that, in that case, the solution D , that is, the closed-form solution $\Delta(\boldsymbol{\alpha}^*)$ described in Theorem 2.2, leads to a sparse optimal rank-one solution as soon as λ is sufficiently large; see Theorem 2.8.

4.1.1 Comparison of the different solutions

We let $n = 1000$ and $k \in \{1, 2, 5, 10, 50, 100, 500\}$, and Table 1 reports the relative objective values (obj), the average relative sparsity (spars) and the computational time for 10 randomly generated matrices.

		MH	D	S	$CG(10^{-2})$	$CG(10^{-4})$	$CG(0)$	GS
$k = 1$	obj (%)	43.76	178.88	23.10	22.73	11.48	11.47	11.47
	spars (%)	62.54	99.90	58.91	58.87	46.03	45.93	45.90
	time (s.)	0.01	0.20	0.03	0.06	0.99	1.57	25.53
$k = 2$	obj (%)	58.71	175.53	11.62	11.21	7.51	7.51	7.51
	spars (%)	59.93	99.90	31.92	31.62	25.29	25.29	25.29
	time (s.)	< 0.01	0.19	0.04	0.08	0.63	0.79	25.16
$k = 5$	obj (%)	64.13	160.52	3.89	3.75	3.10	3.10	3.10
	spars (%)	54.56	99.90	12.91	12.69	10.88	10.88	10.88
	time (s.)	< 0.01	0.04	0.08	0.19	0.74	1.30	25.03
$k = 10$	obj (%)	65.23	105.36	1.87	1.82	1.69	1.69	1.69
	spars (%)	52.39	99.90	6.21	6.14	5.50	5.50	5.50
	time (s.)	< 0.01	0.02	0.22	0.48	1.57	1.97	25.61
$k = 50$	obj (%)	66.78	19.72	1.02	0.94	0.94	0.94	0.94
	spars (%)	50.51	99.90	1.52	1.41	1.40	1.40	1.40
	time (s.)	0.03	0.01	2.68	3.41	4.10	4.81	28.27
$k = 100$	obj (%)	66.57	13.65	1.40	1.39	1.39	1.39	1.39
	spars (%)	50.26	99.89	1.21	1.20	1.20	1.20	1.20
	time (s.)	0.07	0.01	16.19	18.57	18.57	21.01	31.74
$k = 500$	obj (%)	66.74	0.24	0.03	0.03	0.03	0.03	0.03
	spars (%)	50.05	99.90	0.11	0.11	0.11	0.11	0.11
	time (s.)	0.04	0.02	57.29	57.29	57.29	57.29	57.43

Table 1: Experiments on synthetic data sets reporting the average values for ten randomly generated queue-like transition matrices (with about $2k$ non-zero entries per row) and $\hat{\boldsymbol{\mu}} = G^\top \mathbf{1}_n/n$.

We observe the following.

- In terms of relative objective values, D performs rather poorly for small values of k . In particular, for $k \leq 10$, its relative error is larger than 100% while the globally optimal solution, GS , is below 11% in all cases. For $k = 500$, when G is dense, it actually provides a good solution, as predicted by theory because $\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2$ is small (8) (the optimal Δ has a very small norm, 0.03% of that of G). MH does provide reasonable solutions for small k , although far from global optimality, and performs the worst for $k \geq 50$. The solution S provides rather good solutions, especially when $k \geq 5$. This shows that restricting the solution to the support of $G + I$ allows us to recover reasonable solutions, compared to the global one. $CG(10^{-2})$ performs similarly as S ,

while $CG(10^{-4})$ provides solutions of similar quality as $CG(0)$ which provides globally optimal solutions, as GS does.

- In terms of sparsity, the support of D coincide with that of $G + I$ except for one row, as expected (see Lemma 2.5), explaining the 99.9% relative sparsity in all cases (since $n = 1000$). The sparsity of MH is around 50% of that of $G + I$ because the M-H algorithm does not use the full support (some rows of Δ are zeros). All the other solutions are significantly sparser than $G + I$, especially when k grows. This shows that minimizing the component-wise ℓ_1 norm is able to promote sparsity; see Theorem 3.1. For example, when G is dense ($k = 500$), the optimal solution only has 0.11% of non-zero entries. Even for $k = 1$, where $G + I$ has 3 non-zero per rows (except the first and last row which have only 2), the optimal solution has less than 50% of these zeros, meaning 1.5 non-zeros on average per row for the optimal solution (which could potentially use all the n^2 entries of Δ).
- In terms of computational time, computing MH and D is extremely fast, as expected (the reason why the case $k = 1, 2$ is slightly slower for D is explained in Remark 4.2). Computing S is fast for sparse G , and becomes more expensive as k grows; see section 4.1.2 for an extensive numerical experiment on this case. The computational cost of $CG(\delta)$ grows as δ decreases, as expected, but remains close to that of S (remember, S is computed at the first step of Algorithm 1, that is, $S = CG(\delta)$ for any δ sufficiently large). The reason is that the computational cost of each iteration of the column-generation approach (Algorithm 1) is proportional to the computation of S : the number of columns added is roughly the same at each step. Finally, $CG(0)$ is significantly faster than GS , as expected, since GS does not leverage the column generation and solves the full problem at once, which requires forming a large constraint matrix. Note that, when G is dense ($k = 500$), S , $CG(\delta)$ and GS coincide since they all solve the full problem, on $\text{supp}(G + I)$, which is time consuming.

Remark 4.2 (Conditioning). *To compute $\Delta(\alpha^*)$, the stationary distribution μ of G is needed. For large n and small k , the condition number to calculate μ is very high; see [24] for a discussion on this issue. For example, for $n = 1000$ and $k = 1$, the `eigs` function of MATLAB runs into numerical problems and cannot return a feasible solution (it return NaN for the largest eigenvalue, and μ with negative entries). In this case, we resort to 100 iterations of the power method, initialized with $\mu = \mathbf{1}_n/n$, for simplicity.*

4.1.2 Scalability to compute S

Empirically, we have observed that the number of iterations of Algorithm 1 is relatively small, especially when δ is large; see Figure 4 for an experiment on real data sets. Moreover, we have observed that the computational cost of each iteration is similar, because we add to the support, Ω , roughly the same number of index pairs. Hence, the cost of computing $CG(\delta)$ is the cost of computing S times the number of iterations, which is typically small; for example, for $\delta = 10^{-2}$, it never exceeded 3 iterations. Hence it makes sense to analyse the computational cost of computing S , to get an idea how Algorithm 1 scales with n and k . Figure 1 reports the total time to compute S for various values of $k \in \{1, 2, 4, 8, 12, 16, 32, 64, 128\}$ and

$$n \in \{1000, 1669, 2783, 4642, 7743, 12916, 21545, 35939, 59949, 10^5, 2 \cdot 10^5\}$$

with log-spaced values¹⁰. To limit the computational time, we combine values of n and k only when $\text{nnz}(G) \approx 2nk \leq 450,000$. Above 450,000, the computational time exceeds 200 seconds.

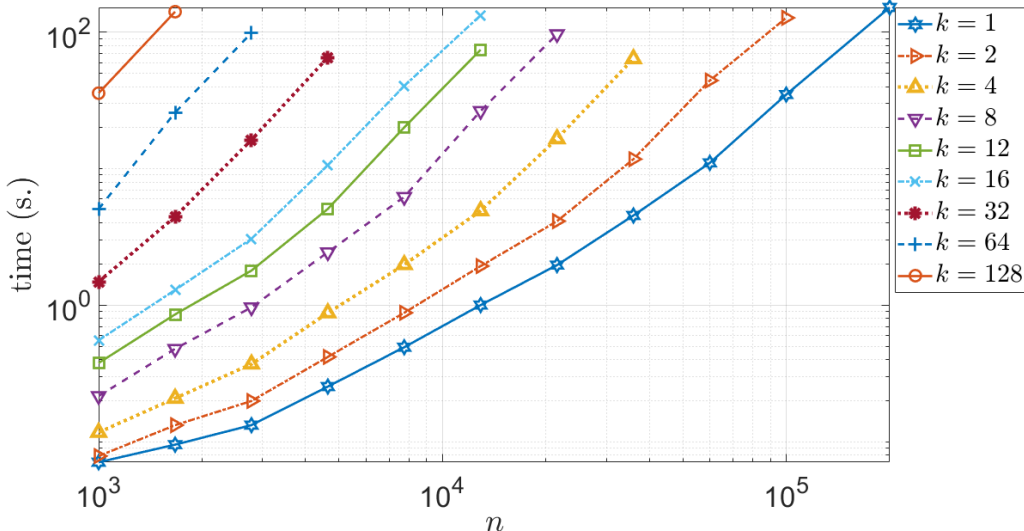


Figure 1: Average computational time to compute the solution S , that is, the solution of (11) with $\Omega = \text{supp}(G + I)$, for five queue-like transition matrices generated randomly (with about $2k$ non-zero entries per row) and $\hat{\mu} = G^\top \mathbf{1}_n/n$. This time accounts for the formulation of (11) and its resolution.

There is a dependence between the computational time and n that is somewhat linear, in a log-log plot. The average slope for computational times larger than 1 second is 2.2, meaning that the time depends roughly quadratically on n , for k fixed. Similarly, the average slope in a log-log plot between the computational time and k is 1.98, hence the time also depends quadratically on k , for n fixed. This allows us to solve large problems relatively fast, e.g., $n = 2 \cdot 10^5$ and $k = 1$ in about 150 seconds. It turns out that, quite surprisingly, the bottleneck of our algorithm is not to solve the large (sparse) linear optimization problems, but to form the constraint matrix A , of size¹¹ $2n \times (2\text{nnz}(G) + n)$, with two non-zeros per column of A . Constructing such a large sparse matrix does not require a linear time in the number of non-zeros (there are about $4\text{nnz}(G)$) but larger due to the access to memory, among other things.

Figure 2 reports the time to solve the linear optimization problems to obtain S . Comparing to Figure 1, we see that the time to solve the linear optimization problems is an order of magnitude smaller than formulating the problem. Possibly the construction of the sparse constraint matrix could be accelerated by using another language than MATLAB, e.g., C or C++.

In summary, the proposed approach to solve the ℓ_1 -TDSP with support constraint requires $O(\text{nnz}(G))$ memory and, empirically, about $O(\text{nnz}(G)^2)$ time. Figure 3 shows the time required to solve (11) with $\Omega = \text{supp}(G + I)$ as a function of $\text{nnz}(G) \approx 2nk$ for the synthetic data sets (these are the same values as in Figure 1 but presented differently), which follows a quadratic trend, as noted above, namely, $\text{time} \approx \zeta \text{nnz}(G)^2$ with $\zeta = 9 \cdot 10^{-10}$.

¹⁰We used `[ceil(logspace(3, 5, 10)), 2*1e5]` in MATLAB.

¹¹Recall the number of variables in the linear programs are $2\text{nnz}(G)$ for Δ^+ and Δ^- , and n for Δ^0 since the diagonal entries of our synthetic data sets G are zeros.

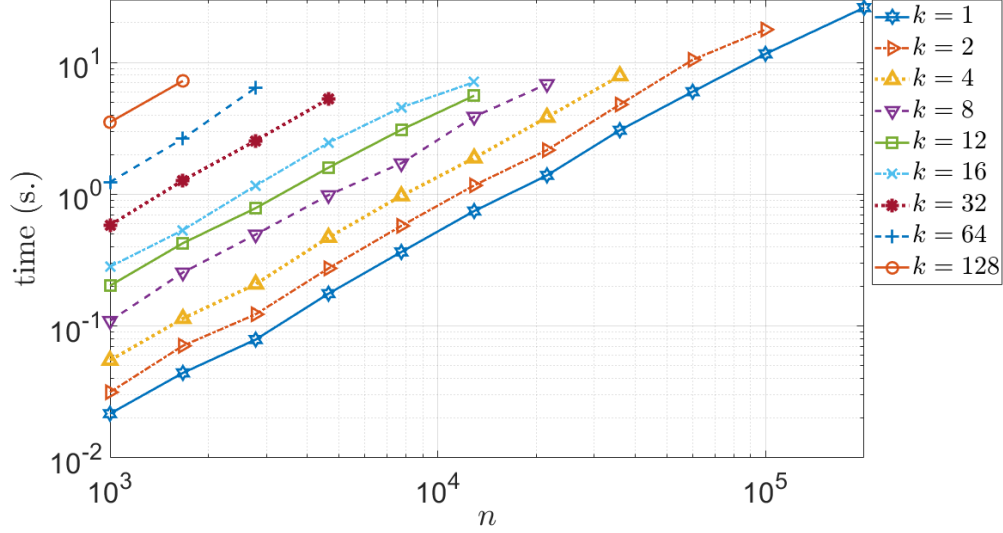


Figure 2: Average computational time to solve of (11) with $\Omega = \text{supp}(G + I)$, for 5 queue-like transition matrix generated randomly (with about $2k$ non-zero entries per row) and $\hat{\mu} = \mathbf{1}_n/n$. This time accounts only for the resolution (11), not the formulation.

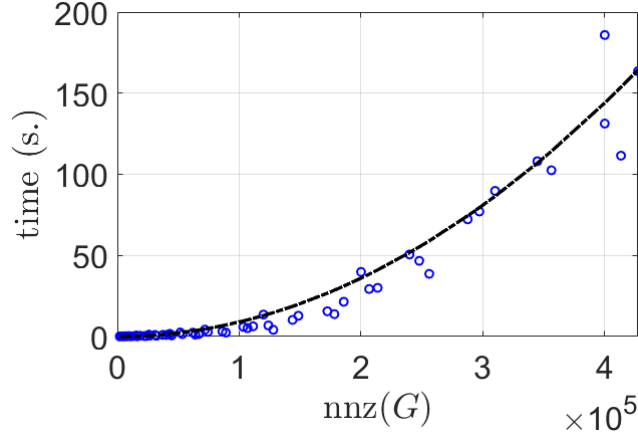


Figure 3: Average computational time to formulate and solve (11) with $\Omega = \text{supp}(G + I)$ for 5 queue-like transition matrix generated randomly (with about $2k$ non-zero entries per row) and $\hat{\mu} = \mathbf{1}_n/n$. The dashed curve is $\zeta \text{nnz}(G)^2$ with $\zeta = 9 \cdot 10^{-10}$. These are the same values as in Figure 1.

4.2 Real data sets

In this section, we use the same three sparse data sets as in [5]; see Table 2. They come from the KONECT project (<http://konect.cc/>). They represent the following:

- email: email-conversation network of university employees at the University of Rovira i Virgili¹².

¹²<http://konect.cc/networks/arenas-email/>

data sets	email	road	moreno
size (n)	1133	1030	2155
average number of non-zeros per row	9.62	2.51	5.32

Table 2: Sparse real data sets from [5].

Nodes are employees and each undirected edge represents that at least one email was sent between the employees. The support of G is symmetric, but G is not symmetric after the normalization required to make G stochastic.

- road: road network between the largest cities in Europe¹³. As for the previous data set, the support of G is symmetric, but G is not symmetric.
- moreno: high-school network of student relationships from a survey from 1994/1995 on a high school where each student had to indicate his/her 5 best female and male friends¹⁴. This is a directed network, that is, the support of G and G are not symmetric.

We use $\hat{\mu} = (1 - \epsilon)\mu + \epsilon\frac{1n}{n}$ for $\epsilon \in \{0.01, 0.1, 0.5\}$. This means that we are trying to balance the importance of the nodes in the chain, e.g., for the road network, to make the roads more evenly congested; see Remark 4.1. Table 3 reports the results for $\epsilon = 0.01$, Table 4 for $\epsilon = 0.1$, and Table 5 for $\epsilon = 0.5$.

		MH	D	S	$CG(10^{-2})$	$CG(10^{-4})$	$CG(0)$	GS
email	obj (%)	3.82	5.02	0.42	0.20	0.17	0.17	0.17
	spars (%)	53.44	99.40	11.77	10.64	10.26	10.25	10.25
	time (s.)	0.03	0.03	0.15	0.32	0.86	1.15	36.31
road	obj (%)	0.72	1.92	0.41	0.36	0.31	0.28	0.28
	spars (%)	45.96	99.70	35.74	35.96	33.57	31.52	31.16
	time (s.)	0.02	0.02	0.03	0.08	0.25	1.26	27.25
moreno	obj (%)	116.54	21.73	4.90	0.98	0.79	0.78	0.78
	spars (%)	81.67	99.93	24.42	21.72	20.14	20.03	20.03
	time (s.)	0.11	0.09	0.21	0.94	3.98	6.41	257.86

Table 3: Experiments on real data with $\epsilon = 0.01$.

We observe the following:

- In terms of objective function values, D is significantly worse than GS , especially when ϵ is large: the objective function values of D gets smaller as ϵ gets smaller, that is, as $\|\mu - \hat{\mu}\|_2$ gets smaller, as expected; see (8). MH provides good solutions for the road data set. An explanation is that the support of G is symmetric, as it is a road network. For the moreno data set, whose support is asymmetric (43.55% of non-zeros of G are not in G^T), it performs poorly. For the email data set, whose support is symmetric as well, it performs better than D but is far from optimality. S provides a solution with objective relatively close to that of GS , except for the moreno data set where it is significantly worse (e.g., for $\epsilon = 0.1$, obj of S is 18.29% while it is 5.60% for GS).

¹³http://konect.cc/networks/subelj_euroroad/

¹⁴http://konect.cc/networks/moreno_highschool/

		MH	D	S	$CG(10^{-2})$	$CG(10^{-4})$	$CG(0)$	GS
email	obj (%)	26.68	36.92	4.23	2.13	2.13	2.13	2.13
	spars (%)	53.44	99.40	12.94	12.84	12.83	12.84	12.83
	time (s.)	0.03	0.03	0.12	0.48	0.68	0.89	37.05
road	obj (%)	6.67	18.39	4.01	3.60	2.86	2.86	2.86
	spars (%)	45.96	99.70	35.96	35.96	31.49	31.43	31.16
	time (s.)	0.02	0.02	0.03	0.08	0.99	1.85	27.33
moreno	obj (%)	116.54	69.92	18.29	6.40	5.61	5.60	5.60
	spars (%)	81.67	99.93	29.88	22.94	21.52	21.51	21.51
	time (s.)	0.11	0.09	0.21	1.43	5.73	7.20	255.08

Table 4: Experiments on real data with $\epsilon = 0.1$.

		MH	D	S	$CG(10^{-2})$	$CG(10^{-4})$	$CG(0)$	GS
email	obj (%)	67.65	107.47	28.22	24.93	24.93	24.93	24.93
	spars (%)	53.44	99.40	33.64	35.50	35.52	35.52	35.51
	time (s.)	0.03	0.03	0.25	0.88	1.16	1.16	38.22
road	obj (%)	26.57	80.85	19.73	18.78	16.82	16.82	16.82
	spars (%)	45.96	99.70	40.70	40.45	38.45	38.23	37.96
	time (s.)	0.02	0.02	0.04	0.08	0.75	1.18	27.31
moreno	obj (%)	117.75	135.08	39.00	21.84	21.45	21.45	21.45
	spars (%)	81.67	99.93	40.08	29.97	29.34	29.34	29.33
	time (s.)	0.11	0.10	0.24	2.32	5.51	5.51	253.31

Table 5: Experiments on real data with $\epsilon = 0.5$.

All $CG(\delta)$ variants perform similarly with slight improvements when δ decreases. As expected, the objectives of $CG(0)$ and GS coincide.

- In terms of computational cost, MH and D are again extremely fast. $CG(0)$ outperforms GS , as for synthetic data sets since G is sparse. However, $CG(0)$ sometimes requires more time than $CG(10^{-2})$ and $CG(10^{-4})$ for a negligible improvement in the objective (e.g., for the moreno data set with $\epsilon = 0.01$, from 3.98 seconds to 6.41 seconds to reduce the relative objective from 0.79% to 0.78%). This is because $CG(0)$ can only stop when it has found an optimal solution. The solution S is computed extremely fast (less than 0.25 seconds in all cases) but sometimes produces high objective function values. Hence, in practice, using CG with a value of $\delta \in [10^{-2}, 10^{-4}]$ seems to be a good compromise between the quality of the solution and the run time. Figure 4 shows the evolution of the objective function values between the iterations of Algorithm 1 (that computes CG). This confirms that a few iterations of Algorithm 1 provides good solutions.
- In terms of sparsity, D has essentially the same sparsity of $G + I$, except for one row; this is expected, see Lemma 2.5. MH has a sparsity close to 50% for the symmetric data sets, as for the synthetic data sets, and around 80% for the asymmetric one, moreno. The solution S is significantly sparser than $G + I$, between 3 and 10 times. The global solution (GS and $CG(0)$) generates surprisingly sparse solutions, slightly sparser than S . In quite a few cases, the globally optimal solutions generated by GS and $CG(0)$ do not have the same sparsity, meaning that the solution of (11) is not unique.

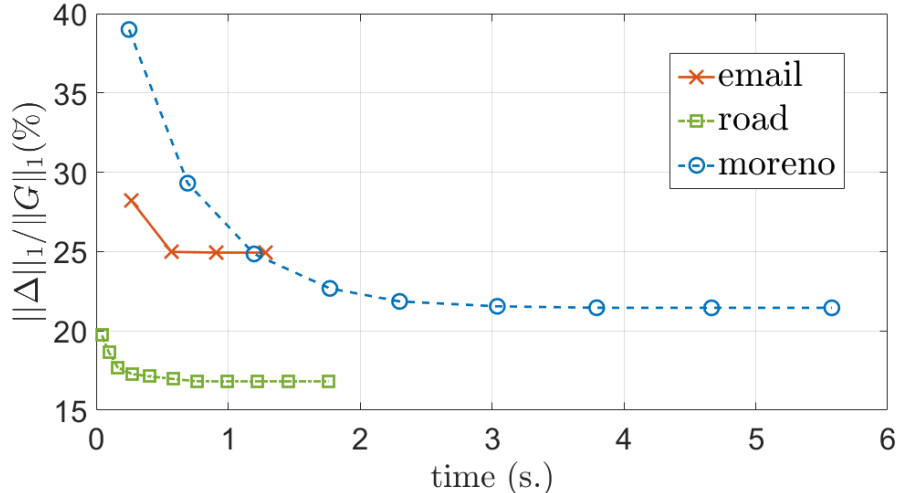


Figure 4: Evolution of the relative objective function between each iteration of the column-generation approach (Algorithm 1) for the three real data sets, for $\epsilon = 0.5$. (Note that the timings are not exactly the same as in Table 5, this is because we regenerated the solutions to make this plot.)

5 Concluding remarks

In this paper we proposed several algorithms for assigning a target stationary distribution $\hat{\mu}$ to a perturbed stochastic matrix $\hat{G} = G + \Delta$, with a constraint on the support of Δ . We first analyzed the special case where $\Delta := \text{Diag}(\alpha)(I_n - G)$ whose support is restricted to the union of the supports of G and the identity, which implies that it does not destroy the sparsity of the original matrix G . We proved several properties of this solution: its optimality for that class of perturbations, its sparsity, when it is of rank-one, and when its norm is minimized depending the ordering of $\hat{\mu}$ compared to μ . Unfortunately, numerical experiments show that this proposed solution is in most cases quite far from global optimality, because the feasible set of perturbations is too constrained, while not being very sparse as its support essentially coincides with that of $G + I$. It works well however when $\|\mu - \hat{\mu}\|$ is small enough (see Equation (7), and Table 1 when $k = 500$), and is optimal when $\hat{\mu} = \frac{1}{1+\lambda}(\mu + \lambda e_j)$ for any j and λ that is sufficiently large (Theorem 2.8). We then proposed an effective linear optimization formulation of the problem when minimizing the component-wise ℓ_1 norm of Δ which promotes sparse solutions, as proved in Theorem 3.1. To solve this linear optimization problem efficiently, we designed a dedicated column generation approach; see Algorithm 1 which can be stopped before global optimality to improve the trade off between solution quality and run time. Algorithm 1 allows us to solve large sparse problems, with and without support constraint and up to global optimality, extremely fast, for sparse matrices up to size $n = 200,000$ in a few minutes. This is because empirically the main computational cost of the method is to construct a large sparse matrix of dimension $2n \times O(\text{nnz}(G))$ with two non-zeros per column.

A limitation of our column generation approach is that it relies on the support of $G + I$ to provide a first feasible solution. If G is dense, this will make the algorithm not scale as well. However, even for dense matrices, the optimal solutions are (often) sparse; see, e.g., the last rows of Table 1. Hence it would be interesting to address the following problem: Given G and $\hat{\mu}$, provide a sparse set Ω so that the TSDP with support constraints (11) is feasible. This would not only allow us to initialize

Algorithm 1 more efficiently, but also provide sparse solutions to the TSDP. It could be useful even when G is sparse by initializing Ω with a support sparser than $G + I$, leading to computational gains. We conjecture that Ω needs, generically, at least $2n$ elements : one per column to satisfy $\hat{\boldsymbol{\mu}}^\top \Delta = \hat{\boldsymbol{\mu}}^\top (I - G)$, and 2 per row to satisfy $\Delta \mathbf{1}_n = 0$.

Acknowledgment

We thank Sam Power (University of Bristol) for letting us know of the use of the Metropolis-Hastings algorithm in the context of the TSDP, and for helpful discussions.

We are also very grateful to the two anonymous reviewers who carefully read the manuscript; their feedback allowed us to improve it significantly.

A Proof of Theorem 2.10

Lemma A.1. *Let $0 < a_- \leq a_+$ and $0 < b_- \leq b_+$, then*

$$\frac{a_-}{b_+} \leq \min \left(\frac{a_-}{b_-}, \frac{a_+}{b_+} \right) \leq \max \left(\frac{a_-}{b_-}, \frac{a_+}{b_+} \right) \leq \frac{a_+}{b_-},$$

which implies that $\left[\min(\frac{a_-}{b_-}, \frac{a_+}{b_+}), \max(\frac{a_-}{b_-}, \frac{a_+}{b_+}) \right] \subset \left[\frac{a_-}{b_+}, \frac{a_+}{b_-} \right]$.

This inclusion is strict if and only if $a_- < a_+$ and $b_- < b_+$.

Proof. It follows from the assumptions that

$$a_- b_- \leq a_- b_+, a_+ b_- \leq a_+ b_+.$$

Dividing all quantities by the positive product $b_- b_+$ yields

$$\frac{a_-}{b_+} \leq \frac{a_-}{b_-}, \frac{a_+}{b_+} \leq \frac{a_+}{b_-},$$

from which the inclusion result follows. It is easy to see that if either $a_- = a_+$ or $b_- = b_+$, then both intervals are equal, and that if $a_- < a_+$ and $b_- < b_+$, then the inclusion of the intervals is strict. \square

Lemma A.2. *Let the elements of the vectors $\boldsymbol{\mu}$, $\hat{\boldsymbol{\mu}}$ and $\mathbf{r} := \boldsymbol{\mu} / \hat{\boldsymbol{\mu}}$ be positive and assume those of $\hat{\boldsymbol{\mu}}$ are ordered in a non-decreasing way, that is, $0 < \hat{\mu}_1 \leq \hat{\mu}_2 \leq \dots \hat{\mu}_n$. Let the elements of the permuted vector $\tilde{\boldsymbol{\mu}} := \tilde{P} \boldsymbol{\mu}$ also be ordered in a non-decreasing way, that is, $0 < \tilde{\mu}_1 \leq \tilde{\mu}_2 \leq \dots \tilde{\mu}_n$. Then*

$$\left[\min_i \tilde{\mu}_i / \hat{\mu}_i, \max_i \tilde{\mu}_i / \hat{\mu}_i \right] \subset \left[\min_i \mu_i / \hat{\mu}_i, \max_i \mu_i / \hat{\mu}_i \right]. \quad (14)$$

Proof. The permutation \tilde{P} that reorders the elements of the vector $\boldsymbol{\mu}$ to $\tilde{\boldsymbol{\mu}} := \tilde{P} \boldsymbol{\mu}$, can be factored in a sequence of reorderings of just two elements that are not ordered in a non-decreasing way. At each such step, the interval (14) in which the new ratios lie, can only decrease because of Lemma A.1. \square

B Rank-one solution for the component-wise ℓ_1 norm

Let us show that there exists an optimal rank-one solution to the TSDP for the component-wise ℓ_1 norm without the nonnegativity constraints. This can be done by proving that there exists a rank-one optimal solution to the problem

$$\min_{\Delta} \|\Delta\|_1 \quad \text{such that} \quad \hat{\boldsymbol{\mu}}^\top \Delta = \hat{\boldsymbol{\mu}}^\top (I - G), \quad (15)$$

which satisfies the sum-to-one condition, that is, $(G + \Delta)\mathbf{1}_n = \mathbf{1}_n$.

Let us denote $\mathbf{b} := (I - G)^\top \hat{\boldsymbol{\mu}}$. The problem (15) is separable by columns of Δ , that is, the columns of Δ can be optimized independently. The subproblem of (15) for the k th column of Δ is

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{such that} \quad \hat{\boldsymbol{\mu}}^\top \mathbf{z} = b_k. \quad (16)$$

We need to find the \mathbf{z} with smallest ℓ_1 norm such that $\hat{\boldsymbol{\mu}}^\top \mathbf{z} = b_k$. It is easy to convince oneself that \mathbf{z} with a single non-zero entry corresponding to the largest entry in $\hat{\boldsymbol{\mu}}$ is optimal. An optimal solution of (16) is therefore given by $\mathbf{z}^* = \frac{b_k}{\hat{\mu}_i} \mathbf{e}_i$ for $i \in \operatorname{argmax}_k \hat{\mu}_k$. Therefore,

$$\Delta^* = \frac{1}{\hat{\mu}_i} \mathbf{e}_i \mathbf{b}^\top = \frac{1}{\hat{\mu}_i} \mathbf{e}_i \hat{\boldsymbol{\mu}}^\top (I - G),$$

is a rank-one optimal solution to (15). Moreover, $G + \Delta^*$ respects the sum-to-one condition, as $\Delta^* \mathbf{1}_n = 0$, which concludes the proof.

References

- [1] Abbas, K., Berkhout, J., Heidergott, B.: A critical account of perturbation analysis of Markov chains. *Markov Processes and Related Fields* **22**, 227–266 (2016)
- [2] Avrachenkov, K., Borkar, V., Nemirovsky, D.: Quasi-stationary distributions as centrality measures for the giant strongly connected component of a reducible graph. *Journal of Computational and Applied Mathematics* **234**(11), 3075–3090 (2010)
- [3] Bach, F., Jenatton, R., Mairal, J., Obozinski, G., et al.: Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning* **4**(1), 1–106 (2012)
- [4] Ballard, G., Kolda, T.G., Pinar, A., Seshadhri, C.: Diamond sampling for approximate maximum all-pairs dot-product (MAD) search. In: *IEEE International Conference on Data Mining*, pp. 11–20 (2015)
- [5] Berkhout, J., Heidergott, B., Van Dooren, P.: Perturbation and inverse problems of stochastic matrices. *SIAM Journal on Matrix Analysis and Applications* **45**(1), 553–584 (2024)
- [6] Berman, A., Plemmons, R.J.: *Nonnegative matrices in the mathematical sciences*. SIAM (1994)
- [7] Billera, L.J., Diaconis, P.: A geometric interpretation of the Metropolis-Hastings algorithm. *Statistical Science* pp. 335–339 (2001)

- [8] Bolch, G., Greiner, S., De Meer, H., Trivedi, K.S.: Queueing networks and Markov chains: modeling and performance evaluation with computer science applications. John Wiley & Sons (2006)
- [9] Boyd, S., Diaconis, P., Xiao, L.: Fastest mixing Markov chain on a graph. *SIAM Review* **46**(4), 667–689 (2004)
- [10] Boyd, S.P., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
- [11] van den Brand, J.: A deterministic linear program solver in current matrix multiplication time. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 259–278 (2020)
- [12] Breen, J., Kirkland, S.: Stationary vectors of stochastic matrices subject to combinatorial constraints. *The Electronic Journal of Linear Algebra* **28**, 25–53 (2015)
- [13] Brualdi, R.A.: Convex sets of non-negative matrices. *Canadian Journal of Mathematics* **20**, 144–157 (1968)
- [14] Caswell, H.: Sensitivity analysis of discrete Markov chains via matrix calculus. *Linear Algebra and its Applications* **438**, 1727–1745 (2013)
- [15] Cho, G., Meyer, C.: Comparison of perturbation bounds for the stationary distribution of a Markov chain. *Linear Algebra and its Applications* **335**, 137–150 (2001)
- [16] Crisostomi, E., Kirkland, S., Shorten, R.: A Google-like model of road network dynamics and its application to regulation and control. *International Journal of Control* **84**(3), 633–651 (2011)
- [17] Diaconis, P., Saloff-Coste, L.: What do we know about the Metropolis algorithm? In: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, pp. 112–129 (1995)
- [18] Gleich, D.F.: Pagerank beyond the web. *SIAM Review* **57**(3), 321–363 (2015)
- [19] Gregory, D., Kirkland, S., Pullman, N.: Row-stochastic matrices with a common left fixed vector. *Linear Algebra and its Applications* **169**, 131–149 (1992)
- [20] Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1), 97–109 (1970)
- [21] Higham, N.J., Relton, S.D.: Estimating the largest elements of a matrix. *SIAM Journal on Scientific Computing* **38**(5), C584–C601 (2016)
- [22] Ipsen, I., Selee, T.: Ergodicity coefficients defined by vector norms. *SIAM Journal on Matrix Analysis and Applications* **32**, 153–200 (2011)
- [23] Kirkland, S.: On a question concerning condition numbers for Markov chains. *SIAM Journal on Matrix Analysis and Applications* **23**, 1109–1119 (2002)
- [24] Kirkland, S.: Column sums and the conditioning of the stationary distribution for a stochastic matrix. *Operators and Matrices* **4**, 431–443 (2010)

- [25] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21**(6), 1087–1092 (1953)
- [26] Meyer, C.: The condition of a finite Markov chain and perturbation bounds for the limiting probabilities. *SIAM Journal of Algebraic Discrete Methods* **1**, 273–283 (1980)
- [27] Mitrophanov, A.Y.: Sensitivity and convergence of uniformly ergodic Markov chains. *Journal of Applied Probability* **42**(4), 1003–1014 (2005)
- [28] Nemhauser, G.L.: Column generation for linear and integer programming. *Optimization Stories* **20**(65), U73 (2012)
- [29] Seneta, E.: Sensitivity analysis, ergodicity coefficients, and rank-one updates of finite Markov chains. In: *Numerical Solutions of Markov Chains*, Marcel Dekker, New York (2001). Stewart, WJ (ed.)
- [30] Wasserman, S., Faust, K.: *Social network analysis: Methods and applications* (1994)