OPTIMAL FUZZY AGGREGATION OF NETWORKS

MARCO SARICH*, CHRISTOF SCHÜTTE*, AND ERIC VANDEN-EIJNDEN†

ABSTRACT. This paper is concerned with the problem of fuzzy aggregation of a network with non-negative weights on its edges into a small number of clusters. Specifically we want to optimally define a probability of affiliation of each of the n nodes of the network to each of $m \ll n$ clusters or aggregates. We take a dynamical perspective on this problem by analyzing the discrete-time Markov chain associated with the network and mapping it onto a Markov chain describing transitions between the clusters. We show that every such aggregated Markov chain and affiliation function can be lifted again onto the full network to define the so-called *lifted* transition matrix between the nodes of the network. The optimal aggregated Markov chain and affiliation function can then be determined by minimizing some appropriately defined distance between the lifted transition matrix and the transition matrix of the original chain. In general, the resulting constrained nonlinear minimization problem comes out to have continuous level sets of minimizers. We exploit this fact to devise an algorithm for identification of the optimal cluster number by choosing specific minimizers from the level sets. Numerical minimization is performed by some appropriately adapted version of restricted line search using projected gradient descent. The resulting algorithmic scheme is shown to perform well on several test examples.

1. INTRODUCTION

Recent advances in science and technology as well as events in our society have brought new challenges to the field of network science. The rapid growth and wide usage of the World Wide Web (WWW) has offered one of the most important and fascinating examples of a network whose structure and complexity has gone far beyond the examples studied before in the classical computer science literature. Networks have also become popular in the social sciences to represent and analyze the interactions between individual or communities. On another front, cell biology has also evolved to a stage where elementary biochemical reactions of many intra-cellular processes are understood well enough and their overall structure is often expressed in the form of networks. Yet another example of application is provided by molecular science where networks have become popular recently to analyze the enormous amount of data one can nowadays generate by molecular dynamical simulations.

In all of these applications, the networks are typically very large and very complex. In order to understand the structure and function of these networks a common strategy is to partition them into smaller networks which are simpler yet retain some basic properties of the original ones. One possible approach to partitioning is to lump or aggregate

Key words and phrases. fuzzy aggregation, clustering, partitioning, network, random walk, Markov chain, affiliation function, constrained nonlinear minimization problem, numerical minimization, restricted line search.

*Institut für Mathematik und Informatik, Freie Universitt Berlin, Arnimallee 2-6, 14195 Berlin, Germany (sarich@math.fu-berlin.de, schuette@math.fu-berlin.de). These authors' research was supported by the DFG Research Center MATHEON "Mathematics for Key Technologies" in Berlin.

†Courant Institute of Mathematical Sciences, New York Univerity, New York, NY 10012 (eve2@cims.nyu.edu). This authors' research was partially supported by NSF grants DMS02-09959,DMS02-39625 and DMS07-08140, and ONR grant N00014-04-1-0565.

the nodes in the networks into clusters (or communities, aggregates, or dominant conformations depending on the application) such that the affiliation of each node to a cluster is deterministic. The approach with deterministic affiliations is rather popular and many variants have been widely discussed in the literature, e.g. k-means, spectral decomposition methods and variants thereof [3, 4, 5, 6, 15, 16, 17, 18], methods based on modularity [7], etc.

All approaches with deterministic affiliations have one essential drawback, however. In most real-world scenarios there are nodes that do not belong to one of the communities but bear affiliations with several communities. In these cases, it is better to use partitioning approaches in which the affiliation of a node to a given network is probabilistic rather than deterministic. Such approaches are usually referred to as *fuzzy* partitioning techniques and the literature contains many different types of these techniques: Gaussian mixture models [8, 9] or other Bayesian network models [10], fuzzy k-means and other clustering algorithms [11] and fuzzy variants of spectral decomposition techniques [12, 13].

The different approaches to partitioning, either deterministic or fuzzy, differ by the criterion they use to measure the quality of the clustered network. Indeed, one of the main issue in partitioning is to define what is meant by best in this context. In this paper we will adopt the strategy proposed in Ref. [1] which exploits the well-know isomorphism between networks and Markov chains. If Ω denotes the set of nodes in the network and we assume that the weight $w(x, y)$ between any two nodes $x \in \Omega$ and $y \in \Omega$ is non-negative, $w(x, y) \geq 0$, then one can associate with the network a discrete-time Markov chain with stochastic matrix P with entries

$$P(x, y) = \frac{w(x, y)}{d(x)} \qquad d(x) = \sum_{y \in \Omega} w(x, y) \tag{1}$$

where $d(x)$ is the (out) degree of nodes x. The partitioning strategy proposed in Ref. [1] amounts to (i) constructing a simplified stochastic matrix $\tilde{P}$ in which the probability to hop from a node in one cluster to one in another is uniform in the starting cluster, and (ii) optimizing the definition of the clusters in such a way that the dynamical properties of the simplified chain with stochastic matrix $\tilde{P}$ remains as close as possible to the ones of the original chain with stochastic matrix P. Here close is understood in the sense of some appropriately weighted Frobenius norm. In Ref. [1], this approach was used in the context of deterministic partitioning. Here we generalize the technique by allowing the assignment to the clusters to be fuzzy. Since stochastic constraints apply, the resulting optimization problem is a *non-convex minimization problem with mixed equality and inequality constraints*. We show how to solve this problem by adapting standard algorithmic strategies to the special structure at hand and analyze the complexity of the algorithm. We also show how the new strategy permits to identify the optimal number of clusters to be used. The optimal coarse graining is identified by the minimum of the weighted Frobenius functional.

With respect to other partitioning strategies, the one we propose has the advantage that it explicitly takes into account and is tailored to preserve the dynamical properties of the Markov chain associated with the network. This is especially suitable in applications where the network is indeed the stage of some dynamics, like in the examples of biological networks or the networks used to analyze the time-series data obtained from molecular dynamics. In this context, nodes that have a high probability to be assigned to one particular cluster are nodes that are strongly linked together kinetically, and form core clusters inside the clusters in which the chain spend a significant amount of time. In contrast, the nodes

that have a significant probability to be assigned to more than one cluster can be viewed as part of the transition regions between these core cluster.

The remainder of this paper is organized as follows. In Sec.2, we formulate the constrained minimization problem and discuss the underlying probabilistic interpretation. Then, we show that there is no unique minimum and discuss how to choose amongst the minima. Based on this, Sec. 3 describes our strategy for the identification of the number of clusters. Then, Sec. 4 introduces the algorithmic considerations, i.e., the solver for the constrained minimization problem and finally, in Sec. 5 we will discuss some numerical experiments including a network with hierarchical structure and an application to a transition network that results from diffusion in a potential energy landscape.

2. Deterministic and Fuzzy Clustering via Constrained Minimization

As explained in the introduction, any network (i.e. a weighted directed graph) with positive weights on its edges is isomorphic with a Markov chain, see (1). From now on we will therefore focus this Markov chain. More specifically, we will consider a discrete-time Markov chain with finite state space $\Omega = \{1, \dots, n\}$. We denote the transition matrix of this chain by P and we assume that it has unique invariant measure μ satisfying $\mu^T P = \mu^T$. We also assume that the entries of μ are all positive, $\mu(x) > 0$ for all $x \in \Omega$.

Our aim is to partition the state of this chain into a set of $m \leq n$ clusters, which we shall denote by $\hat{\Omega} = \{1, \dots, m\}$. To do this, consistent with the general strategy proposed in Ref. [1] we will proceed as follows. Assuming that the m clusters are given (deterministically or stochastically), we will introduce a discrete-time Markov chain on the state space $\hat{\Omega}$ by specifying a $m \times m$ stochastic matrix $\hat{P}$. Below we will denote by $\hat{\mu}$ the invariant measure of this chain, i.e. $\hat{\mu}^T \hat{P} = \hat{\mu}^T$. We will then define in some appropriate way the lift of this stochastic matrix in the original state space Ω, which is a certain $n \times n$ stochastic matrix $\tilde{P}$, and measure the distance between this $\tilde{P}$ and the original P using some suitable norm. Finally, we will propose to minimize this distance to obtain the optimal set of clusters, the optimal $\hat{P}$ on these clusters, and the optimal lift $\tilde{P}$. Note that this strategy not only gives a partitioning of the original network; it also give a coarse-graining of the dynamics specified by the stochastic matrix P isomorphic to this network.

2.1. Deterministic clustering.

A deterministic clustering can be specified by a map $C : \Omega \to \hat{\Omega}$ which assigns every node in Ω to one and only one clusters in $\hat{\Omega}$, and the associated affiliation function $W_d : \Omega \times \hat{\Omega} \to \{0, 1\}$ defined as

$$W_d(x, i) = \begin{cases} 1 & \text{if } C(x) = i \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Taken together, a map C and a $m \times m$ stochastic matrix $\hat{P}$ specifying a Markov chain on these clusters naturally induce a dynamics on the original state space Ω. This dynamics is defined by following 3-step rule:

(1) Given that the system is in state x, let $i = C(x)$.
(2) Pick $j \in \hat{\Omega}$ according to the probability $\hat{p}(i, \cdot)$.
(3) Pick $y \in \Omega$ according to the probability $\mu_j(\cdot)$ where $\mu_j(x)$ is the equilibrium probability distribution of the original chain conditional on $C(x) = j$

$$\mu_i(x) = \frac{\mu(x) W_d(x, i)}{\sum_{y=1}^{n} \mu(y) W_d(y, i)} \tag{3}$$

This 3-step rule can be graphically explained as

$$\begin{array}{ccc} x \in \Omega & \xrightarrow{\tilde{P}(x,\cdot)} & y \in \Omega \\ {\scriptstyle i=C(x)}\downarrow & & \uparrow{\scriptstyle \mu_j(\cdot)} \\ i \in \hat{\Omega} & \xrightarrow[\hat{P}(i,\cdot)]{} & j \in \hat{\Omega} \end{array} \tag{4}$$

and it is easy to see that it defines a discrete-time Markov chain on Ω with stochastic matrix (the upper arrow in the graph above):

$$\tilde{P}(x,y) = \hat{P}(C(x), C(y))\mu_j(y). \tag{5}$$

It is easy to see that $\tilde{P}$ and P will share the same equilibrium distribution $\hat{\mu}$ if the equilibrium distribution of $\hat{P}$ satisfies

$$\hat{\mu}(j) = \sum_{x=1}^{n} W_d(x,j)\mu(x), \tag{6}$$

Assuming that this constraint is satisfied, (5) can also be written as

$$\tilde{P}(x,y) = \sum_{i,j=1}^{m} W_d(x,i)\hat{P}(i,j)W_d(y,j)\frac{\mu(y)}{\hat{\mu}(j)}. \tag{7}$$

This matrix can be thought of as the lift in Ω of the stochastic matrix $\hat{P}$, an operation that we shall denote as $\tilde{P} = L_{W_d}(\hat{P})$. In matrix notation we have

$$L_{W_d}(\hat{P}) := \tilde{P} = W\hat{P}D_{\hat{\mu}}^{-1}W^T D_\mu, \tag{8}$$

with $D_\mu = \mathrm{diag}\{\mu_1, ..., \mu_n\}$ and $D_{\hat{\mu}} = \mathrm{diag}\{\hat{\mu}_1, ..., \hat{\mu}_m\}$.
The partitioning strategy proposed in Ref. [1] amounts to finding the $\tilde{P} = L_{W_d}(\hat{P})$ which is the closest to P, where closeness is measured in terms of the weighted Frobenius norm of the difference between P and $L_{W_d}(\hat{P})$:

$$E(\hat{P}, W_d) = \sum_{x,y=1}^{n} \frac{\mu(x)}{\mu(y)} \Big(P(x,y) - L_{W_d}(\hat{P})(x,y)\Big)^2 \tag{9}$$

The choice of the functional is motivated in the Appendix. This functional is to be minimized over W_d and $\hat{P}$ subject to the constraint (6) as well as

$$\begin{aligned} &\forall i,j \in \hat{\Omega} \ : \ \hat{P}(i,j) \geq 0, \\ &\forall i \in \hat{\Omega} \ : \ \sum_{j=1}^{m} \hat{P}(i,j) = 1 \end{aligned} \tag{10}$$

and

$$\forall x \in \Omega, i \in \hat{\Omega} \ : W_d(x,i) \in \{0,1\}. \tag{11}$$

If we keep W_d fixed and minimize (9) over $\hat{P}$ subject to (6) and (10) assuming that (11) holds, a straightforward calculations shows that the unique minimizer is

$$\hat{P}(i,j) = \sum_{x,y=1}^{n} \frac{\mu(x)}{\hat{\mu}(i)} W_d(x,i)P(x,y)W_d(y,j). \tag{12}$$

We will refer to this stochastic matrix as the inverse lift of P and denote it as $\hat{L}_{W_d}(P)$. In matrix notation, it is

$$\hat{L}_{W_d}(P) = D_{\hat{\mu}}^{-1} W_d^T D_\mu P W_d. \tag{13}$$

Note that the inverse lift of the lift is the identity, $\hat{L}_{W_d}(L_{W_d}(\hat{P})) = \hat{P}$, but the lift of the inverse lift is not, $L_{W_d}(\hat{L}_{W_d}(P)) = \tilde{P} \neq P$ in general.

Inserting (13) in (9) leaves us with an objective function for W_d alone

$$E(W_d) := E(\hat{L}_{W_d}(P), W_d) = \sum_{x,y=1}^{n} \frac{\mu(x)}{\mu(y)} \Big(P(x,y) - L_{W_d}(\hat{L}_{W_d}(P))(x,y) \Big)^2 \tag{14}$$

We can then minimize this objective function subject to the constraint in (11) to obtain the best deterministic partition W_d. An algorithm to perform this minimization was proposed in Ref. [1].

2.2. **Stochastic interpretation of fuzzy clustering.** Now we generalize the approach presented in Sec. 2.1 to stochastic or fuzzy clustering. The basic idea is to make the map $C : \Omega \to \hat{\Omega}$ random rather than deterministic. Specifically, we assume that entries $C(x)$ of the map are statistically independent from one another, and that the probability that $C(x) = i$ for some $i \in \hat{\Omega}$ is given by

$$\mathbb{P}[C(x) = i] = W(x, i). \tag{15}$$

where $W : \Omega \times \hat{\Omega} \to [0, 1]$ is a fuzzy affiliation function. $W(x, i)$ can be interpreted as the probability that the state $x \in \Omega$ is affiliated with the cluster state i, i.e. $W(x, \cdot)$ gives a probability distribution on the clustered space $\hat{\Omega}$. Obviously, this requires that

$$\begin{aligned} &\forall x \in \Omega, i \in \hat{\Omega} \ : \ W(x, i) \geq 0 \\ &\forall x \in \Omega \ : \ \sum_{i=1}^{m} W(x, i) = 1. \end{aligned} \tag{16}$$

Thus, one can think of a stochastic or fuzzy clustering as an ensemble of deterministic ones distributed according to the probability distribution given by W on the set of all deterministic clusterings. Therefore the natural extension of the lift given by (7) is achieved by taking the ensemble average, i.e. the expectation value with respect to the distribution given by the affiliation function W on the set of all possible deterministic clusterings C. Let $\mathcal{C}$ denote this set. Then we have

$$\tilde{P}(x, y) = \mathbb{E}\left[\hat{P}(C(x), C(y)) \frac{\mu(y)}{\hat{\mu}(C(y))} \right] = \sum_{C \in \mathcal{C}} \mathbb{P}[C] \hat{P}(C(x), C(y)) \frac{\mu(y)}{\hat{\mu}(C(y))} \tag{17}$$

Reorganizing the sum by considering all clustering that have the same value on x and y and using statistical independence as well as (15), $\tilde{P}(x, y)$ can be re-expressed as

$$\begin{aligned} \tilde{P}(x, y) &= \sum_{i,j} \mathbb{P}[C(x) = i] \hat{P}(i, j) \mathbb{P}[C(y) = j] \frac{\mu(y)}{\hat{\mu}(j)} \\ &= \sum_{i,j=1}^{m} W(x, i) \hat{P}(i, j) W(y, j) \frac{\mu(y)}{\hat{\mu}(j)}. \end{aligned} \tag{18}$$

Thus, the lift $L_W(\hat{P})$ of a clustered stochastic matrix can be defined in the fuzzy clustering setting as in the deterministic setting, by (7) except that the deterministic affiliation

function $W_d(x,i)$ in (2) has now be replaced by a fuzzy affiliation function $W(x,i)$ satisfying (16).

The dynamics induced by the stochastic matrix $\tilde{P}(x,y)$ is the following generalization of the 3-step rule dynamics defined in Sec. 2.1:

(1) Given that the system is in state x, pick i with probability $W(x,\cdot)$.
(2) Pick $j \in \hat{\Omega}$ according to the probability $\hat{p}(i,\cdot)$.
(3) Pick $y \in \Omega$ according to the probability $\mu_j(\cdot)$ where $\mu_j(x)$ is the equilibrium probability distribution of the original chain conditional on $C(x) = j$

$$\mu_i(x) = \frac{\mu(x)W(x,i)}{\sum_{y=1}^{n} \mu(y)W(y,i)} \tag{19}$$

The diagram associated with this dynamics is

$$\begin{array}{ccc} x \in \Omega & \xrightarrow{\tilde{P}(x,\cdot)} & y \in \Omega \\ {\scriptstyle W(x,\cdot)}\downarrow & & \uparrow{\scriptstyle \mu_j(\cdot)} \\ i \in \hat{\Omega} & \xrightarrow[\hat{P}(i,\cdot)]{} & j \in \hat{\Omega} \end{array} \tag{20}$$

and, assuming that (6) holds, it can be represented by the natural extension of the lift (18).

Proceeding as in Sec. 2.1 we can now determine the optimal W and $\hat{P}$ by minimizing the objective function (9) with W_d replaced by W, $E(\hat{P},W)$, over all permissible W and $\hat{P}$. This problem, however, turns out to be more complicated in the present case than it was for deterministic clustering. In particular, because of the constraints on $\hat{P}$ and W, the minimizer of $E(\hat{P},W)$ on $\hat{P}$ at fixed W is non-explicit in general, which means that the minimization of $E(\hat{P},W)$ over both $\hat{P}$ and W has to be performed numerically. This is the topic of the next section. The following diagram shows the connection between the objects introduced above.

$$\begin{array}{ccc} P & & \tilde{P} = L_W(\hat{P}) \\ {\scriptstyle \text{inverse lift of } P \text{ wrt } W}\downarrow & \searrow^{\min E} & \uparrow{\scriptstyle \text{lift of } \hat{P} \text{ wrt } W} \\ \hat{L}_W(P) & \longleftarrow & W, \hat{P} \end{array}$$

2.3. **The constrained minimization problem.** Finding the optimal $\hat{P}$ and W leads to the following constrained variational problem: **minimize**

$$\min_{W,\hat{P}} E(\hat{P},W) = \sum_{x,y=1}^{n} \frac{\mu(x)}{\mu(y)} \Big(P(x,y) - L_W(\hat{P})(x,y) \Big)^2 \tag{21}$$

subject to the**constraints**:

$$W(x,i) \geq 0, \qquad \forall x \in \Omega, i \in \hat{\Omega} \tag{22}$$

$$\sum_{i=1}^{m} W(x,i) = 1, \qquad \forall x \in \Omega \tag{23}$$

$$\hat{P}(i,j) \geq 0, \qquad \forall i,j \in \hat{\Omega} \tag{24}$$

$$\sum_{j=1}^{m} \hat{P}(i,j) = 1, \qquad \forall i \in \hat{\Omega} \tag{25}$$

$$\hat{\mu}(i) = \sum_{x=1}^{n} W(x,i)\mu(x), \qquad \forall i \in \hat{\Omega} \tag{26}$$

Since (21) is nonconvex and subject to the constraints in (22)– (26), this constraint minimization is nontrivial and most of the sequel of this paper is concerned with the development of an algorithm to perform it. Before getting there, however, we discuss some specific difficulties that we will have to deal with when minimizing (21).

2.4. **No unique solution in general.** Because we deal with a non-convex functional (21), an obvious question is whether there is a unique minimizer or not. The first trivial observation is that the numbering of the clusters cannot have any effect on the resulting lift $\tilde{P}$ and therefore a renumbering will not change the energy E. This means, if $(W, \hat{P})$ was a minimizer, $(W\Pi, \Pi^T \hat{P}\Pi)$ would be a solution, too, for any permutation matrix Π. Clearly, this not much of an issue since all these minima have the same interpretation.

There is, however, a more fundamental difficulty, namely that in general there are uncountably many minimizers. To see this consider the following transformed matrices:

$$\begin{aligned} \hat{Q}(i,j) &= \frac{\sqrt{\hat{\mu}(i)}}{\sqrt{\hat{\mu}(j)}}\hat{P}(i,j) \\ V(x,i) &= \frac{\sqrt{\mu(x)}}{\sqrt{\hat{\mu}(i)}}W(x,i) \\ Q(x,y) &= \frac{\sqrt{\mu(x)}}{\sqrt{\mu(y)}}P(x,y) \\ \tilde{Q}(x,y) &= \frac{\sqrt{\mu(x)}}{\sqrt{\mu(y)}}L_W(\hat{P})(x,y) \end{aligned} \tag{27}$$

By definition of the lift in (18), the last equation in (27) can be written as $\tilde{Q} = V\hat{Q}V^T$, and in terms of these new matrices, the objective function (21) can be re-expressed as the following functional of $\hat{Q}$ and V:

$$\bar{E}(\hat{Q}, V) = \sum_{x,y=1}^{n} \Big(Q(x,y) - (V\hat{Q}V^T)(x,y)\Big)^2 \tag{28}$$

Clearly

$$\bar{E}(\hat{Q}, V) = \bar{E}(\hat{Q}_A, V_A) \tag{29}$$

for any V_A and $\hat{Q}_A$ satisfying

$$V_A = VA, \qquad \hat{Q}_A = A^{-1}\hat{Q}A^{-T} \tag{30}$$

for any invertible $m \times m$ matrix A. As a result the minimizer of (29) is not unique. Of course, the transformation (30) leads to new W_A, $\hat{P}_A$ and $\hat{\mu}_A$ defined as

$$\hat{\mu}_A = (A^T\sqrt{\hat{\mu}})^2, \qquad W_A = WT_A, \qquad \hat{P}_A = T_A^{-1}\hat{P}T_A^{-T} \tag{31}$$

with $T_A = D_{\sqrt{\hat{\mu}}}^{-1} A D_{\sqrt{\hat{\mu}_A}}$ and so A cannot be chosen arbitrarily, because W_A, $\hat{P}_A$ and $\hat{\mu}_A$ must also satisfy the constraints (6), (12) and (16). A direct calculation shows that if W, $\hat{P}$ and $\hat{\mu}$ satisfy (6) and (12), then W_A, $\hat{P}_A$ and $\hat{\mu}_A$ also satisfy these constraints if A is such that

$$\begin{aligned} A^T A &= Id \\ (VA)(x,i) &\geq 0, \qquad \forall x \in \Omega, i \in \hat{\Omega} \\ \hat{\mu}_A(i) &\geq 0, \qquad \forall i \in \hat{\Omega} \\ (A^T\hat{Q}A)(i,j) &\geq 0, \qquad \forall i,j \in \hat{\Omega}. \end{aligned} \tag{32}$$

Obviously a permutation Π has all these properties, but rotations may also work if the rotation angle is small enough to keep the non-negativity constraints satisfied. There will be no possible rotation if the following holds:

$$\forall i \in \hat{\Omega} \quad \exists x \in \Omega \ : \ W(x,i) = 0. \tag{33}$$

In this case indeed, each of the $(m-1)$-dimensional hyperplanes that generate the boundary of the non-negativity constraint set defined by (32) contains at least one point $V(x,i)$. As a result if we apply A to the matrix V, all the points in the boundary of the non-negativity constraint set will be rotated in the same direction and not all these points can simultaneously stay inside of the set. This means that if (33) holds only $A = Id$ satisfies all the constraints in (32). In the general case, however, (33) does not hold, nontrivial rotations are possible, and the minimizers of (29) form a continuous level set of minimizers. In order to understand how to handle this problem we have to show how rotations can be used to characterize a level set of minimizers: If (33) does not hold for a local minimizer of $E(\hat{P}, W)$, then there is a continuous set of rotations matrices that respect the constraints in (32). Each rotation A can be represented in a canonical way by the product of $m(m-1)$ elementary rotations

$$A = \prod_{\substack{i,j=1 \\ j>i}}^{m} A_{i,j}. \tag{34}$$

Here $A_{i,j}$ denotes a rotation restricted to the two-dimensional plane $E_{i,j} = \text{span}\{e_i, e_j\}$, where e_k denotes the k-th unit vector and i, j range from 1 to m with $j > i$. For each of these planes there will be a real closed interval $\Gamma = [\gamma^-; \gamma^+], \gamma^- < \gamma^+$ and for each angle $\gamma \in \Gamma$ we will have an associated rotation matrix. In the subsequent, we will denote the set of all permissible rotations by $\mathcal{A}$.

The nonuniqueness is not necessarily a problem, however, and it can be exploited. Indeed we can choose a special rotation A to select a specific minimizer $(W_A, \hat{P}_A)$ out of the continuum of possible minimizers with special properties that make their interpretation easier, in particular to choose the number m of clusters. Sec. 3 is devoted to this issue.

Remark 2.1. *The first constraint in (32) which says that A must be orthogonal is sufficient but not necessary in order that W_A, $\hat{P}_A$ and $\hat{\mu}_A$ satisfies the non-negativity constraints. However, orthogonal matrices are sufficient for the purpose of the discussion in Sec. 3 and so we shall restrict ourselves to those.*

3. NUMBER OF CLUSTERS

Up to now we have assumed the number of clusters m is given *a priori*. This is not completely satisfactory, because knowing what is the "right" number of clusters means knowing a lot about the structure of the Markov chain and its transition matrix P. It cannot be assumed that we have this knowledge *a priori*. Next we investigate how to determine *a posteriori* the number of clusters which are in some way *essential* for the Markov chain. Let us first clarify what we mean with *essential*.

We start with the observations that $E(W,\hat{P}) = 0$ *iff* $P = L_W(\hat{P})$ and $\min E(W,\hat{P}) \to 0$ as $m \to n$ since $L_W(\hat{P}) = \hat{P} = P$ in that limit. So unless we can represent exactly the original transition matrix P by $L_W(\hat{P})$ for some $m < n$, the energy of the minimizer will decrease if we increase the cluster number m. This means that in general we cannot hope to determine m simply by looking at the energy of the minimizer.

If we assume that we have already computed a (perhaps local) minimizer of the constrained minimization problem for some given m, another possible strategy is to try to reduce the number of clusters to some $\tilde{m} < m$ by using our insights about the non-uniqueness. To this end, in Sec. 3.1 we will first explain how to identify specific minimizers in a level set of minimizers in which one or several of the clusters are not used (in the sense that $\hat{\mu}(i) = 0$ for some $i \in \hat{\Omega}$). Next in Sec. 3.2 we will extend this approach to the much more frequent case of almost unused clusters ($\hat{\mu}(i) \approx 0$ for some $i \in \hat{\Omega}$). If maximally $r < m$ clusters are almost unused then $\tilde{m} = m - r$ may be used as an *a posteriori* indicator for the cluster number. We will then see in Sec. 3.3 how to use this observation in practice by considering what we will call kinetically almost separated clusters. Finally, some algorithmic aspects will be discussed in Sec. 3.4.

3.1. **Unused clusters.**

Consider a level set of minimizers that are generated by rotations from $\mathcal{A}$. In order to reduce the cluster number one can look for the minimizer in this level set which uses as few clusters as possible, i.e. such that we have $W_A(x,i) = 0$ for all $x \in \Omega$ and for as many $i \in \hat{\Omega}$. Clearly, this amounts to finding the rotation $A \in \mathcal{A}$ which brings $W_A = WT_A$ into this form. Let us first give an illustrative example.

Example 3.1. *Take*

$$W = \begin{pmatrix} 1 & 0 & 0 \\ 0.8 & 0.1 & 0.1 \\ 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{pmatrix}, \quad \hat{P} = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.4 & 0.2 & 0.4 \\ 0 & 0.4 & 0.6 \end{pmatrix}, \quad \mu = \begin{pmatrix} 0.3 \\ 0.2 \\ 0.2 \\ 0.1 \\ 0.2 \end{pmatrix} \quad \hat{\mu} = \begin{pmatrix} 0.66 \\ 0.17 \\ 0.17 \end{pmatrix}.$$

Imagine that we have $P = L_W(\hat{P})$. Then of course $(W,\hat{P})$ would be a minimizer, because the approximation is exact, i.e. $E(W,\hat{P}) = 0$. But with the rotation matrix

$$A = \begin{pmatrix} 1.0000 & 0 & 0 \\ 0 & 0.7071 & -0.7071 \\ 0 & 0.7071 & 0.7071 \end{pmatrix}$$

we get an equivalent minimizer of the form

$$W_A = \begin{pmatrix} 1 & 0 & 0 \\ 0.8 & 0.2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \hat{P}_A = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.2 & 0.8 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \hat{\mu}_A = \begin{pmatrix} 0.66 \\ 0.34 \\ 0 \end{pmatrix}.$$

The third cluster is not used in this case, so it would be possible to get the same lift with $m = 2$.

3.2. **Almost unused clusters.** In general we will not be able to achieve $W_A(\cdot, i) = 0$ for some $i \in \hat{\Omega}$, i.e. there will be no fully unused clusters. We thus should ask whether we can find rotations that make some clusters *almost unused*. In order to understand what this means, first observe that if $W_A(x, i) = 0 \ \forall x \in \Omega$, then we have $\hat{\mu}_A(i) = \sum_{y=1}^{n} W_A(x, i)\mu(x) = 0$. However, setting entries of $\hat{\mu}_A$ to zero means increasing the 2-norm $\|\hat{\mu}_A\|_2$, because of the probability constraint $\|\hat{\mu}_A\|_1 = 1$. (Too see this, remember that the unit spheres of the 1- and 2-norm intersect only in the unit vectors, which have the maximal number of zero entries under all probability vectors.) This suggest to determine A via the maximization problem

$$\max_A \|\hat{\mu}_A\|_2. \tag{35}$$

subject to the constraints (32). Since $\hat{\mu}_A = (A^T\sqrt{\hat{\mu}})^2$ (see (31)), an algorithmic procedure for maximizing $\|\hat{\mu}_A\|_2$ is as follows:

(a) parametrize A by the angles $\gamma_{i,j} \in \Gamma_{i,j}$ defined above, i.e. describe A in the canonical way given by (34), and;
(b) perform the maximization of $\|\hat{\mu}_A\|_2$ in each plane $E_{i,j}$ separately, i.e. find the largest possible angles $\gamma_{i,j}$, such that the non-negativity constraints of (32) are still satisfied.

Having solved this maximization problem the rotation matrix A leads to a special choice of minimizer $(W_A, \hat{P}_A)$ out of the level set as in (31). Let us denote this clustering by $(W_A^*, \hat{P}_A^*)$.

A possible strategy is to eliminate almost unused clusters such that $\hat{\mu}_A^*(i) < \mathrm{tol}$ for some given threshold tol. The problem with this strategy, however, is that it is not clear what tol should be, and the answer to this question depends sensitively on the original invariant measure since all entries of $\hat{\mu}_A^*$ will decrease if we take more clusters. A practical way around this difficulty is proposed next.

3.3. **Almost separated clusters.** Because the map C is random, two states $x, y \in \Omega$ are mapped onto different clusters in some realizations, and on the same one in some other realizations. This means that the states x and y are not strictly separated by the clustering. However, there may be partition of Ω into $\tilde{m}$ cluster $S_i \subset \Omega$ with $i = 1, ..., \tilde{m}$, which are separated in most of the clustering realizations, i.e. which are such that

$$\text{if } \ x \in S_i, y \in S_j \text{ with } i \neq j \ \text{ then } \ \mathbb{P}[C(x) = C(y)] \leq \epsilon_{\text{sep}} \tag{36}$$

Here $\epsilon_{\text{sep}} > 0$ is a small adjustable parameter which measures the strength of the separation of the sets $(S_i)_{1,...,\tilde{m}}$: the smaller ϵ_{sep}, the more stringent condition (36) is, i.e. the smaller the cardinal $\tilde{m}$ of the partition(s) $\{S_1, \ldots, S_{\tilde{m}}\}$ such that (36) holds. Therefore, given ϵ_{sep}, we can identify the partition such that (36) holds which has maximum cardinal, and take this maximum as the *a posteriori* estimator $\tilde{m} \leq m$ for the effective number of clusters needed. We can then re-apply our soft-clustering procedure by setting $m = \tilde{m}$. Because the family of random variables $(C(x))_{x\in\Omega}$ is independent we have

$$\mathbb{P}[C(x) = C(y)] = \sum_{i=1}^{m} \mathbb{P}[C(x) = i]\mathbb{P}[C(y) = i] = \sum_{i=1}^{m} W(x, i)W(y, i). \tag{37}$$

This measure of separability might not be well-defined because of non-uniqueness of the minimizer W, and we can again exploit this uniqueness. To see how let us define

$$S(x,y) := (\mathbb{P}[C(x) = C(y)]) = (WW^T)(x,y). \tag{38}$$

The first observation is that in the limit $\epsilon_{\text{sep}} \to 0$ all minimizers from the continuous level set that we had found in (2.4) lead to the same completely separated states, because

$$S(x,y) = 0 \Leftrightarrow S_A(x,y) = (W_A W_A^T)(x,y) = 0 \tag{39}$$

(The proof can be found in the appendix.) That is, for $\epsilon_{\text{sep}} = 0$ we will get the same *a posteriori* estimator $\tilde{m} \leq m$ regardless of the choice of minimizer W_A. Please note also that if we had $\tilde{m} = m$ there would not be a continuous level set of equivalent minimizers at all (see (33)).

For $\epsilon_{\text{sep}} > 0$ this is not true, because we have in general

$$S \neq S_A, \text{ i.e. } \sum_{i=1}^{m} W(x,i)W(y,i) \neq \sum_{i=1}^{m} W_A(x,i)W_A(y,i).$$

Therefore we have to specify which minimizer we want to take for analysis of separability in order to have a well-defined *a posteriori* estimator. Herein we simply take the special minimizer W_A^* that solves the maximization problem in (35), $\max_A \|\hat{\mu}_A\|_2$.

Remark 3.2. *Using the definition (38) the maximization problem (35) leads to*

$$\|\hat{\mu}_A\|_2^2 = \mu^T S_A \mu = \sum_{x,y=1}^{n} \mu(x)\mu(y)(W_A W_A^T)(x,y) \sum_{x,y=1}^{n} \mu(x)\mu(y) S_A(x,y). \tag{40}$$

That is, the solution W_A^* to (3.2) leads to a separation matrix $S_A^*(x,y)$, which tends to have higher values for pairs of states x, y whose equilibrium probabilities $\mu(x)$, $\mu(y)$ has higher values. Because we need $S_A^*(x,y) \leq \epsilon_{sep}$ for separation this automatically leads to a more stringent condition for states with relatively high probability. Thus, unlike the procedure in Sec. 3.3, the one above is less sensitive to the value of the equilibrium distribution, which makes the choice of $\epsilon_{sep} > 0$ easier.

3.4. **Algorithmic determination of number of clusters.** The *a posteriori* estimator $\tilde{m}$ of the number of clusters can be found by calculating the maximal pairwise ϵ_{sep}-separated system according to the separation matrix S_A^*, i.e. $\tilde{m}$ is the maximal number, such that there is a set $\{x_1, ..., x_{\tilde{m}}\}$ with $S_A^*(x_i, x_j) < \epsilon_{\text{sep}}\ \forall i \neq j$.

To interpret this in a graph theoretical way we introduce the separation graph $\mathcal{G} = (V, E)$ with $V = \Omega$ and adjacency matrix A, that is defined by $A(x,y) = 1 \Leftrightarrow S_A^*(x,y) < \epsilon_{\text{sep}}$. Then the *a posteriori* estimator is exactly the clique number of $\mathcal{G}$ and we can calculate $\tilde{m}$ by solving the maximum clique problem, that is equivalent to the maximal independent set problem. Both are NP-complete but in special cases it is possible to calculate a solution in polynomial time in $|V| = n$. There are also a lot of algorithms in graph theory and combinatorial optimization, e.g. [21], which calculate the clique number approximately with polynomial or even linear effort. There are also *a priori* bounds on the clique number depending on the number of nodes and edges. More details on the maximum clique problem can be found in the extensive literature on graph theory or combinatorial optimization, e.g. [22].

Finally, we note that if the procedure searches for an $\tilde{m} \leq m$ given m; if it returns $m = \tilde{m}$, we should check for more clusters by doubling m.

4. Constrained Minimization Algorithm

In this section we explain how to modify the standard approach in [20] to solve the optimization problem described in Sec. 2.3. We stress that there are several other possible algorithms (interior point methods, Newton, mirror prox, etc.) to perform this optimization, and some of these algorithms may be more efficient. The method we use has one crucial advantage, however: it finds a minimizer which fulfills the constraints *exactly*, not approximately as may other methods do. We also note that the complexity per iteration of our algorithm is dominated by the evaluation of the functional.

We begin by noting that there is a unique solution to the *unconstrained* optimization problem

$$\hat{P}^* = \text{argmin}_{\hat{P}} E(\hat{P}, W)$$

for *fixed* full rank matrix W. The minimizer $\hat{P}^*$ can be calculated in two steps as follows:

$$\text{Set } V := W^T D_\mu W, \qquad \text{where } D_\mu = \text{diag}\{\mu(1), .., \mu(n)\} \tag{41}$$

$$\text{Solve } V\hat{P}^* D_{\hat{\mu}}^{-1} V = W^T D_\mu P W, \qquad \text{where } D_{\hat{\mu}} = \text{diag}\{\hat{\mu}(1), .., \hat{\mu}(m)\} \tag{42}$$

The minimizer $\hat{P}^*$ will automatically fulfill all constraints, except perhaps (24). If (24) is violated by the solution $\hat{P}^*$, we will have to use numerical minimization techniques.

The above property leads us to the following general subspace iteration scheme for optimization:

Input: $P \in \mathbb{R}^{n,n}$, Parameters m, ϵ, initial matrix $W^{(0)}$
Output: Approximations $W, \widehat{P}$
Initialize iteration index i=0
Repeat
- (M1) Solve $\hat{P}^{(i)} = \text{argmin}_{\hat{P}} E(\hat{P}, W^{(i-1)})$ under constraints (24), (25), and $\hat{\mu}^T = \hat{\mu}^T \hat{P}$ with $\hat{\mu}$ due to (26). If $\hat{P}^{(i)} = \hat{P}^*$ with $\hat{P}^*$ from (42) using $W = W^{(i-1)}$ fails to satisfy the constraints then apply appropriate numerical constrained minimization techniques.
- (M2) Solve $W^{(i)} = \text{argmin}_W E(\hat{P}^{(i)}, W)$ under constraints (22) and (23) via appropriate numerical constrained minimization techniques.

until $E(W^{(i-1)}, \hat{P}^{(i-1)}) - E(W^{(i)}, \hat{P}^{(i)}) < \epsilon$
Output $W = W^{(i+1)}$, and $\widehat{P} = \hat{P}^{(i+1)}$.

To perform the numerical constrained minimization, we propose to use an appropriately modified projected line search method using gradient descent. Next, we explain this technique for the step (M2) from above (i.e., $\hat{P} = \hat{P}^{(i)}$ is fixed and we have to minimize E wrt. to W); it can be used in complete analogy for step (M1). First we explain the basic procedure for choosing the stepsize of every step of this iterative method; then we discuss how to handle the equality and inequality constraints.

Stepsize. At every step we choose a search direction S according to the present state W, such that $\langle \nabla_W E, S \rangle < 0$, and perform an efficient line-search in this direction. Therefore we have to find an $\alpha > 0$, the stepsize, such that (i) $E(W + \alpha S) < E(W)$ and (ii) the energy reduction $E(W + \alpha S) - E(W)$ is acceptable, i.e. big enough. In order to find α, one considers the ratio q of the reduction and the linear forecast

$$q(\alpha) = \frac{E(W + \alpha S) - E(W)}{\alpha \nabla_W E^T S}.$$

The denominator is always negative, so we have an energy reduction *iff* $q(\alpha) > 0$. Now one could choose from several line search conditions to get an acceptable α, for example Armijo-Goldstein (AG) conditions:

$$q(\alpha) > \sigma \text{ and } \exists \tilde{\alpha} < \beta\alpha : q(\tilde{\alpha}) < 0$$

with predefined parameters $\sigma \in (0, \frac{1}{2})$ and $\beta > 1$. If an α satisfying the AG conditions can be found, then we have a guaranteed reduction (because $q(\alpha) > \sigma > 0$). At the same time we have to make sure that the step is not too small, because in the range of $\beta\alpha$ there is a state that does not fulfill the reduction condition. The determination of such an α can be done by backtracking algorithms.

Equality constraints. One possible choice of the search direction S is steepest (gradient) descent, i.e., $S = -\nabla_W E$. However, we have to find a stochastic matrix W along that direction and this may be an infeasible. Denote the current iterate of the clustering matrix by $W_0 \in \mathcal{S}_1$ and observe that the set $\mathcal{S}_1 = \{A \in \mathbb{R}^{n,m} : \sum_{j=1}^{m} A_{ij} = 1\}$ (rowsum one matrices) is an affine subspace of $\mathbb{R}^{n,m}$. Obviously it holds $S_1 = W_0 + S_0$ with W_0 any matrix in $\mathcal{S}_1$ where $\mathcal{S}_0 = \{A \in \mathbb{R}^{n,m} : \sum_{j=1}^{m} A_{ij} = 0\}$ denotes the linear space of rowsum zero matrices. Thus we have to choose a search direction $S \in \mathcal{S}_0$, because then

$$W_0 + \alpha S \in \mathcal{S}_1 \ \forall \alpha.$$

We choose S as orthogonal projection of $-\nabla_W E$ onto $\mathcal{S}_0$, i.e. $S = -\nabla E T T^T$ with

$$\tilde{T}_{ij} = \begin{cases} 1, & \text{if } j \leq i \\ -i, & \text{if } j = i+1 \\ 0, & \text{else} \end{cases} \qquad i = 1, ..., m-1, \quad j = 1, ..., m$$

and T being the normalized matrix with columns

$$T_i = \frac{\tilde{T}_i}{\|\tilde{T}_i\|}.$$

Inequality constraints. We also need to account for the non-negativity constraints. This means optimizing on the set $\mathcal{S}_1^+ = \{A \in S_1 | A_{i,j} \geq 0\}$, which is a closed convex subset of the affine space $\mathcal{S}_1$. Now two problems arise:

(1) The line-search minimum $W_{k+1} = W_k + \alpha S_k$ could lie outside of $\mathcal{S}_1^+$. This can be easily cured by just setting $q(\alpha) < 0$, if $W_k + \alpha S_k \notin \mathcal{S}_1^+$.

(2) We could reach the boundary. If $W_k \in \partial\mathcal{S}_1^+$ in the most cases $-\nabla E(W_k)$ and therefore S_k will point out of $\mathcal{S}_1^+$. Then there would not be any feasible $\alpha \geq 0$ that satisfies the Armijo-Goldstein conditions. This is precisely the case when one of the two following conditions holds for some x, i:
   (i) $W(x,i) = 0$ and $\nabla_W E_{x,i} > 0$,
   (ii) $W(x,i) = 1$ and $\nabla_W E_{x,i} < 0$
   This problem can be solved by doing the line-search row-wise and project only the elements, for which (i) or (ii) do not hold, and set $S_x(i) = 0$ for all x, i with (i) or (ii). If there are k elements in a row which should be projected, this can be done by using the left upper $(k, k-1)$-submatrix of $T \in \mathbb{R}^{m,m-1}$, $k \leq m$, which defines the projection in $\mathbb{R}^k$. In practise one should not wait until the boundary is completely reached, because this will imply taking very small line-search steps. There should be a tolerance tol, such that W is treated like a boundary point, if $\text{dist}(W, \partial\mathcal{S}_1^+) < tol$.

Algorithm. Putting everything together we arrive at the following constrained minimization algorithm:

Input: $P \in \mathbb{R}^{n,n}$, Parameters $m, \sigma \in (0, \frac{1}{2}), \beta > 1, tol, \epsilon > 0$
Output: $W, \widehat{P}$
Initialize W^0, $\widehat{P}^0 \in \mathcal{S}_1^+$ arbitrarily, $k = 0$
while $|E(W^{k+1}, \widehat{P}^{k+1}) - E(W^k, \widehat{P}^k)| > \epsilon$ **do**
  $W^{k+1} := W^k$
  $G := \nabla_W E(W^k, \widehat{P}^k)$
  for *x=1 to n* **do**
    $I_x := \{j : (i) \text{ and } (ii) \text{ do not hold for } (G_x(j), W^{k+1}(x, j)\}$
    $l := \#I_x$
    $\phi : \{1, ..., l\} \to I_x$ onto and one-to-one
    Define $\widehat{G}_i := G_x(\phi(i)),\ i = 1, ..., l$
    Set T_l as left upper $(l, l-1)$-submatrix of projection T
    $\widehat{S} := -\widehat{G} T_l T_l^T$
    $S_i := \begin{cases} \widehat{S}_{\phi^{-1}(i)} & , i \in I_x \\ 0 & , \text{else} \end{cases},\ i = 1, ..., n$
    Perform a line-search in direction S to update $W^{k+1}(x, \cdot)$
  end
  $\widehat{\mu} = (W^{k+1})^T \mu$
  $V = (W^{k+1})^T D_\mu W^{k+1}$
  Solve the linear system $V \widehat{P}^{k+1} D_{\widehat{\mu}}^{-1} V = (W^{k+1})^T D_\mu P W^{k+1}$
  if *(24) does not hold* **then**
    Perform line-searches as above to find $\widehat{P}^{k+1} \in \mathcal{S}_1^+$
  end
end

After termination, if needed one can compute the *a posteriori* estimator $\tilde{m}$ for the number of clusters as introduced in Sec. 3.

One can choose to do the row-wise line-searches in succession or in parallel. It is also possible to perform line-searches in matrix valued directions, i.e. optimizing everything at the same time. In this case one needs a strategy which avoids that reaching the boundary in one row of W will stop each line-search so that the clustering of the other states can only get better in another iteration. One possibility is to start with a larger boundary tolerance tol and let it decrease with each iteration.

Initial iterates. Concerning the choice of the initial values for W and $\widehat{P}$ one has several options. The one preferred by the authors is to use as initial value a deterministic clustering computed by means of an efficient combinatorial method. In general one has only to make sure that the initial values W^0 and $\widehat{P}^0$ fulfill the constraints in (22)–26 because this guarantees that each iterate will then lie inside of the feasible set. One could use completely random but normalized initial values too. The advantage of using a good deterministic clustering as initial value is that the algorithm starts on the boundary of the permissible set and the projected search directions will point into this set. Hence, even if one uses the matrix-valued variant, there will be no slow down effect on the convergence when one state approaches the boundary.

Remarks on Complexity. Computing the lift $\tilde{P}$ for given W and $\hat{P}$ is obviously of order $O(n^2m^2)$ and computing the energy from the lift has complexity $O(n^2)$. Therefore evaluating the functional requires $O(n^2m^2)$ operations. The computation of the projected gradient with respect to W and $\hat{P}$ also has the same complexity, and so the total complexity of the algorithm is $O(n^2m^2)$. Notice however that this statement assumes that the number of iterations of our gradient descent technique is bounded with respect to to n and m. We cannot guarantee that it does not increase with n and m; this is a general possible pitfall of (constrained) gradient descent techniques. We will not comment on this issue in general; however, our numerical experiments (with moderate values for m and n) did not show such an increase.

Matrix-valued variant. Performing one iteration in the matrix-valued variant of the algorithm means performing one line-search in the direction of negative projected gradient. As a result we have to evaluate the functional several times to get an acceptable step-length α. In practise it will need only a few backtracking steps to find such an α, and there also is a fixed upper bound inside of the backtracking algorithm for this number of steps. In other words the complexity of doing one iteration with the matrix-valued variant is $O(n^2m^2)$. Obviously one can also try to minimize the required number of iterations till convergence choosing good initial values via the strategies discussed above.

Line-search for each row separately. In this case each iteration consists of n line-searches and therefore the cost of one iteration is $O(n^3m^2)$. However, we observed that the convergence of this method (in terms of required number of iteration) is much faster, even for random initial values. The numerical examples below were computed with at most $5m$ iterations. Moreover one could approximate the line-search reduction quotient $q(\alpha)$ in a more sophisticated way than by simply evaluating the functional, because only one row of $W + \alpha S$ is changed with respect to W. This could again lead to a complexity that is $O(n^2m^2)$.

5. NUMERICAL EXAMPLES

As a first illustration of our approach,in Secs. 5.1 and 5.2 we cluster the test network with 30 nodes and associated 30×30 transition matrix P shown in Fig. 1. Then in Sec. 5.3

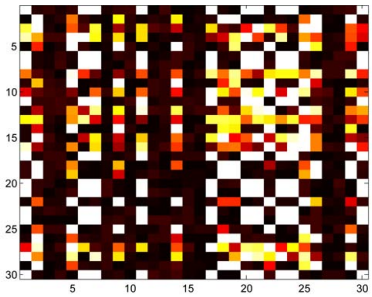

FIGURE 1. Transition matrix P. Brighter values indicate higher entries, i.e., higher transition probabilities.

we apply our method to cluster the transition network resulting from a diffusion process in a double-well energy landscape.

In all the calculations reported below, the initial values of the optimization algorithm were chosen randomly and normalized to be consistent with the stochastic constraints.

5.1. **Test network: strong cluster separation, $\epsilon_{\text{sep}} = 0.05$.** Fig. 2 shows the *a posteriori* estimator $\tilde{m}$ plotted against the number m of clusters used in the minimization. The

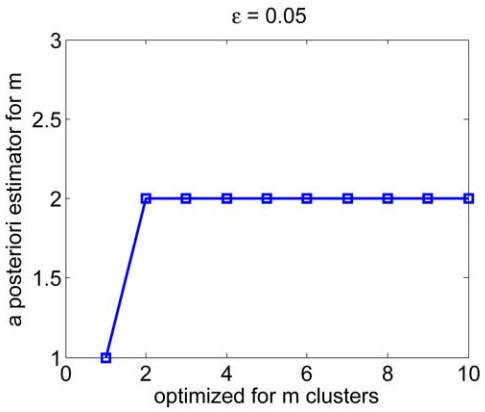

FIGURE 2. *A posteriori* estimator $\tilde{m}$ for separation parameter $\epsilon_{\text{sep}} = 0.05$ versus number m of clusters used in the minimization for the 30×30 matrix P mentioned in the text. The results indicate 2 clusters.

results in the figure clearly indicate that we should take only two clusters. Setting $m = 2$ results in the clustering $W(\cdot, i), i = 1, 2$ shown in Fig. 3. The inverse lift gives the transi-

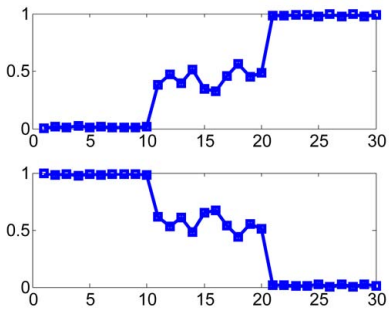

FIGURE 3. Clustering $W(\cdot, i), i = 1, 2$ as function of x for $m = 2$.

tion rates on the clusters and it also shows strong metastability.

$$L_W(P) = \begin{pmatrix} 0.8403 & 0.1597 \\ 0.1397 & 0.8603 \end{pmatrix}$$

These results can be understood by re-ordering the original matrix P appropriately. Fig. 4 shows the effect of some permutation of P that uncovers the block structure hidden in P. We observe that indeed two clusters are visible which are connected by transition states. Finally, Fig. 5 shows the lift $\tilde{P}$ associated with the clustering with $m = 2$ clusters. We observe that the optimal clustering reproduces the (hidden) block structure of the original matrix; however, as a lift of the 2×2 matrix it cannot reproduce the fine structure within the blocks.

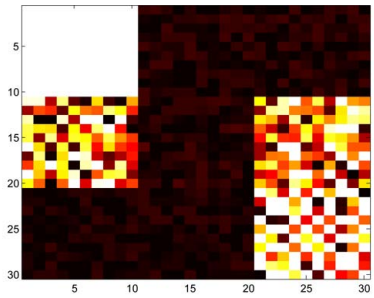

FIGURE 4. Re-ordering $\Pi P \Pi^T$ of the 30×30 matrix P such that the dominant blocks hidden in P become visible.

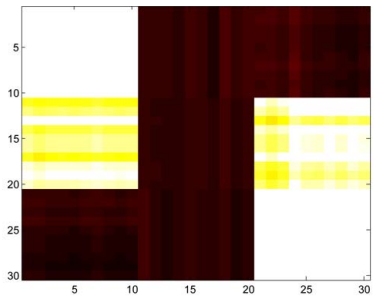

FIGURE 5. Lift $\tilde{P}$ based on clustering with $m = 2$.

5.2. **Test network: weaker cluster separation, $\epsilon_{\text{sep}} = 0.1$.** Now we increase ϵ_{sep} to the value of 0.1, i.e., we lower our separation requirement. Looking at the *a posteriori* estimator $\tilde{m}$ in this case, we observe that not all m lead to $\tilde{m} = 2$, see Fig. 6.

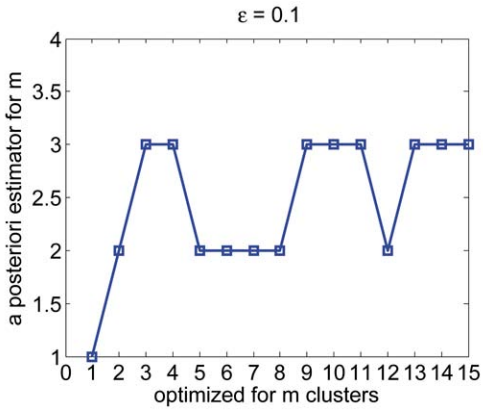

FIGURE 6. *A posteriori* estimator $\tilde{m}$ for separation parameter $\epsilon_{\text{sep}} = 0.1$ versus number m of clusters used in the minimization for the 30×30 matrix P mentioned in the text. The results indicate 2 or 3 clusters.

According to Fig. 6 we should also consider $\tilde{m} = 3$ clusters if we are less strict in terms of separation. The clustering with $m = 3$ results in the clustered matrix

$$\hat{P} = \begin{pmatrix} 0.9897 & 0.0012 & 0.0090 \\ 0.0001 & 0.9466 & 0.0534 \\ 0.0181 & 0.0656 & 0.9163 \end{pmatrix},$$

the inverse lift

$$L_W(P) = \begin{pmatrix} 0.8349 & 0.0684 & 0.0967 \\ 0.1435 & 0.5202 & 0.3363 \\ 0.1390 & 0.2317 & 0.6294 \end{pmatrix}$$

and the clustering and lift as given in Fig. 7. The inverse lift $L_W(P)$ shows that the

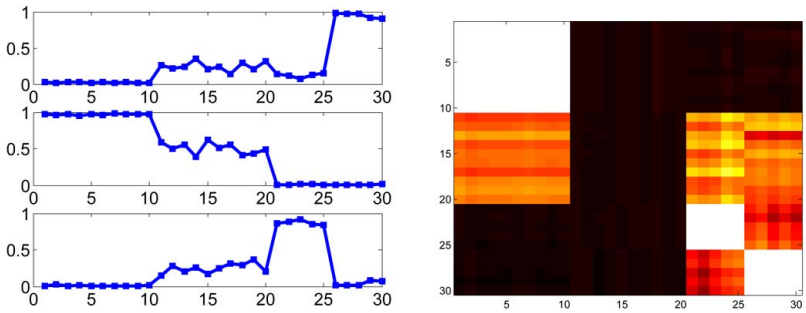

FIGURE 7. Results for $m = 3$. Left: Clustering $W(\cdot, i), i = 1, 2, 3$. Right: Lift $\tilde{P}$ resulting form $\hat{P}$ as given in the text.

second and third clusters are not as stable as the first one: this is why the only appear when $\epsilon_{\text{sep}} = 0.1$ (weaker cluster separation) and not when $\epsilon_{\text{sep}} = 0.05$ (strong cluster separation). The clustering $W(\cdot, i), i = 1, 2, 3$, and the lift $\tilde{P}$ show that the additional clusters uncover some additional hidden block structure in the lower left dominant block. In fact, this can also be made visible by some permutation $\tilde{\Pi}$ of P which is different from the one used in Fig. 4 but now uncovers more of the block structure hidden in P, see Fig. 8.

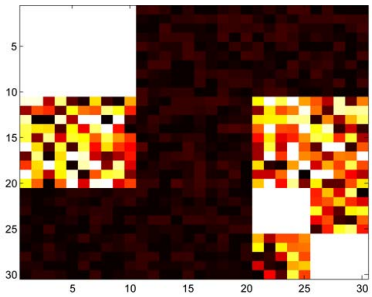

FIGURE 8. Reordering $\tilde{\Pi} P \tilde{\Pi}^T$ as explained in the text.

5.3. **Diffusion in potential energy landscape.** In our second example we consider the one-dimensional continuous stochastic process $(X_t)_{t\in\mathbb{R}}$ governed by the following stochastic differential equation

$$\gamma dX_t = -\nabla V(X_t)dt + \sigma dB_t \tag{43}$$

where $\sigma > 0$ is a parameter, B_t denotes a standard Brownian motion, and V is the potential illustrated in Fig.9.

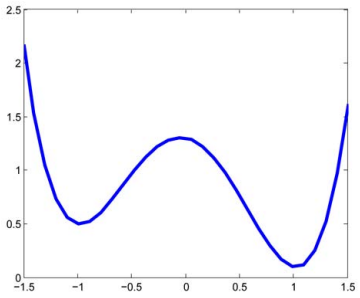

FIGURE 9. The potential V.

Equation (43) is used as follows to to construct a transition network. First, we discretize space by decomposing the real line into uniform boxes B_j, $j \in \mathbb{Z}$ with box size $h = 0.1$, and centers $b_j = jh$, $j \in \mathbb{Z}$. Then we compute a realization $\{x_k\}$ of the process X_t in the time interval $[0, 10000]$ by discretizing it in time using with the Euler-Maruyama scheme with time step $\Delta t = 0.002$, and performing $N = 5$ million time steps. In the specific realization that we studied, the time series enters the 31 boxes B_i, $i = -15, \ldots, 15$ only. We take these these boxes as nodes of our transition network. From the time series we then compute the transition probabilities between spatial discretization boxes with lag time $\tau = 0.1$, i.e. we compute the probability for going from box B_i to box B_j as

$$P(i,j) = \frac{\#\{t_k \in T_i : x_{k+\tau/\Delta t} \in B_j\}}{\#T_i}, \quad T_i = \{t_k : x_k \in B_i, k = 1, \ldots, N\}.$$

where $\#A$ means the cardinality of the set A. The resulting 31×31 transition matrix P of the network is shown in Fig. 10. We observe two clear clusters centered around the boxes belonging to the vicinity of the minima of the energy landscape and a clear transition region between these clusters belonging to the vicinity of the saddle point in the energy landscape. The metastability inherent in P is apparent from its dominant eigenvalues: $\lambda = 1.000, 0.980, 0.195$.

We did apply our approach to the transition matrix P. First, Fig. 11 shows the *a posteriori* estimator $\tilde{m}$ plotted against the number of clusters m used in optimization. The figure indicates that $\tilde{m} = 2$ and $\tilde{m} = 3$ are the two possible choices for the number of clusters, and so we re-applied our clustering procedure with $m = 2$ and $m = 3$.

Let us consider $m = 2$ first. The fuzzy affiliation functions for $m = 2$ are shown in Fig. 12. They indicate two clear cluster centers around the vicinity of the minima of the energy landscape where affiliations are close to deterministic and non-deterministic affiliations in the transition region. As expected, the shape of the affiliations show close similarity to the second eigenvector of the original matrix P.

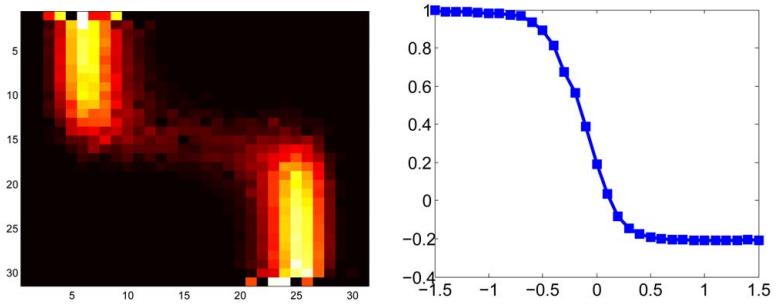

FIGURE 10. Left: Transition matrix P of diffusion-induced transition network; details as explained in the text. Right: Eigenvector associated with its second largest eigenvalue.

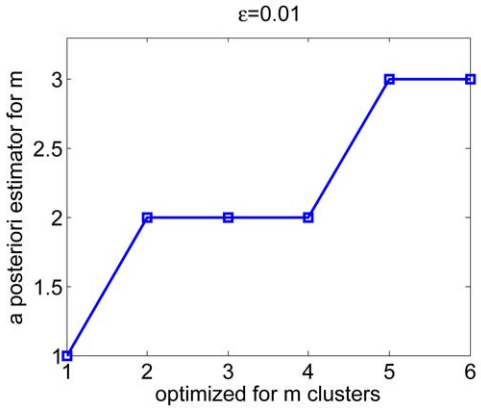

FIGURE 11. *A posteriori* estimator $\tilde{m}$ plotted against the number of clusters used in optimization.

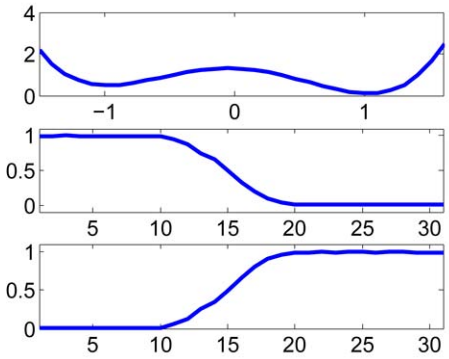

FIGURE 12. Results for $m = 2$. Top: potential energy landscape. Middle and bottom: $W(\cdot, 1)$ and $W(\cdot, 2)$.

Accordingly, the inverse lift $L_W(P)$ shows strong metastability:

$$L_W(P) = \begin{pmatrix} 0.9875 & 0.0125 \\ 0.0581 & 0.9419 \end{pmatrix}$$

The eigenvalues of the aggregated matrix $\hat{P}$ are $\lambda = 1.000, 0.989$, which are pretty close to the original dominant eigenvalues. This indicates that the original metastability is well kept in the fuzzy clustering.

$$\hat{P} = \begin{pmatrix} 0.9983 & 0.0017 \\ 0.0086 & 0.9914 \end{pmatrix}$$

The results of clustering with $m = 3$ are illustrated in Fig. 13. We see that the third cluster is only used in the transition region. The inverse lift is given by

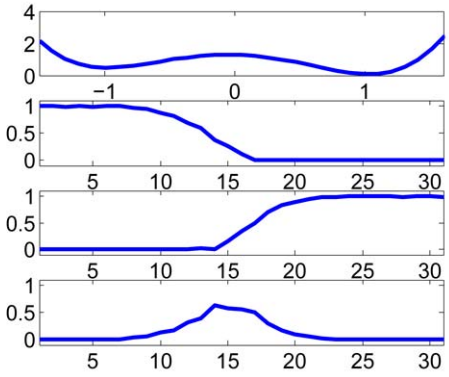

FIGURE 13. Results for $m = 3$. Top: potential energy landscape. Middle and bottom: $W(\cdot, i)$, $i = 1, 2, 3$.

$$L_W(P) = \begin{pmatrix} 0.9178 & 0.0406 & 0.0417 \\ 0.0085 & 0.9767 & 0.0148 \\ 0.3482 & 0.5909 & 0.0609 \end{pmatrix},$$

and we observe immediately that just cluster 1 and 2 are metastable and cluster 3 consists of transition states. This is backed up by the eigenvalues: $\lambda = 1.000, 0.920, 0.036$.

6. CONCLUSION

We considered optimal fuzzy aggregation of networks based on kinetic properties of the Markov chain associated with the network. This was done by comparing a suitably defined lifted transition matrix with the original one and minimizing the weighted Frobenius norm $\|\cdot\|_F$ of the difference between these matrices. The choice of this norm is natural since it bounds from above the difference between the eigenvalues of the lifted transition matrix and those of the original one. Thus optimal approximation in terms of $\|\cdot\|_F$ means optimal approximation in a kinetic sense.

Our approach leads to a nonlinear, nonconvex, constrained minimization problem with continuous levelset minimizers, and we showed how this nonuniqueness can be exploited to determine *a posteriori* the number of clusters to be used. Specifically, we suggested to pick this number as the maximal number of clusters which can be kinetically almost separated. This also permits to determine a specific minimizer from that level set.

For the minimization itself, we proposed some adapted version of restricted line search using projected gradient descent. The algorithm approximates the desired minimizer iteratively. The cost of each step is dominated by the evaluation of the functional and has complexity $O(n^2m^2)$. This in principle allows application to large networks; however, the

number of iterations is known to be possibly quite large for gradient descent techniques in application to non-convex constrained problems. The question of how the proposed algorithm performs for large networks will thus be subject of further research.

Finally, we note that since our fuzzy aggregation methodology is tailored to the kinetic properties of the Markov chain, we should expect that if the original Markov chain of the network exhibits metastability the optimal affiliation functions will be closely related to the dominant eigenvectors of the original transition matrix. This was indeed observed in the numerical experiments on diffusion in potential energy landscapes reported in Sec. 5.3. It would be interesting to give a more rigorous quantification of this property and this will be left to future research.

ACKNOWLEDGEMENTS

This work is part of a joint effort with Weinan E and Tiejun and the present paper should be viewed as a companion paper of [2] where a similar soft-clustering strategy is developed. The research of M. S. and Ch. Sch. was supported by the DFG Research Center MATHEON "Mathematics for Key Technologies" in Berlin and the work of E. V.-E. was partially supported by NSF grants DMS02-09959, DMS02-39625 and DMS07-08140, and ONR grant N00014-04-1-0565.

7. APPENDIX

7.1. **The weighted Frobenius norm.** Here we motivate the choice our choice for the functional (21) from the viewpoint that we are concerned with optimal clustering of Markov transition matrices. Consider some Markov process in discrete time with finite state space $\Omega = \{1, \ldots, n\}$ and transition matrix P. The transport of probability vectors induced by P is given by the associated transfer operator $T : L^p(\Omega) \to L^p(\Omega)$ with

$$\mathbf{T}u(x) = \sum_{y=1}^{n} u(y)P(y,x),$$

Assume that $\mathbf{T}$ has (unique) invariant measure $\mu > 0$ (satisfying $\mu^T P = \mu^T$), and that the associated Markov chain is reversible, i.e., satisfies detailed balance

$$\mu(x)P(x,y) = \mu(y)P(y,x). \tag{44}$$

Then, it is well-known that $\mathbf{T}$ is self-adjoint in the Hilbert space ℓ^2_μ with respect to the scalar product $\langle u, v\rangle_\mu = \sum_x u(x)v(x)\mu(x)$. ℓ^2_μ is the *natural* space associated with all Markov chains on Ω with (unique) invariant measure μ and, if they have to be compared, it is then best to do so in this space. Thus considering $\mathbf{T}$ as an operator in ℓ^2_μ, the transfer operator has to transport probability as in the original space, only weighted with respect to μ. Therefore $\mathbf{T}_\mu : \ell^2_\mu(\Omega) \to \ell^2_\mu(\Omega)$ reads

$$\mathbf{T}_\mu u(x) = \sum_{y=1}^{n} \underbrace{\frac{1}{\mu(x)} P(y,x)}_{=P^*(y,x)}\, u(y)\,\mu(y),$$

Note that P^* is no longer a stochastic matrix. Furthermore, consider a second transfer operator $\tilde{\mathbf{T}}_\mu$ with associated transition matrix $\tilde{P}$ and with the same invariant measure μ. The natural measure for the difference between $\mathbf{T}_\mu$ and $\tilde{\mathbf{T}}_\mu$ is the Frobenius norm in $\ell^2_\mu(\Omega)$,

namely

$$\|\mathbf{T}_\mu - \tilde{\mathbf{T}}_\mu\|_F^2 = \sum_{x,y=1}^{n} \left| P^*(y,x) - \tilde{P}^*(y,x) \right|^2 \mu(x)\mu(y)$$
$$= \sum_{x,y=1}^{n} \left| P(y,x) - \tilde{P}(y,x) \right|^2 \frac{\mu(y)}{\mu(x)}.$$

In this situation, Theorem 3 of [**?**] applies and yields:

Corollary 7.1. *The above assumptions on $\boldsymbol{T}_\mu$, $\tilde{\boldsymbol{T}}_\mu$ imply that there exist enumerations $\{\lambda_i\}$, and $\{\nu_i\}$ of the eigenvalues of $\boldsymbol{T}_\mu$ and $\tilde{\boldsymbol{T}}_\mu$, in ℓ^2_μ, or P and $\tilde{P}$, in the original (unweighted) space $\ell^2(\Omega)$ respectively, such that*

$$\sum_{i=1}^{\infty} |\lambda_i - \nu_i|^2 \leq \|\boldsymbol{T}_\mu - \tilde{\boldsymbol{T}}_\mu\|_F^2.$$

7.2. **Proof of (39).**
We have

$$S(x,y) = \sum_{i=1}^{m} W(x,i)W(y,i) = 0 \quad \Leftrightarrow \quad W(x,i)W(y,i) = 0 \qquad \forall i = 1,..,m, \tag{45}$$

because $W(x,i) \geq 0\ \forall x,i$. Since

$$V(x,i)V(y,i) = \frac{\sqrt{\mu(x)\mu(y)}}{\hat{\mu}(i)} W(x,i)W(y,i)$$

we have that $WW^T = 0$ *iff* $VV^T = 0$, and from (45) this implies that

$$S(x,y) = 0 \quad \Leftrightarrow \quad (VV^T)(x,y) = 0.$$

On the other hand

$$VV^T = VIdV^T = VAA^TV^T = V_A V_A^T,$$

which proves that $S(x,y) = 0$ *iff* $S_A(x,y) = 0$.

REFERENCES

[1] W. E, T. Li, and E. Vanden-Eijnden, Optimal partition and effective dynamics of complex networks, PNAS, 105, pp. 7907-7912, 2008

[2] T. Li, J. Liu and W. E, A probabilistic framework for network partition, submitted.

[3] A. Jain and R. Dubes. Algorithms for Clustering Data. Prentice-Hall, 1988.

[4] J. A. Hartigan and M. A. Wong. A K-Means Clustering Algorithm. Applied Statistics Vol. 28 (1): 100-108, 1979.

[5] P. Deuflhard, M. Weber. Robust Perron Cluster Analysis in Conformation Dynamics. In: M. Dellnitz, S. Kirkland, M. Neumann and C. Schütte (eds.), Linear Algebra and Its Applications: Special Issue on Matrices and Mathematical Biology, Vol. 398C:161-184, 2004.

[6] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. Lin. Alg. Appl., 315:3959, 2000.

[7] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E, vol. 69 (026113), 2004.

[8] G.McLachlan and K.E. Basford. Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York, Basel, 1988.

[9] G.McLachlan and D. Peel. Finite mixture models. Wiley, New York, 2000.

[10] F. V. Jensen. Bayesian Networks and Decision Graphs. Springer, 2001.

[11] F. Höppner, F. Klawonn, R. Kruse and T. Runkler. Fuzzy cluster analysis. John Wiley and Sons, New York, 1999.

[12] S. Kube, M. Weber. A Coarse Graining Method for the Identification of Transition rates between Molecular Conformations. J. Chem. Phys., Vol. 126 (2) , 2007.

[13] M. Weber, W. Rungsarityotin, A. Schliep. Perron Cluster Analysis and Its Connection to Graph Partitioning for Noisy Data. (04-39) ZIB Report, 2004.
[14] J. Shi and J. Mali. Normalized Cuts and Image Segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence, 22, pp. 888–905, 2000.
[15] M. Meila and J. Shi, A Random Walks View of Spectral Segmentation, AI and Statistics (AISTATS), 2001.
[16] M. Belkin and P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, Neural Computation, 6, pp. 1373–1396, 2003.
[17] D.L. Donoho and C. Grimes, Hessian Eigenmaps: New Locally Linear Embedding Techniques for High-Dimensional Data, Proc. Nat. Acad. Sci. USA, 100, pp. 5591–5596, 2003.
[18] S. Lafon and A. B. Lee, Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28, pp. 1393–1403, 2006.
[19] P.G. Doyle and J. L. Snell, Random Walks and Electric Networks, arXiv:math/0001057v1, 2000.
[20] A. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing 2005.
[21] P. R. Östergard. A fast algorithm for the maximum clique problem. Discrete Appl. Math. 120, 1-3, 197-207, 2002.
[22] I. M. Bomze, M. Budinich, P. M. Pardalos and M. Pelillo. The maximum clique problem. Handbook of Combinatorial Optimization, 1-74, Kluwer Academic Publishers, 1999.
[23] L. Elsner and S. Friedland, Variation of the Discrete Eigenvalues of Normal Operators, Proceedings of the American Mathematical Society, Vol 123, No 8, pp 2511-2517, 1995.