Aggregation of Graph Models and Markov Chains by Deterministic Annealing

Yunwen Xu, Srinivasa M. Salapaka, and Carolyn L. Beck

***Abstract*—We consider the problem of simplifying *large weighted directed graphs* by aggregating nodes and edges. This problem is recast as a clustering/resource allocation problem, and a solution method that incorporates features of the deterministic annealing (DA) algorithm is proposed. The novelty in our method is a quantitive measure of *dissimilarity* that allows us to compare *directed* graphs of possibly different sizes (i.e., the original and the aggregated graphs). The approach we propose is insensitive to initial conditions and less likely to converge to poor local minima than Lloyd-type algorithms. We apply our graph-aggregation (clustering) method to Markov chains, where low-order Markov chains that approximate high-order chains are obtained through appropriate aggregation of state transition matrices. We further develop a decentralized computational scheme to improve tractability of the algorithm.**

***Index Terms*—Deterministic annealing (DA) algorithm.**

I. Introduction

The need for graph models that characterize coarser interactions between subsystems within a larger interconnected system arise in various applications, such as in neuroscience studies of functional relationships in the brain, in coordination of multi-agent systems and in networked dynamical systems. Both physical modeling and data-based modeling methods typically yield large models with numerous nodes and complex interactions represented by edges; this makes the analysis of fundamental system behavior intractable. Therefore, to identify the dominant or ensemble interactions of a system, it is often necessary to have a simple representative graph that reflects the core structures. The Markov chain reduction (aggregation) problem is an important special case of the graph-simplification problem, which by itself represents a large class of applications areas. In general, mathematical formulations of graph-simplification problems lead to NP-hard problems [1]. Many formulations pose combinatorial optimization problems, whose cost surfaces typically comprise of many local minima [2], [3].

Existing graph-simplification approaches have primarily focused on graph partitioning, where subgraphs are identified within a graph based on how strongly or weakly they are connected to the rest of the graph [4]. Most graph partition algorithms use "cut-based" methods, which typically require computation of eigenvalues and eigenvectors of the large adjacency or Laplacian matrices associated with the graphs [5], [6]. These methods become increasingly intractable as the size of graph grows, since the corresponding spectral decomposition becomes computationally challenging. Moreover, cut-based algorithms produce a series of bipartitions which do not always lead to representations of actual clusters.

In comparison there are significantly fewer results on simplifying graphs through aggregating nodes, which use data clustering approaches. In fact, there are algorithms that convert data clustering problems into graph partitioning problems and apply spectral methods; only a few address the opposite direction [7]. Among those that apply data clustering methods, most are based on k-means clustering methods [4], [8]. These methods typically get trapped in poor local minima, and therefore require many implementation runs. Some methods use simulated annealing to avoid poor minima, but they require very large computation times [3], [8], [9].

The Markov chain aggregation problem has been studied using specific tools. Approaches include singular perturbation methods [10], [11] and simulation-based aggregation methods [12]. A general discussion of model reduction methods for Markov chains can be found in [13].

In this technical note, we provide a method for obtaining a coarse graph by aggregating nodes that are similar in terms of their connectivity to the rest of the graph. Our contributions are:

1) Definition of a general dissimilarity function based on edge-weight matrices for comparing two graphs, possibly of different sizes, to *systematically* evaluate aggregation performance. This definition of dissimilarity is motivated by many applications including those listed in the beginning of this section. This function captures the notion that two nodes should be aggregated if their corresponding rows in the edge-weight matrix are close under some distance measure. The main challenge in such a formulation arises from the difference in the dimensions of the original and simplified graphs.
2) Reformulation of the graph reduction problem in an optimal *resource allocation* framework, incorporating soft partitioning and the deterministic annealing (DA) algorithm. Specifically, we consider aggregation of general weighted directed graphs using a clustering approach, where each node of the graph is assigned a representative *supernode*. The edge weights between supernodes that comprise the aggregated graph are obtained by minimizing the dissimilarity between the original and the aggregated graphs. We adapt the DA algorithm to solve this optimization problem. The DA algorithm is an iterative method, characterized by an annealing parameter, which detects underlying clusters hierarchically and finer (possibly multiple) subclusters are identified as the algorithm progresses. The number of iterations is small as the annealing parameter is typically increased geometrically. This algorithm is insensitive to initialization conditions and is designed to avoid poor local minima [3]. We also present a guided decentralization modification to improve algorithm scalability by gradually exploiting localization information; this modification allows for a user-selected trade-off in computational time versus the deviation from the centralized solution.

Manuscript received February 7, 2013; revised February 10, 2013 and November 1, 2013; accepted February 18, 2014. Date of publication April 23, 2014; date of current version September 18, 2014. This work was supported in part by the National Science Foundation (NSF) grant CMMI-1100257. Recommended by Associate Editor C. Szepesvári.

Y. Xu and C. L. Beck are with the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: xu27@illinois.edu; beck3@illinois.edu).

S. M. Salapaka is with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: salapaka@illinois.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2014.2319473

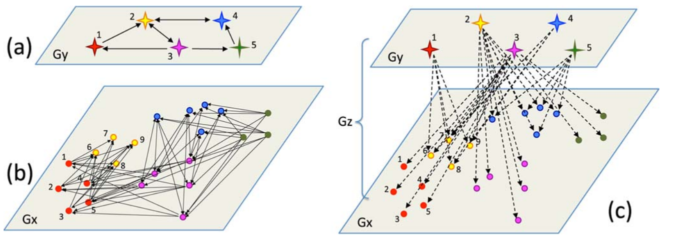

Fig. 1. The relationship of the original graph, $\mathscr{G}_x = (V_x, E_x, X)$ in (b), the reduced graph, $\mathscr{G}_y = (V_y, E_y, Y)$ in (a), and one composite graph, $\mathscr{G}_z = (V_z, E_z, Z) \in \mathbf{G}_{xy}$ in (c). The edges of $\mathscr{G}_x$ are shown by solid arrows in (b), assuming all edges have unit weight; for example, the outgoing vector of the first node is $x(1) = [0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 1, 0, \ldots, 0]$. The edges of $\mathscr{G}_y$ are shown in (a), for example, let the outgoing vector of the first and second supernodes be $y(1) = [0,\ 4,\ 0,\ 0,\ 0]$ and $y(2) = [0,\ 0,\ 5,\ 5,\ 0]$. By Definition 1, $\mathscr{G}_z$ contains all nodes (stars and dots) from $\mathscr{G}_x$ and $\mathscr{G}_y$, and edges that initiate from supernodes and terminate at regular nodes (all dashed arrows), moreover, the supernodes partition the regular nodes, as shown with different colors. For this particular $\mathscr{G}_z$, let $\phi^{-1}(1) = \{1,\ 2,\ 3,\ 4,\ 5\}$, $\phi^{-1}(2) = \{6,\ 7,\ 8,\ 9\}$, which are indicated in red and yellow, the weighting matrix Z satisfies (2) in Definition 1 (iii), that is, $Z_{16} + Z_{17} + Z_{18} + Z_{19} = 4 = Y_{12}$ and $Z_{11} + Z_{12} + Z_{13} + Z_{14} + Z_{15} = 0 = Y_{11}$. Then either $\hat{z}(1) = [0, \ldots, 0,\ 1,\ 1,\ 1,\ 1]$ or $\tilde{z}(1) = [0, \ldots, 0,\ 0.5,\ 1.5,\ 1,\ 1]$ is a valid outgoing vector for $z(1)$, but for this partition, setting $z(1) = \hat{z}(1)$ gives a smaller $\rho(\mathscr{G}_x, \mathscr{G}_z)$ value than setting $z(1) = \tilde{z}(1)$.

Simulation results show our method significantly outperforms spectral based algorithms.

3) Application of our method to Markov chain aggregation problems. We interpret the state transition matrix as an edge-weight matrix and use the Kullback-Leibler (K-L) divergence as a distance measure in our dissimilarity function.

We note that dissimilarity functions have been pursued in [12] and [14]. In particular, [12] focuses on comparing two Markov chains with stationary distributions, using a *lift* operation. General probability distributions with different (finite) cardinalities are considered in [14].

II. Problem Formulation

A. Mathematical Formulation

Consider a *directed weighted graph* $\mathscr{G} = (V, E, W)$ with $|V| = N$ nodes, in which $V, E \subset V \times V$ and $W \in \mathbb{R}_+^{N \times N}$ denote the set of *nodes*, set of *edges* and the *edge-weight matrix*, respectively. Assume for all (i, j) pairs, the edge weights $w_{ij} < B < \infty$. Define the *outgoing vector* of the ith node by the weights on its outgoing edges, that is $w(i) \triangleq [w_{i1}, w_{i2}, \cdots, w_{iN}]$. Here $w_{ij} = 0$ if and only if there is no (directed) edge from the ith node to the jth node. Let $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_N\}$ be the weighting vector of all nodes, satisfying $\mu_i \geq \delta > 0$, and $\sum_i \mu_i = 1$. If not explicitly specified, the node weights are assumed to be equal.

The graph aggregation problem can be stated as: *Given a directed graph $\mathscr{G}_x = (V_x, E_x, X)$ with N nodes, and the weighting matrix $X = [x(1)^T x(2)^T \cdots x(N)^T]^T \in \mathbb{R}_+^{N \times N}$, find a smaller graph $\mathscr{G}_y = (V_y, E_y, Y)$ with $M (< N)$* supernodes *and weighting matrix $Y \in \mathbb{R}_+^{M \times M}$, such that $\mathscr{G}_y$ provides a coarse representation of the original graph $\mathscr{G}_x$*. Our goal is to *define* and *minimize* a *dissimilarity* function $\nu(\mathscr{G}_x, \mathscr{G}_y)$ over graphs of different sizes, in particular for $\mathscr{G}_y$ smaller than $\mathscr{G}_x$. A pictorial illustration is given in Fig. 1.

B. Dissimilarity Measure

The outgoing vectors characterize the influence from one node to the others, and form the basis for comparison between two nodes. If $\mathscr{G}_x$ and $\mathscr{G}_y$ are of the same size, the dissimilarity of node $i \in \mathscr{G}_x$ and node $j \in \mathscr{G}_y$ can be computed in terms of $d(x(i), y(j))$ for some vector distance function $d(\cdot, \cdot) : \mathbb{R}_+^N \times \mathbb{R}_+^N \to \mathbb{R}_+$, since $x(i)$ and $y(j)$ are of the same length. However, if $\mathscr{G}_x$ and $\mathscr{G}_y$ are of different sizes, the outgoing vectors $x(i)$ and $y(j)$ are of different lengths. We therefore propose the notion of *composite graphs*, derived from $\mathscr{G}_x$ and $\mathscr{G}_y$, to make such vector comparisons possible.

Let $\mathscr{N} = \{1, 2, \ldots, N\}$ and $\mathscr{M} = \{1, 2, \ldots, M\}$ be index sets, $N \geq M \geq 2$. A *partition function* ϕ is a map from $\mathscr{N}$ *onto* $\mathscr{M}$, such that $\phi^{-1}(\mathscr{M})$ is a partition of $\mathscr{N}$. That is, for any $1 \leq j \neq k \leq M$, $\phi^{-1}(j) \subset \mathscr{N}$ is non-empty, $\phi^{-1}(j) \bigcap \phi^{-1}(k) = \emptyset$ and $\bigcup_{j=1}^{M} \phi^{-1}(j) = \mathscr{N}$. Each partition function ϕ defines an *aggregation matrix* $\Phi = [\phi_{ij}] \in \{0, 1\}^{N \times M}$ as

$$\phi_{ij} = \begin{cases} 1 & \text{if } i \in \phi^{-1}(j), \\ 0 & \text{if } i \notin \phi^{-1}(j). \end{cases} \tag{1}$$

Therefore, we have $\Phi \mathbf{1}_M = \mathbf{1}_N$, and the kth column of Φ equals $\sum_{i \in \phi^{-1}(k)} e_i$, where $\mathbf{1}_N$ denotes an all-one vector of dimension N, and e_i denotes a vector in $\mathbb{R}^N$ with all 0 elements excepting a 1 in the ith coordinate.

Definition 1 (Composite Graph Set): Given two graphs $\mathscr{G}_x(V_x, E_x, X)$ with N nodes and $\mathscr{G}_y(V_y, E_y, Y)$ with M nodes $(M < N)$, the *composite graph set* associated with $\mathscr{G}_x$ and $\mathscr{G}_y$ is defined as $\mathbf{G}_{xy} \triangleq \{\mathscr{G}_z^\phi(V_z, E_z, Z)\}$, such that each *composite graph* $\mathscr{G}_z^\phi(V_z, E_z, Z) \in \mathbf{G}_{xy}$ along with a partition $\phi : \mathscr{N} \to \mathscr{M}$ satisfy the following conditions:

(i) The node set $V_z = V_y \cup V_x$ is the union of all nodes in $\mathscr{G}_x$ and $\mathscr{G}_y$. Moreover, for notational simplicity, V_z is indexed in an order such that the first M nodes are from $\mathscr{G}_y$ and the remaining N nodes are from $\mathscr{G}_x$.

(ii) The edges in $\mathscr{G}_z^\phi$ originate from supernodes in $\mathscr{G}_y$ and end in regular nodes in $\mathscr{G}_x$, that is, $E_z \subset V_y \times V_x$. Therefore, although $\mathscr{G}_z^\phi$ has $M + N$ nodes, we can represent its weighting matrix by an M-by-N matrix, $Z = [z(1)^T\ z(2)^T \cdots z(M)^T]^T$, with the outgoing vector $z(j) = [z_{j1},\ z_{j2}, \ldots, z_{jN}] \in \mathbb{R}^N$.

(iii) The partition function ϕ provides an aggregative relation between the edge weights of $\mathscr{G}_y$ and $\mathscr{G}_z$ given by $y_{jk} = \sum_{i \in \phi^{-1}(k)} z_{ji}$, for all j, k, which can be compactly expressed by adaptation of the aggregation matrix Φ defined in (1) by

$$Y = Z\Phi. \tag{2}$$

By construction, the outgoing vectors of any $\mathscr{G}_z^\phi \in \mathbf{G}_{xy}$ are the same dimension as those of $\mathscr{G}_x$ (see Fig. 1). Thus we define a distance matrix $D(X, Z) = [d_{ij}] \in \mathbb{R}^{N \times M}$ as

$$d_{ij} = d\left(x(i), z(j)\right)$$

with $d(\cdot, \cdot)$ being a convex function that measures the vector distance. The form of $d(\cdot, \cdot)$ is case-specific, for example, the Euclidean distance and the K-L divergence, respectively, can be choices for geographical graphs and Markov chains. Then the dissimilarity of $\mathscr{G}_x$ and any $\mathscr{G}_z^\phi$ can be defined as the weighted average distance between the corresponding outgoing vectors assigned by partition ϕ, given as

$$\rho(\mathscr{G}_x, \mathscr{G}_z^\phi) \triangleq \sum_{i=1}^{N} \mu_i d\left(x(i), z\left(\phi(i)\right)\right) = trace\left(\Lambda D \Phi^T\right) \tag{3}$$

in which $\Lambda = \mathrm{diag}(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$. We will see later that this formulation gives a novel resource allocation perspective to the graph aggregation problem. We define the *dissimilarity* between $\mathscr{G}_x$ and $\mathscr{G}_y$ by the minimum distance achieved over all composite graphs, that is

$$\nu(\mathscr{G}_x, \mathscr{G}_y) \triangleq \min\left\{\rho\left(\mathscr{G}_x, \mathscr{G}_z^\phi\right) | \mathscr{G}_z^\phi \in \mathbf{G}_{xy}\right\} \tag{4}$$

in particular, this is the minimum over all aggregation matrices Φ and weight matrices Z. This minimum exists since the number of possible partitions ϕ is finite (although combinatorial), and $\rho(\mathscr{G}_x, \mathscr{G}_z^{\phi})$ is convex in the rows of Z for each ϕ.

With this dissimilarity measure, our objective of finding a M-node aggregated graph for $\mathscr{G}_x$ becomes solving the following optimization problem:

$$\operatorname*{arg\,min}_{\Phi, Y:|V_y|=M} \; \nu(\mathscr{G}_x, \mathscr{G}_y), \tag{5}$$

$$\text{s.t.} \quad \Phi \in \{0,1\}^{N\times M}, \Phi \mathbf{1}_M = \mathbf{1}_N, \; Y = Z\Phi. \tag{6}$$

This problem is NP-hard [1], partially resulting from the constraint that Φ belongs to $\{0,1\}^{N\times M}$; thus we aim to approximate the optimal $\Phi^\star$ and $Y^\star$.

III. Problem Solution

A. Data-Clustering Formulation

We decompose the optimization problem (5) into two stages:

(I) Node clustering: Solve $\arg\min_{Z,\Phi} \rho(\mathscr{G}_x, \mathscr{G}_z^{\phi})$, where $\mathscr{G}_z^{\phi} \in \{\mathbf{G}_{xy} : |V_y| = M\}$.

(II) Edge aggregation: Obtain $Y^\star$ (and therefore $\mathscr{G}_y^\star$) from (2) using $Z^\star$ and $\Phi^\star$ from step (I).

The optimization problem in step (I) can be viewed as a resource allocation problem in which the set of N nodes in the given graph $\mathscr{G}_x$ is partitioned into M cells; to each cell a representative supernode is to be allocated such that the averaged pairwise distance between a node and its representative supernode (3) is minimized. Equivalently we want to partition the set of N outgoing vectors $\{x(i)\}_{i=1}^N$ into M cells and to each cell allocate a representative outgoing vector $z(j)$ (the jth row of the weight matrix Z) such that the total representation error $\rho(\mathscr{G}_x, \mathscr{G}_z^{\phi})$ is minimized. As indicated, we seek an approximation of the optimal $\Phi^\star$ and $Z^\star$ by considering this problem with a relaxed version of the integer constraint (6). Specifically, we adapt the DA algorithm [3], [15] to address step (I).

The main idea of the DA algorithm is to incorporate *soft partitioning*, that is, instead of using a partition function ϕ that defines a binary aggregation matrix $\Phi \in \{0,1\}^{N\times M}$ as in (1), each node i $(1 \le i \le N)$ is associated with *all* supernodes j $(1 \le j \le M)$ via *nonnegative association weights* $p_{j|i}$. We assume $\sum_{j=1}^M p_{j|i} = 1$ for all i, and define a *soft aggregation matrix* $P_c \in [0,1]^{N\times M}$, with $[P_c]_{ij} = p_{j|i}, i = 1, \ldots, N, j = 1, \ldots, M$. Then, we modify the dissimilarity functions in (3) and (4) as

$$\hat{\rho}\left(\mathscr{G}_x, \mathscr{G}_z^{P_c}\right) \triangleq \sum_{i=1}^N \mu_i \sum_{j=1}^M p_{j|i} d\left(x(i), z(j)\right) = trace\left(\Lambda D P_c^T\right), \tag{7}$$

$$\hat{\nu}(\mathscr{G}_x, \mathscr{G}_y) \triangleq \min \left\{ \hat{\rho}\left(\mathscr{G}_x, \mathscr{G}_z^{P_c}\right) | Y = Z P_c, P_c \in [0,1]^{N\times M} \right\}. \tag{8}$$

The association weights $\{p_{j|i}\}$ are determined by minimizing $\hat{\rho}(\mathscr{G}_x, \mathscr{G}_z^{P_c})$ under the following weighted entropy constraint:

$$H(\mathscr{G}_z|\mathscr{G}_x) = \boldsymbol{\mu}^T \mathbf{h} = \sum_{i=1}^N \mu_i h_i = H_0 \tag{9}$$

where $\mathbf{h} = [h_1, \ldots, h_N] \in \mathbb{R}_+^N$ is defined by

$$h_i = H\left(\mathbf{p}_c^j\right) = -\sum_{k=1}^M p_{k|i} \log p_{k|i}$$

with $H(\cdot)$ being the Shannon entropy, and $\mathbf{p}_c^j = [p_{1|j}, \ldots, p_{M|j}]$ being the jth row of the soft aggregation matrix P_c. In short, we modify step (I) to solve a continuous relaxation of (5)

$$\operatorname*{arg\,min}_{P_c, Z:|V_y|=M} \; \hat{\rho}\left(\mathscr{G}_x, \mathscr{G}_z^{P_c}\right) = trace\left(\Lambda D P_c^T\right), \tag{10}$$

$$\text{s.t. } P_c \in [0,1]^{N\times M}, P_c \mathbf{1}_M = \mathbf{1}_N, Y = Z P_c, \; H(\mathscr{G}_z|\mathscr{G}_x) = H_0 \tag{11}$$

for a feasible value of H_0 $(0 \le H_0 \le \log M)$. This is solved by minimizing the Lagrangian $F(\mathscr{G}_x, \mathscr{G}_z^{P_c}) \triangleq \hat{\rho}(\mathscr{G}_x, \mathscr{G}_z^{P_c}) - (1/\beta) H(\mathscr{G}_z^{P_c}|\mathscr{G}_x)$ with respect to $\{p_{j|i}\}$, where $1/\beta$ is a Lagrange multiplier. This yields a *Gibbs distribution*

$$p_{j|i} = \frac{\exp\{-\beta d_{ij}\}}{\sum_{k=1}^M \exp\{-\beta d_{ik}\}}. \tag{12}$$

Remark 1: Note that each value of the parameter β corresponds to a value of H_0 (obtained by substituting (12) in (9)). In [3], [16], it is shown that the larger the value β, the smaller the value of the corresponding H_0. In the DA algorithm, the relaxed problem (10) is repeatedly solved with increasing values of parameter β, i.e., decreasing (yet feasible) values of H_0.

Substituting the association weights (12) into (7) and (9), the Lagrangian F becomes

$$F^\star\left(\mathscr{G}_x, \mathscr{G}_z^{P_c}\right) = -\frac{1}{\beta} \sum_{i=1}^N \mu_i \log \sum_{k=1}^M \exp\{-\beta d_{ik}\}. \tag{13}$$

At each iteration of the DA algorithm, the parameter β is fixed and a local minimum of (13) is computed. That is, the representative outgoing vectors $z^\star(j)$ are computed using the following implicit equation:

$$0 = \nabla_{z(j)} F^\star\left(\mathscr{G}_x, \mathscr{G}_z^{P_c}\right) = \sum_{i=1}^N \mu_i p_{j|i} \nabla_{z(j)} d_{ij}. \tag{14}$$

Equation (14) is solved using gradient descent methods where the solutions from the previous iteration are used as starting values in the current iteration. These computations are repeated as the parameter β is increased (this is referred to as annealing). The rationale behind the annealing process is as follows. Note that for $\beta \approx 0$, minimizing the cost function F is approximately the same as minimizing $-H$, which is convex and has a global minimum. In fact in this case, the association weights given by (12) are approximately uniform (i.e., $p_{j|i} \approx 1/M, \;\forall i, j$), so all outgoing vectors $z^\star(j)$ are coincident, thus there is a single distinct supernode. As β is increased, the soft aggregation matrix P_c becomes more and more binary, moreover, the annealing process exhibits a series of *phase transitions* as shown in [3], [15], where the solutions $\{z^\star(j)\}$ are insensitive to changes in β except at critical values β_c; the number of *distinct* outgoing vectors in the composite graph increases at these critical values. When β is very large, $F \approx \hat{\rho} \approx \rho$ (since $p_{j|i} \approx 1$ if $j = \arg\min_k d_{ik}$ and otherwise is $\approx$0), and thus we recover a hard partition and the original dissimilarity function. The underlying heuristic of the DA algorithm is that it finds the global minimum of the Lagrangian at very small β and tracks the minimum as β is increased.

After obtaining the weighting matrix $Z^\star$ from step (I) as above, we can determine the weighting matrix $Y^\star$ by soft edge aggregation as in step (II): $Y^\star = Z^\star P_c$.

The key aspects of the graph reduction (aggregation) process described in the preceding and the DA algorithm are summarized in *Lemma 1* below. For clearer exposition, we choose d to be the squared Euclidean distance function.

Lemma 1: Let $d(u,v) \triangleq \|u - v\|_2^2$, then: (i) the weighting matrix $Z^\star$ of $\mathscr{G}_z^\star$ that satisfies the first-order optimality condition in (14) for a given β and the corresponding weighting matrix $Y^\star$ for $\mathscr{G}_y^\star$ are given by

$$Z^\star = Q_c^T X \text{ and } Y^\star = Q_c^T X P_c \tag{15}$$

where $P_c, Q_c \in [0,1]^{N\times M}$, $[P_c]_{ij} = p_{j|i} = e^{-\beta d(x(i), z^\star(j))} / (\sum_k e^{-\beta d(x(i), z^\star(k))})$ and $[Q_c]_{ij} = q_{i|j} = (p_r p_{k|r}) / (\sum_k p_r p_{k|r})$, for $i=1,\ldots,N$, and $j=1,\ldots,M$; (ii) the number of distinct outgoing vectors increases when β surpasses a critical value β_c, at which the determinant of the Hessian, $\det(\nabla^2_{z^\star(j_0)} F^\star(\mathscr{G}_x,\mathscr{G}_z^{P_c}))=0$ for some $1\le j_0\le M$. Moreover, β_c^{-1} is given by twice the maximum eigenvalue of the matrix given by $\sum_{i=1}^N q_{i|j}(x(i)-z^\star(j_0))(x(i)-z^\star(j_0))^T$.

Remark 2: Lemma 1 is from [3]. Part (i) is obtained by solving (14) with the squared Euclidean distance. Part (ii) is a direct consequence of the DA algorithm and its properties (see [3], [15] for details). It should be noted that the solution $Z^\star$ in (15) is insensitive to changes in β between two successive critical values of β_c (see [15] for quantitative details). As β is increased beyond a critical β_c, the number of *distinct* solutions $z^\star(j)$ to (14) increases. Thus for graph reduction, we update P_c, Q_c, $Z^\star$, and $Y^\star$ by (15) for each β value as β is increased, and stop when the number of *distinct* outgoing vectors $z^\star(j)$ (or the rows of $Z^\star$) equals M.

B. Markov Chain Reduction

A discrete Markov chain with finite state space $\mathscr{X}=\{\mathscr{X}_1,\mathscr{X}_2,\cdots\}$ and transition matrix Π (with ith row $\pi(i)$ and $[\Pi]_{i,j}=\pi_{ij}$) can be represented by a graph $\mathscr{G}_\Pi(V_\Pi,E_\Pi,\Pi)$. When the state space $|\mathscr{X}|=N$ is large, we aim to find a low-order Markov chain $\mathscr{Y}$ with M states and transition matrix Ψ (with the jth row being $\psi(j)$ and $[\Psi]_{i,j}=[\psi_{ij}]$) to represent $\mathscr{X}$, where representativeness is defined through their graphs. That is, our two-step framework for graph reduction applies here by aggregating similar states, such that the dissimilarity between their corresponding graphs $\mathscr{G}_\Pi$ and $\mathscr{G}_\Psi(V_\Psi,E_\Psi,\Psi)$ given by $\nu(\mathscr{G}_\Pi,\mathscr{G}_\Psi)$ in (5), is minimized. After applying soft aggregation, the optimization problem associated with the first step is given by (10) with constraints

$$Z\ge 0,\ Z\mathbf{1}_N=\mathbf{1}_M \tag{16}$$

in addition to (11). In this setting, we choose the K-L divergence for $d(\cdot,\cdot)$, that is $d_{ij}=d(\pi(i),z(j))=\sum_{k=1}^N \pi_{ik}\log(\pi_{ik}/z_{jk})$, which is convex, and we assume that the support set of $\pi(i)$ is contained in the support set of $z(j)$ for all i and j (i.e., $z_{jk}=0\Rightarrow \pi_{ik}=0,\ \forall\, i,j,k$), and $H_0\in[0,\log M]$. The constraint (16) defines transition probabilities from a *superstate* in $\mathscr{Y}$ to the original states in $\mathscr{X}$, and also guarantees that $\Psi^\star$ computed by ZP_c is a valid Markov transition matrix.

Remark 3: If the original Markov chain $\mathscr{X}$ has a limiting distribution $\xi\in\mathbb{R}_+^{1\times N}$ satisfying $\xi=\xi\mathbf{\Pi}$ and $\sum_{i=1}^N \xi_i=1$ (e.g., is an irreducible and aperiodic Markov chain), a natural choice of the node weights is $\mu_i\overset{\Delta}{=}\xi_i$ for all i. In this case the resulting dissimilarity function $\hat{\nu}(\mathscr{G}_\Pi,\mathscr{G}_\Psi)\overset{\Delta}{=}\min_{\mathbf{\Psi}:|V_\Psi|=M}\hat{\rho}(\mathscr{G}_\Pi,\mathscr{G}_Z^{P_c})$ provides a soft version of the *lifted* K-L divergence rate metric between two Markov chains proposed in [17].

The Lagrangian, after accounting for the Markov constraints (16), becomes $\tilde{F}=\hat{\rho}(\mathscr{G}_\Pi,\mathscr{G}_Z^{P_c})-(1/\beta)H(\mathscr{G}_Z^{P_c}|\mathscr{G}_\Pi)-\sum_{j=1}^M \eta_j\sum_{k=1}^N(Z_{jk}-1)$, with η_j's being the Lagrange multipliers associated with (16). We follow the general two step framework, which leads to results stated in *Lemma 2* and *Theorem 1*.

Lemma 2: Given a Markov chain with N states and transition probability matrix Π, the transition matrix $\Psi^\star$ of the low-order Markov chain calculated from the two step aggregation with $M<N$ superstates is given by $\Psi^\star=Z^\star P_c$, where $Z^\star={Q_c}^T\Pi$, $[P_c]_{il}=p_{l|i}=e^{-\beta d(\pi(i),z^\star(l))}/(\sum_{t=1}^M e^{-\beta d(\pi(i),z^\star(t))})$, $l=1,\ldots,M$, $i=1,\ldots,N$, with $d(\cdot,\cdot)$ being the K-L divergence, and $[Q_c]_{sk}=q_{s|k}=(p_s p_{k|s})/(\sum_t p_t p_{k|t})$, $k=1,\ldots,M$, $s=1,\ldots,N$.

Proof: Since the edge weight constraints in (16) do not depend on the $p_{l|i}$ values, taking $(\partial\tilde{F})/(\partial p_{l|i})=0$ yields the same Gibbs distribution as in (12).

Substituting $p_{l|i}$ into the Lagrangian $\tilde{F}$ to obtain $\tilde{F}^\star$, and setting $\partial\tilde{F}^\star/\partial z_{ki}=0$ for each k and i, we have $\nu_k z^\star_{ki}=\sum_s p_s p_{k|s}\Pi_{si}$. Considering constraints $\sum_{i=1}^N z^\star_{ki}=\sum_{i=1}^N \pi_{si}=1,\ \forall 1\le k\le M$ and $1\le s\le N$, we have $z^\star_{ki}=\sum_{s=1}^N q_{s|k}\Pi_{si}$. Note that all resulting entries of $Z^\star$ are nonnegative, every entry of $\Psi^\star$ is a convex combination of the corresponding column in $Z^\star$, and the resulting $\Psi^\star$ is a nonnegative stochastic matrix. □

The critical value β_c that leads to phase transition is given by β for which the *second variation of* $\tilde{F}^\star$ at $Z^\star$ (see Lemma 2) defined by $\Delta^2\tilde{F}^\star(\beta,W)\overset{\Delta}{=}(\mathrm{d}^2/\mathrm{d}\epsilon^2)\tilde{F}^\star(Z^\star(\beta)+\epsilon W)|_{\epsilon=0}$ becomes nonpositive for some $W\in\mathscr{W}_a$, where $\mathscr{W}_a\overset{\Delta}{=}\{W\in\mathbb{R}^{M\times N}:\sum_k w(k)^T w(k)=1,\ \sum_{j=1}^N W_{ij}=0,\ \text{for all } i\}$ denotes an *admissible perturbation* set. This ensures that when the perturbation $W\ne 0$, the perturbed weighting matrix $\hat{Z}^\star=Z^\star+\epsilon W$ satisfies (16), and the second variation is independent of the size of W, yielding the following result.

Theorem 1: Suppose for some β_0, the matrix $Z^\star(\beta_0)$ in Lemma 2 satisfies $\Delta^2\tilde{F}^\star(\beta_0,W)>0$ for all $W\in\mathscr{W}_a$ and the number of distinct outgoing vectors $m<M$. Then the critical $\beta_c\overset{\Delta}{=}\min\{\beta>\beta_0:\Delta^2\tilde{F}^\star(\beta,W)\not>0$ for some $W\in\mathscr{W}_a\}$ is achieved when $\min_k\lambda_{min}(\Gamma_\beta(k))=0$, where $\Gamma_\beta(k)$ is defined in (17).

Proof: The second variation of $\tilde{F}^\star$ at $Z^\star$ is given by $\Delta^2\tilde{F}^\star(\beta,W)=2\gamma_1(\beta,W)+\gamma_2^2(\beta,W)$ for $W\in\mathscr{W}_a$, where $\gamma_1(\beta,W)=\sum_{k=1}^m q_k w(k)^T\Gamma_\beta(k)w(k)\in\mathbb{R}$, $q_k=\sum_{i=1}^N \mu_i p_{k|i}$, $w(k)$ is the kth row of W

$$\Gamma_\beta(k)=\Lambda_\beta(k)-\beta C_\beta(k),$$
$$\text{where }\Lambda_\beta(k)=\operatorname{diag}\left\{\frac{\left[\sum_{i=1}^N q_{i|k}\pi(i)\right].}{(z^\star(k).^2)}\right\},$$
$$\text{and }C_\beta(k)=\left\{\sum_{i=1}^N q_{i|k}\left[\frac{\pi(i).}{z^\star(k)}\right]\left[\frac{\pi(i).}{z^\star(k)}\right]^T\right\}. \tag{17}$$

$\gamma_2^2(\beta,W)=\beta\sum_{i=1}^N(\sum_{k=1}^M p_{k|i}[\pi(i)./z^\star(k)]^T w(k))^2$ and $[\pi(i)./z(k)]$ denotes the element-wise division (we have assumed $z_{jk}=0\Rightarrow\pi_{ik}=0,\ \forall\, i,j,k$).

Since $\Delta^2\tilde{F}^\star(\beta_0,W)>0$, we have

$$\Delta^2\tilde{F}^\star(\beta,W)\ge\gamma_2^2(\beta,W)+\min_k\{\lambda_{\min}(\Gamma_\beta(k))\}$$
$$\ge\min_k\{\lambda_{\min}(\Gamma_\beta(k))\}\ge 0$$

for $\beta_0\le\beta\le\beta_c$, $W\in\mathscr{W}_a$, where we have used the fact that $\Delta^2\tilde{F}^\star(\beta,W)$ is continuous in its arguments, and have applied the Rayleigh-Ritz inequality $x^T\Gamma_\beta(k)x\ge\lambda_{\min}(\Gamma_\beta(k))\|x\|^2$ for symmetric matrices $\Gamma_\beta(k)$. Therefore $\Delta^2\tilde{F}^\star(\beta_c,W)=0$ for some W only if $\min_k\lambda_{\min}(\Gamma_\beta(k))=0$. Moreover, since $C_\beta(k)$ and $\Lambda_\beta(k)$ are insensitive to β, β_c can be approximated by $\lambda_{\min}(C(k)^{-1}\Lambda(k))$ whenever $C_\beta(k)$ is invertible. □

Remark 4: The perturbation matrix W, s.t., $\Delta^2\tilde{F}^\star(\beta_c,W)=0$ is given by $[0,\ldots,0,w(k_0)^T,-w(k_0)^T,0,\ldots,0]^T$, where $\lambda_{\min}(\Gamma_\beta(k_0))=\min_k\lambda_{\min}(\Gamma_\beta(k))$, and $w(k_0)$ is the corresponding eigenvector of $\Gamma_\beta(k_0)$. This choice is possible since $m<M$ and there are at least 2 rows of matrix $Z^\star$ which are equal to $z^\star(k_0)$. The index k_0 identifies the outgoing vector that is replaced by multiple distinct vectors when (14) is solved for β that surpasses β_c during the annealing process.

TABLE I
DECENTRALIZED MODIFICATION VS. THE ORIGINAL ALGORITHM

40-node graph	Running time (sec)	Free energy value
Centralized computation	10.6366	386.0528
Decentralized modification	5.3401	415.3161

Thus there is an increase in the number of distinct outgoing vectors; this process repeats at subsequent critical values as β is increased in the DA algorithm.

IV. DISCUSSION

By viewing the graph aggregation problem from a resource allocation perspective, we can apply the principles of the DA algorithm to identify an appropriate reduced graph size, use the flexibility in choosing a distance measure to define a dissimilarity function, impose case specific constraints, and avoid poor local minima [3], [15], [18]. Further, the computational cost of solving large eigenvalue/eigenvector problems in spectral-decomposition methods is avoided. Unlike many clustering based reduction methods that conduct repeated bipartitioning (such as the min-flow cut and the simulation based aggregations [12]), hierarchical multiple-partitions result from splitting. In fact, as β tends to infinity, the proposed algorithm mimics Lloyd's algorithm, but with a carefully selected initial guess for the representative points (obtained from the previous annealing steps).

Since the aggregation result is insensitive to the value of β (see Remark 2 and [15]), our algorithm uses a geometric annealing schedule (e.g., $\beta_{k+1} = \alpha\beta_k$ for $\alpha > 1$) and thus requires far fewer iterations than popular annealing algorithms such as simulated annealing (which typically has a logarithmic cooling law [9]). However, the DA algorithm requires *centralized* computations to overcome convergence to local minima, i.e., it uses information from all $x(i)$'s to compute each $z^\star(j)$; in this sense, the computational effort at each iteration is high. We note, however, that as β increases, the association weights given by (12) tend to either 0 for distant node-supernode pairs or 1 for nearby pairs, yielding a natural decentralization of the algorithm. This decentralization can be exploited by replacing the most expensive computation $z^\star(j) = \sum_{i=1}^{N} q_{i|j} x(i)$ in (15) by $\hat{z}^\star(j) = \sum_{i \in C_j} q_{i|j} x(i)$ for an appropriate choice of index set C_j. For instance, we can choose C_j by the Voronoi partition centered at the outgoing vectors $z(j)$, i.e., $C_j \triangleq \{i \in V_x : \min_k d(x(i), z(k)) = d(x(i), z(j))\}$, $1 \le j \le m$. For this selection, it is straightforward to show that a bound on the approximation error $\|z^\star(l) - \hat{z^\star}(l)\|_\infty < \tau$ can be guaranteed for any $\{C_j\}$ that satisfies $\sum_{l \in C_i} \mu_l p_{j|l} < \epsilon_0 = \tau/(M-1)\max_t |x_{it}|$, $\forall i \neq j$. Alternative choices for the cells C_j can be found in [18], [19]. In these methods, the approximation can be improved if the localization is carried out in an adaptive way. Simulation results demonstrate the potential to improve scalability (Table I). Ongoing work to study autonomous schemes to further improve the scalability of algorithm is underway.

V. EXAMPLES AND SIMULATIONS

A. An Illustrative Example of Markov Chain Aggregation

We first discuss aggregating a small Markov chain $\mathscr{X}$ with 3 states solely to demonstrate our framework and the annealing process. Let the transition matrix of $\mathscr{X}$ be

$$\Pi = \begin{bmatrix} 0.97 & 0.01 & 0.02 \\ 0.02 & 0.48 & 0.50 \\ 0.01 & 0.75 & 0.24 \end{bmatrix}$$

and the state weights $\{\mu_i\}$ given by [0.3471, 0.3883, 0.2646], the limiting distribution of $\mathscr{X}$.

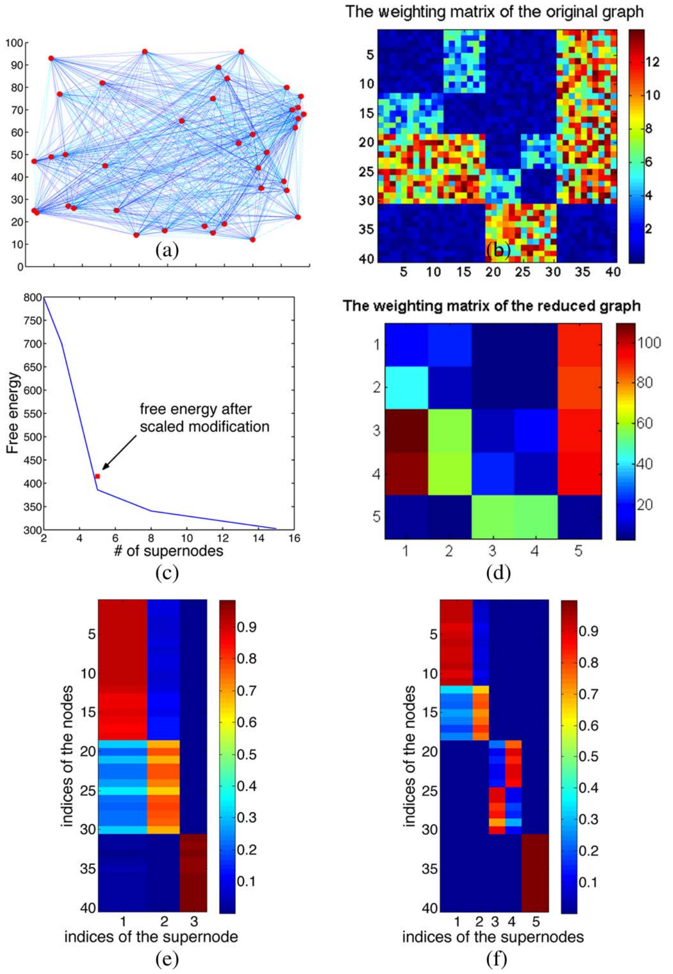

Fig. 2. Multi-scaled reduction results with squared Euclidean distance for a graph of 40 nodes. Plot (a) shows the original directed graph and the locations of all nodes, cyan arrows indicate weak connections (with small edge weights) and blue arrows indicate strong connections. Plot (b) shows the weighting matrix of the original graph. Plot (c) depicts the dissimilarity value versus the number of supernodes, the red dot marks the dissimilarity achieved by the 5-supernode reduced graph after applying decentralization. Plot (d) provides the weighting matrix of the 5-supernode aggregated graph. Plot (e) shows the association weights when there are 3 supernodes, from which we see the influence from remote nodes in determining the $z(1)$ is very small, so we omit the nodes outside the Voronoi cells for supernodes z_1 and z_2 in subsequent calculations. Plot (f) shows the association weights from each node to each supernode when the algorithm terminates as 5 supernodes have been identified.

We use the results from Section III-B to obtain an aggregated Markov chain $\mathscr{Y}$ with 2 superstates. For small $\beta\ (= 0.001)$, the association weights are identical; that is, $p_{j|i} = 0.5$ and therefore $z_{ij} = 1/3$ for $i = 1, 2, 3$ and $j = 1, 2$. The results for β values just beyond the critical $\beta_c = 1.0837$ and a very high value $(\beta = 54.2540)$ are shown below:

β	Z^T	P_c	Ψ
1.084	$\begin{bmatrix} 0.0243 & 0.9539 \\ 0.5845 & 0.0194 \\ 0.3912 & 0.0267 \end{bmatrix}$	$\begin{bmatrix} 0.0165 & 0.9835 \\ 0.9898 & 0.0102 \\ 0.9928 & 0.0072 \end{bmatrix}$	$\begin{bmatrix} 0.9673 & 0.0327 \\ 0.0614 & 0.9386 \end{bmatrix}$
54.25	$\begin{bmatrix} 0.0159 & 0.9700 \\ 0.5894 & 0.0100 \\ 0.3946 & 0.0200 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \\ 1.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.9841 & 0.0159 \\ 0.0300 & 0.9700 \end{bmatrix}$

Note that P_c in both cases indicates that one superstate is representative of the first state in $\mathscr{X}$ while the other superstate represents the

TABLE II
COMPARISON WITH GRAPH SPECTRAL CLUSTERING

n	100	1000	5000	7500	10000
t_{SM} (sec)	0.0401	0.9250	57.1976	178.5284	400.4039
t_{DA} (sec)	0.0521	0.8808	23.038	52.8473	76.344
$(\nu_{DA}/\nu_{SM})\%$	8.14	8.11	9.38	7.86	7.87

remaining two states; note the large β value gives a hard partition. This example also has been considered in [12]; we obtain the same partition result, and in addition provide the superstate weights reflecting the proportion of states being aggregated.

B. A General Graph Reduction

Fig. 2 presents results for an aggregation of a graph with 40 nodes into a graph with 5 supernodes. This example is motivated by neuroscience studies, and the graph (plot (a)) shows the information flows among 40 networked neurons. Specifically, each node stands for a neuron and an arrow indicates the directed information between two neurons. The weighting matrix is shown in plot (b), in which the color of the ith row and jth column represents the edge weight x_{ij} (indicated by the color bar). We select a uniform node weight $\mu_i = 1/40,\ \forall i$, and adopt the squared Euclidean distance for $d(\cdot,\cdot)$ in (3) to compare the functional dissimilarities between neurons. We apply the two-step aggregation method, where for small $\beta = 0.01$, all supernodes are coincident. As β is increased, the number of distinct supernodes increases from 1 to 15 through splitting, and the dissimilarity achieved by the reduced graph decreases (plot (c)). From the dissimilarity curve, we estimate the natural size of the reduced graph to be $M = 5$ clusters. The weighting matrix for the resulting 5-supernode reduced graph is given in plot (d), which is consistent with the way that this test data was constructed.

To demonstrate the decentralization method of Section IV, we form the Voronoi cells for each supernode at each β, and for those cells that satisfy the threshold criterion, the computations are restricted to nodes within the cells as discussed. Plots (e) and (f) show 3 and 5 supernodes at different stages of the annealing process. The running times and dissimilarity, achieved by the original and the modified algorithms to obtain a 5-supernode graph are presented in Table I, which shows a significant reduction in computation time.

We finally compare our algorithm with a standard Normalized Graph Cut algorithm proposed by Shi and Malik [5]. As the size of original graph is increased from 50-node to 10000-node, the times needed to implement spectral clustering (t_{SM}) and our aggregation method (t_{DA}), and the (hard) dissimilarities $\nu(\mathscr{G}_x, \mathscr{G}_y)$ (as in (5)) achieved by both methods are shown in Table II. Note that the times and ratios in Table II represent the average of 5 implementations on the same datasets. Therefore, it is evident that the simplified graph obtained using the DA based algorithm consistently provides a better representativeness $(\nu_{DA}/\nu_{SM} \approx 0.08)$ than Normalized Graph Cut. Also, for very large graphs (roughly more than 1000 nodes), the DA based algorithm is more efficient.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the anonymous reviewers for their helpful suggestions.

REFERENCES

[1] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Mach. Learning*, vol. 75, no. 2, pp. 245–248, 2009.

[2] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, 1st ed. Boston, MA, USA: Kluwer, 1991.

[3] K. Rose, "Deterministic annealing for clustering, compression, classification, regression and related optimization problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210–2239, Nov. 1998.

[4] S. E. Schaeffer, "Graph clustering," *Comp. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, 2007.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[6] X. Wang and I. Davidson, "Flexible constrained spectral clustering," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 563–572, ACM.

[7] I. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1944–1957, Nov. 2007.

[8] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens, "Graph nodes clustering with the sigmoid commute-time kernel: A comparative study," *Data Knowl. Eng.*, vol. 68, no. 3, pp. 338–361, Mar. 2009.

[9] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[10] H. A. Simon and A. Ando, "Aggregation of variables in dynamic systems," *Econometrica*, vol. 29, no. 2, pp. 111–138, 1961.

[11] R. Phillips and P. Kokotovic, "A singular perturbation approach to modeling and control of Markov chains," *IEEE Trans. Autom. Control*, vol. 26, no. 5, pp. 1087–1094, Oct. 1981.

[12] K. Deng, P. Mehta, and S. Meyn, "Optimal Kullback-Leibler aggregation via spectral theory of Markov chains," *IEEE Trans. Autom. Control*, vol. 56, no. 12, pp. 2793–2808, Dec. 2011.

[13] C. Beck, S. Lall, T. Liang, and M. West, "Model reduction, optimal prediction, the Mori-Zwanzig representation of Markov chains," in *Proc. 48th IEEE Conf. Decision Control*, 2009, pp. 3282–3287.

[14] M. Vidyasagar, "A metric between probability distributions on finite sets of different cardinalities and applications to order reduction," *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2464–2477, Oct. 2012.

[15] P. Sharma, S. M. Salapaka, and C. L. Beck, "Entropy-based framework for dynamic coverage and clustering problems," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 135–150, Jan. 2012.

[16] E. T. Jaynes, *Probability Theory—The Logic of Science*. New York: Cambridge Univ. Press, 2003.

[17] Z. Rached, F. Alajaji, and L. Campbell, "The Kullback–Leibler divergence rate between Markov sources," *IEEE Trans. Inform. Theory*, vol. 50, no. 5, pp. 917–921, May 2004.

[18] P. Sharma, S. M. Salapaka, and C. L. Beck, "A scalable approach to combinatorial library design for drug discovery," *J. Chem. Inform. Model.*, vol. 48, no. 1, pp. 27–41, 2008.

[19] A. Kwok and S. Martinez, "A distributed deterministic annealing algorithm for limited-range sensor coverage," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 4, pp. 792–804, Jul. 2011.