2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

On Reduction of Graphs and Markov Chain Models

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

Abstract— **This paper introduces a new method for reducing large directed graphs to simpler graphs with fewer nodes. The reduction is carried out through node and edge aggregation, where the simpler graph is *representative* of the original large graph. Representativeness is measured using a metric defined herein, which is motivated by thermodynamic free energy and vector quantization problems in the data compression literature. The resulting aggregation scheme is largely based on the maximum entropy principle. The proposed algorithm is general in the sense that it can accommodate a large class of functions for characterizing distance between the nodes. As a special case, we show that this method applies to the Markov chain model-reduction problem, providing a *soft-clustering* approach that enables better aggregation of state-transition matrices than existing methods. Simulation results are provided to illustrate the theoretical results.**

I. Introduction

The reduction, or simplification, of graph-based models is critical to the analysis, simulation and control design for systems arising in many diverse areas, such as network routing [1], consensus and cooperation in multi-agent systems [2], image processing [3], statistical learning [4], neuroscience studies of functional relationships in the brain [5], and in distributed control of networked dynamical systems [6], to name a few. Typically, the models for these systems, created from first principles and/or data-based methods, are overly-complicated, rendering the analysis of fundamental system behavior intractable. A common goal in the study of these systems thus is to find a simple mathematical model to represent the behavior of the complex system, such that the resulting coarseness of the simplified model is as negligible as possible. In this paper, we propose a class of graph reduction algorithms aimed at simplifying graph-based models for systems arising in the aforementioned areas. Specifically, we propose a general clustering-based algorithm to reduce the dimensions of the graph. Reduction of Markov chain models is discussed as a special case of our general framework.

Graph-reduction problems, in general, can be formulated as combinatorial optimization problems, where the the objective is to minimize a distance function between the original and aggregated (or reduced) graphs. These problems are computationally complex (NP-hard). The numerical complexity mainly stems from the *combinatorial* nature of the number of ways in which the nodes and edges can be aggregated. The resulting cost functions are non-convex and typically exhibit multiple local minima. In this aspect, graph-reduction closely resembles widely studied data-clustering problems (such as resource allocation) [7]–[11] and therefore can avail tools from the latter. A critical difference lies in the difficulty of formulating a cost function since this requires defining a metric that compares two graphs (the original and the reduced) that are of *different* dimensions. This difficulty is typically overcome by defining the metrics in terms of *intermediate* graphs that are obtained either by *lifting* (where extra nodes and edges are added to the reduced graph [12]) or collapsing (where nodes of the original graph are aggregated), so that the dimensions of graphs become equal. Most of the graph-reduction methods can be broadly classified as either spectral-decomposition or clustering-based approaches. In the spectral-decomposition approach, an *adjacency matrix* comprising the pairwise distances between nodes is formed; the eigenvectors of this matrix are used to identify the underlying node clusters [13]. Moreover, these eigenvalues provide useful analytical information such as convergence rates of the associated algorithms. However, these methods become increasingly intractable as the number of nodes become large since determining the eigenvalues and eigenvectors of the corresponding adjacency matrices becomes computationally difficult. On the other hand, clustering algorithms [14], [15] provide numerically efficient approaches for reduction. In these algorithms, nodes and edges in the graph are aggregated leading to simpler but coarser graphs, for example as in the *kernel k-means* algorithm [16]. In general, these algorithms first determine a partitioning of the nodes into *cells*, and then assign a representative node (or supernode) for each cell, specifying new edges between each pair of cells. However, most of these algorithms provide iterative schemes that achieve certain necessary (but not necessarily sufficient) conditions that the global minimum satisfies. The main disadvantage of the existing algorithms is that they are highly sensitive to the initialization step and typically converge to non-optimal local minima on the cost surface.

Reduction of Markov chains usually appears as an independent class of model reduction problems. In [17], singular perturbation approaches are used to characterize Markov chain models as *completely decomposable* or *nearly completely decomposable models*; the notion of decomposability was first proposed by Ando and Simon in a landmark paper [18]. A survey of additional model reduction methods for these types of models can be found in [19]; this includes spectral methods and optimal prediction techniques. Recently, a simulation-based method has been developed [12], where minimizing the reduction error is posed as the minimization of the Kullback-Leibler (K-L) divergence rate between two stationary Markov chains in the same state space, one of which is a *lifted* version of the reduced Markov chain.

In this paper, we present a method for graph reduction that is motivated by data-clustering algorithms. In the context

This work was partially supported by the NSF grant ECS - 0725708. The authors gratefully acknowledge the support of the NSF through the grant CMMI 1100257.

Yunwen Xu and Carolyn L. Beck are with the department of Industrial and Enterprise Systems Engineering, Srinivasa M. Salapaka is with the department of Mechanical Science and Engineering. They are all in the University of Illinois at Urbana-Champaign, Urbana, Illinois, USA. Email: [xu27, salapaka, beck3]@illinois.edu

of data-clustering, Deterministic Annealing (DA) algorithm [20] developed for codebook design in the context of vector quantization, provides an algorithm that is independent of the initialization step, and is designed to avoid local minima. This algorithm considers analogies to statistical mechanics with entropy maximization as a constraint. In our work, we employ the Maximum Entropy Principle (MEP) and generalize the concepts originated in the DA algorithm to carry out clustering/reduction for directed graphs.

The main contributions of this work are

1. A tractable iterative algorithm that achieves model reduction (aggregation) of large and complex graphs, which is independent of the initialization step and thus avoids poor local optimal solutions.
   (a) The problem formulation and analysis are made under very mild assumptions. The results are therefore applicable to a large class of problems.
   (b) In particular, the algorithm is applicable to a variety of metrics defined on the graphs (Euclidean distance and K-L divergence are used as examples to demonstrate the algorithm).
   (c) The algorithm is applicable to directed graphs unlike many existing algorithms (especially those schemes based on spectral-decomposition that require symmetric adjacency matrices). Specifically, in our algorithms, the weighted adjacency matrices for the graphs are not required to be symmetric.
2. Interpretation of the Markov-Chain reduction problem as a special case of the graph reduction problem, and a resulting algorithm to solve this reduction problem. This algorithm involves a *soft-clustering* approach that seeks a better approximation of the state-transition matrix than mere aggregation of its values as is done in most existing methods.

II. Problem Formulation

A. An illustrative example

Figure 1 (a) represents an inter-neuron causal influence graph, used for analysis in certain neuroscientific studies. The nodes represent neurons and the directed edges represent the *causal influences* of one neuron on another. These graphs are obtained by applying statistical analysis tools to the neurons' spike train data set. A neuron A is said to causally influence a neuron B whenever knowing the spiking history of A helps to better predict future spiking activity of B. These graphs are crucial in understanding the causal relationships between different neurons and therefore infer the information propagation in brain when a human performs some activity. Analysis of these inter-neuron relationships become increasingly complicated as the number of neurons increases. Thus it is desirable to construct a smaller graph (see Figure 1 (b)), in which we combine the neurons that have similar behavior. The result is a new graph based on functional blocks rather than individual nodes (denoted by stars in (b)). We combine the influence relations between individual neurons and re-compute the influence from one block to another (denoted by bold arrows in (b)) to obtain the reduced graph. Note that the red star has bi-directional influence with the green star, and it influences the blue star; the reduced graph inherits some measure of causal influences between the respective functional blocks.

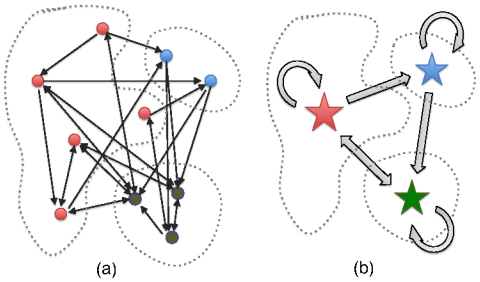

Fig. 1. The original (left) and reduced (right) graph models for a network of 10 neurons. The reduced model has 3 supernodes denoted by stars (functional blocks) and bold arrows indicating causal relations among these blocks. Nodes with same color in the original and reduced graphs have similar behavior.

B. Mathematical formulation

A *directed weighted graph* is represented by the triple $\mathcal{G} = (V, E, W)$, consisting of a set of *nodes* V, a set of *edges* (or *arrows*) $E \subset V \times V$, and a weighting matrix $W \in \mathbb{R}_+^{|V|\times|V|}$, where the ith row $w(i) = [W_{i1}, W_{i2}, \cdots, W_{i,|V|}]$ represents the (nonnegative) weights on all edges initiating from the ith node. Here $W_{ij} = 0$ implies there is no edge from the ith node to the jth node; similarly, a nonzero W_{ij} indicates the existence of a directed edge from the ith node to the jth node, with weight W_{ij}. We refer to $w(i)$ as the *outgoing-vector* of the the ith node. We also define the *weight of a node* as $s_i = \sum_{j=1}^{|V|} W_{ij}$ for all i, which is the sum of the edge weights from the ith node.

Based on the preceding notation, the graph reduction problem can be stated as follows:
Given a directed graph $\mathcal{G}_x = (V_x, E_x, X)$ with $|V_x| = N$ nodes, whose weighting matrix is given by

$$X = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1N} \\ X_{21} & X_{22} & \cdots & X_{2N} \\ & \vdots & & \\ X_{N1} & X_{N2} & \cdots & X_{NN} \end{bmatrix}_{N\times N},$$

find a reduced graph $\mathcal{G}_y = (V_y, E_y, Y)$ with $|V_y| = M$ $(M < N)$ nodes and an $M \times M$ weighting matrix Y such that $\mathcal{G}_y$ is representative of the original graph $\mathcal{G}_x$. The coarseness of this reduced representation should be minimized (where the metric for coarseness is defined in the next section).

A pictorial illustration of this problem is given in Figure 2. The original graph $\mathcal{G}_x$ and the reduced graph $\mathcal{G}_y$ are shown in the lower and upper layers, respectively, which are similar to the two graphs in Figure 1. Since the reduced graph in the upper layer is unknown and determined by the lower layer, we will henceforth call it the *hidden graph*, and we will call the original graph the *observed graph*. Our goal is to measure and minimize the dissimilarity between the hidden and observed graphs. In fact, we search for a solution

$$\mathcal{G}_y^\star = \arg \min_{\mathcal{G}:|V|=M} \rho(\mathcal{G}_x, \mathcal{G}),$$

in which ρ is a metric for dissimilarity, representing the *distance* between two graphs.

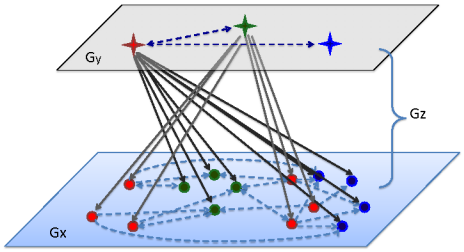

Fig. 2. Diagram demonstrating the relationship of the original graph, $\mathcal{G}_x = (V_x, E_x, X)$, to the reduced graph, $\mathcal{G}_y = (V_y, E_y, Y)$, and the intermediate graph, $\mathcal{G}_z = (V_z, E_z, Z)$. $\mathcal{G}_x$ is shown in the lower layer and $\mathcal{G}_y$ is shown in the upper layer; nodes and dashed-arrows in both layers illustrate the connections of the two graphs. $\mathcal{G}_z$ contains all nodes from both layers, with arrows that initiate only from hidden nodes, and terminate on observed nodes. Nodes in the hidden graph provide a partition of the nodes in the observed graph (shown with different colors), thus the arrows in the hidden graph illustrate the connections between partitions. Note that this functional partition does not necessarily coincide with geometric regions.

We define a dissimilarity metric $\rho(\mathcal{G}_x, \mathcal{G}_y)$ between two graphs in terms of an intermediate set of graphs $\mathbf{G}_{xy} = \{\mathcal{G}_z(V_z, E_z, Z)\}$, where each element $\mathcal{G}_z(V_z, E_z, Z)$ satisfies the following.

(i) The node set $V_z = V_y \cup V_x$ is the union of nodes in both graphs.

(ii) The edges $E_z \subset V_y \times V_x$ consist of arrows directed only from the hidden nodes to the observed nodes, that is, Z_{ij} can be non-zero only when i is a node in the hidden graph and j is a node in the observed graph. Therefore, although the graph $\mathcal{G}_z$ has $N+M$ nodes, we represent the weighting matrix by a $M \times N$ weighting matrix Z.

(iii) The nodes in the hidden graph give a partition of the nodes in the observed graph. That is, there exists a (surjective) partition function $\phi_z : I(x) \to I(y)$ (from the index of V_x to the index of V_y), such that for all $i \in V_x$, there exist a unique $k \in V_y$, such that $\phi_z(i) = k$ (each k corresponds to a color in Figure 2).

Note that it is enough to know $|V_x|$ and $|V_y|$ to generate $\mathbf{G}_{xy}$.

Since the outgoing-vectors of all nodes in the given graph $\mathcal{G}_x$ and all hidden nodes in the intermediate graph $\mathcal{G}_z \in \mathbf{G}_{xy}$ are of the same length, the distance $d(x(i), z(k))$ between the node i of $\mathcal{G}_x$ and node k of $\mathcal{G}_z$ can be computed by defining a distance function, such as the squared Euclidean norm $d(x(i), z(k)) = \|x(i) - z(k)\|^2$. We define a metric $\rho(\mathcal{G}_x, \mathcal{G}_z)$ between the observed graph and an intermediate graph by

$$\rho(\mathcal{G}_x, \mathcal{G}_z) \stackrel{\triangle}{=} \sum_{i=1}^{N} p_i \min_{1 \le j \le M} d\left(x(i), z(j)\right), \tag{1}$$

which is the weighted average distance between the nodes of the two graphs. Here the weights p_i denote the relative importance of the ith node in the observed graph, for example $p_i = 1/N$, if all nodes are equally important. Another reasonable choice for these weights is to let $p_i \propto s_i$, or $p_i \stackrel{\triangle}{=} s_i / (\sum_k s_k)$, which serves to define the importance of a node by its "activity level". The dissimilarity metric for two graphs $\mathcal{G}_x$ and $\mathcal{G}_y$ with *different* sizes can now be defined as

$$\rho(\mathcal{G}_x, \mathcal{G}_y) \stackrel{\triangle}{=} \min_{\mathcal{G}_z \in \mathbf{G}_{xy}} \rho(\mathcal{G}_x, \mathcal{G}_z).$$

Therefore, the graph reduction problem can be formulated as the following optimization problem:

$$\min_{\mathcal{G}_y : |V_y| = M} \rho(\mathcal{G}_x, \mathcal{G}_y). \tag{2}$$

III. Problem Solution: Model-Reduction of Large Graphs

To solve this optimization problem (2), we propose a two-step procedure:

(I) Determine $\mathcal{G}_{z^\star}$ and $\phi_z^\star$ for a fixed number of hidden nodes ($|V_y| = M$) that solves

$$\min_{\mathcal{G}_z \in \mathbf{G}_{xy}, |V_y| = M} \rho(\mathcal{G}_x, \mathcal{G}_z).$$

(II) Obtain E_y and the weighting matrix Y by aggregating the corresponding $Z^\star$, i.e.,

$$Y_{kl} = \sum_{\phi_z^\star(i) = l} Z_{ki}^\star. \tag{3}$$

The hidden graph $\mathcal{G}_y$ with M nodes is thus determined. The number of hidden nodes M representing the size of reduced model is also a decision variable, which can be chosen as the aimed size of reduced model.

We adopt a *soft partitioning* approach from the DA algorithm [7]–[9], [20] (in contrast to the *hard partitioning* approach, where each node uniquely and determinately belongs to one partition), which allows each node i to associate with more than one cell j through a (nonnegative) weighting parameter $p(z_j|x_i)$. More specifically, the dissimilarity function (1) is modified as

$$\rho(\mathcal{G}_x, \mathcal{G}_z) \quad = \quad \sum_{i=1}^{N} p_i \sum_{j=1}^{M} p(z_j|x_i) d(x(i), z(j)), \tag{4}$$

where $p(z_j|x_i)$ is also called the association weight of x_i to z_j. Note that, a choice of uniform distribution for these weights ($p(z_j|x_i) = 1/M, \forall i, j$) would lead to the "softest" partition while associating each x_i to the nearest z_j with probability one will yield a hard partition. The extent of *hardness* (or softness) of a partition for a choice of weight distribution $\{p(z_j|x_i)\}$ is given by its entropy

$$H(\mathcal{G}_z|\mathcal{G}_x) = -\sum_{i=1}^{N} p_i \sum_{j=1}^{M} p(z_j|x_i) \log p(z_j|x_i), \tag{5}$$

where the lower the entropy value, the harder the partition is. Since we need a soft partition to ensure non-local computations (to avoid local minima) and a hard partition for the minimum solution of (4), we successively solve the following optimization problem

$$\min_{\{p(z_j|x_i), z_j\}} F(\mathcal{G}_x, \mathcal{G}_z) \stackrel{\triangle}{=} \rho(\mathcal{G}_x, \mathcal{G}_z) - TH(\mathcal{G}_z|\mathcal{G}_x), \tag{6}$$

by decreasing T (also known as temperature), and thus trade maximizing H for minimizing ρ; that is, trade globality of computations for hardness of the partition. We refer to F as *free energy*, due to an analogue in thermal dynamics [21].

On setting $\nabla_{p(z_j|x_i)} F(\mathcal{G}_x, \mathcal{G}_z) = 0$ for a fixed T, we obtain the *Gibbs* distribution:

$$p(z_j|x_i) = \frac{\exp\left\{-\frac{1}{T} d(x(i), z(j))\right\}}{\sum_{k=1}^{M} \exp\left\{-\frac{1}{T} d(x(i), z(k))\right\}}; \tag{7}$$

which when substituted into (6), the free energy becomes

$$F = -T\sum_{i=1}^{N} p_i \log\left\{\sum_{k=1}^{M}\exp\left[-\frac{1}{T}d(x(i),z(k))\right]\right\}. \qquad (8)$$

On minimizing F with respect to Z by setting $\nabla_{z(j)}F(\mathscr{G}_x,\mathscr{G}_z)=0$, we get:

$$\begin{aligned}\nabla_{z(j)}F(\mathscr{G}_x,\mathscr{G}_z) &= \sum_{i=1}^{N} p_i p(z_j|x_i)\nabla_{z(j)}d(x(i),z(j)) \qquad (9)\\ \Rightarrow 0 &= \sum_{i=1}^{N} p[z(j)]p(x_i|z_j)\nabla_{z(j)}d(x(i),z(j)).\end{aligned}$$

The optimal $Z^\star$ is obtained from above equation when the distance function $d(\cdot,\cdot)$ is specified. After determining the intermediate weighting matrix $Z^\star$, we apply step (II) of our procedure and compute Y (and therefore achieve $\mathscr{G}_y$) by using following soft partition version of (3):

$$Y_{kl} = \sum_{j=1}^{N} p(z_l|x_j)Z_{kj}. \qquad (10)$$

In summary, at every fixed temperature T, we compute (7) - (10) and obtain a soft reduced graph model $\mathscr{G}_y^\star(T)$. Since the association weight (7) is uniform at high temperature, when the entropy maximization is the main goal, and becomes more differentiated as T decreases, when F gradually recovers to the distortion ρ, we gradually lower the temperature while track $\mathscr{G}_y^\star$. During this *annealing process*, the system undergoes *phase transitions*, where each hidden node recursively splits into multiple *distinct* nodes (see [20] for details) at critical values of T. In fact at $T=\infty$, there is only one *distinct* hidden node (supernode) solution ($z_j =$ constant for all j), and as annealing progresses (T decreases), this solution persists till a critical temperature T, when the number of *distinct* hidden nodes increases; and this number remains the same till the next critical temperature, and so on. This splitting can be interpreted as a hierarchical graph reduction, where successive splits identify clusters and consecutive sub-clusters of the nodes in the graph. This phase-transition property implies a multi-scaled graph reduction objective can be approached through the annealing process, which can be terminated once a target distortion is achieved. (See Figure 3 and Figure 4 in the simulation section for an illustration of this multi-scaled reduction.)

We demonstrate the solution procedure developed above by using squared-Euclidean distance $d(u,v)=\|u-v\|^2$, one of the most popular distance metrics in data-clustering and vector quantization literature [20], [22], to characterize distance between nodes. By substituting $d(\cdot,\cdot)$ into the optimality condition (9), we get

$$\begin{aligned}\nabla_{z(j)}F(Z) &= 2p(z_j)\sum_{i=1}^{N} p(x_i|z_j)\,[z(j)-x(i)] = 0,\\ \Rightarrow z^\star(j) &= \sum_{i=1}^{N} p(x_i|z_j^\star)x(i), \qquad (11)\end{aligned}$$

where $p(x_i|z_j)$ is the posterior probability of the association weights (7) and is computed using Bayes' Rule. Equation (11) indicates that the edge weights of $\mathscr{G}_{z^\star}$ are the weighted average of the edge weights in $\mathscr{G}_x$. Moreover, from (11) and (10), we obtain the weighting matrix Y given by

$$Y_{kl} = \sum_{j=1}^{N}\sum_{i=1}^{N} p(z_l^\star|x_j)p(x_i|z_k^\star)X_{il},$$

which is the cluster-wise weighted average of $Z^\star$.

IV. Model reduction for Markov chains

Markov chains can be viewed as directed graphs, where nodes represent the states and edges represent the corresponding transition probabilities, and therefore, in general form, the model reduction algorithm from Section III is directly applicable to Markov chains. Consider a discrete Markov chain $\mathscr{X}(t)=\{\mathscr{X}_1,\mathscr{X}_2,\cdots\}$ with a finite-state space $|\mathscr{X}|=N$ and one-step transition matrix $\mathbb{P}$. We construct the graph $\mathscr{G}_x(V_x,E_x,X)$, whose node set V_x represents the set of states, the set of directed edges represents one-step transitions, and with a weighting matrix X defined by the state-transition matrix of the Markov chain, that is, $X_{ij}\stackrel{\triangle}{=}\mathbb{P}_{ij}=\mathbb{P}(\mathscr{X}_j|\mathscr{X}_i), 1\le i,j\le N$. Therefore, the model reduction objective becomes: find a *similar* Markov model with fewer states. Without loss of generality, constructing a reduced Markov chain, $\mathscr{Y}(t)=\{\mathscr{Y}_1,\mathscr{Y}_2,\cdots\}$ with $|\mathscr{Y}|=M$ ($M<N$), can be viewed as the construction of the hidden graph $\mathscr{G}_y=(V_y,E_y,Y)$, where the goal is to maximize a similarity metric (or, minimize a dissimilarity metric).

We note the following properties of Markov chains and the implication of these properties on the corresponding graph-reduction problem in our framework:

(i) Since the weighting matrices are defined by the probability transition matrix, all node weights have to be one, that is both equations $X\cdot\mathbf{1}=\mathbf{1}$ and $Y\cdot\mathbf{1}=\mathbf{1}$ have to be satisfied.

(ii) Assume, in addition, that the Markov chain is irreducible and aperiodic, then there exists a stationary distribution $\pi=[\pi_1,\pi_2,\cdots,\pi_N]$, satisfying $\pi=\pi\mathbb{P}$ and $\sum_{i=1}^{N}\pi_i=1$. We also know that this is the limiting distribution under above assumptions, so π_i represents the long range proportion of time that the random process spends in the ith state. This interpretation provides a way to define the node importance by $p_i\stackrel{\triangle}{=}\pi_i$ for all i.

(iii) The weighting matrix Z of the intermediate graph $\mathscr{G}_z$ now represents the transition probabilities from *meta-states* [19] to observed states. Therefore, we apply node weight constraints on $\mathscr{G}_z$ to ensure each row of Z represents a distribution, that is,

$$\sum_{k=1}^{N} Z_{jk} = 1, \forall j. \qquad (12)$$

An appropriate metric to quantify the distance between two distributions $x(i)$ and $z(j)$ is the K-L divergence [23] given by

$$d(x(i),z(j))\stackrel{\triangle}{=}\sum_{k=1}^{N} X_{ik}\log\frac{X_{ik}}{Z_{jk}}, \qquad (13)$$

where we assume that the distributions in Z are absolutely continuous with respect to the distributions in X, that is $Z_{jk}=0\Rightarrow X_{ik}=0, \forall i,j,k$.

We employ the graph reduction procedure developed in Section III for the Markov-Chain reduction problem by substituting the K-L divergence (13) in (4) with $p_i = \pi_i$. On accounting for the constraints (12) in the solution procedure, (8) becomes

$$F = -T\sum_{i=1}^{N} p_i \log\left\{\sum_{k=1}^{M}\exp\left[-\frac{1}{T}d(x(i),z(k))\right]\right\} + \sum_{j=1}^{M}\sum_{k=1}^{N}\nu_j(Z_{jk}-1)$$

where the ν_j's are the Lagrange multipliers. To obtain the optimal weighting matrix $Z^\star$, we set $\frac{\partial F}{\partial Z_{lm}} = 0$ for each l and m, and get

$$\begin{aligned}
\nu_l Z^\star_{lm} &= \sum_i p_i p(z^\star_l|x_i) X_{im}, \\
\Rightarrow \sum_{m=1}^{N}\nu_l Z^\star_{lm} &= \sum_{m=1}^{N}\sum_{i}^{N} p_i p(z^\star_l|x_i) X_{im} = \nu_l. \\
\Rightarrow Z^\star_{lm} &= \frac{\sum_i p(x_i,z^\star_l)X_{im}}{\sum_m\sum_i p(x_i,z^\star_l)X_{im}} = \sum_i p(x_i|z^\star_l)X_{im}.
\end{aligned}$$

We obtain the reduced-order transition matrix Y from (10), where each row of Y sums up to one since

$$\sum_{l=1}^{M} Y_{kl} = \sum_{(l,j)=(1,1)}^{M,N} p(z^\star_l|x_j)Z^\star_{kj} = \sum_{j=1}^{N} Z^\star_{kj} = 1.$$

V. Simulation

We demonstrate our reduction algorithm for graphs on a test example comprised of 10 nodes with edges as shown in Figure 3(a). The weighting vector of each node is generated as follows: If node i and node j are connected, the connection weight X_{ij} is a realization of a Gaussian random variable with (prescribed, but random) significant mean and variance; otherwise X_{ij} is some Gaussian noise (with mean 0 and small variance). Figure 3 shows the multi-scaled reduction graphs on the application of our algorithm where the squared Euclidean distance function was used as the dissimilarity metric. Note that the number of *distinct* supernodes increases from 2 to 4 as the annealing parameter T is decreased, exhibiting the phase-transition property as well as the hierarchical reduction process of the graph. Figure 4 shows the simulation results for the Markov-Chain reduction problem using the same data set as above except that each row of the matrix X was scaled so that the elements summed to 1. Again the results demonstrate a useful reduction of the original Markov chain. These examples with 10 nodes were executed primarily to verify the algorithm, where the underlying data for the 10-node graph was generated based on a 4-supernode graph. These preliminary results demonstrate that the algorithm works well, and in fact, recovers the underlying supernode structure. Figures 5 and Figure 6 show the reduction results when we apply our algorithm to a graph with 40 nodes; this is of the equivalent size of many neural science examples (~ 20 to ~ 60 neurons). We generated the original graph and the location coordinates in the same way stated above and use the same group connections.

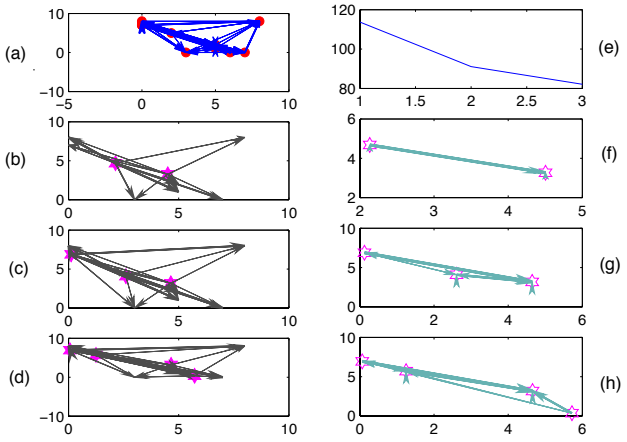

Fig. 3. The reduction results on a original graph with 10 nodes, when the Euclidean distance is adopted. (a) is the original directed graph, in which the red dots denote the observed nodes, and the blue arrows denote the connection weights. (b) depicts the values of free energy achieved by different orders of model reduction, in which there is a downward trend. The second to the fourth rows are reduced graphs with 2 to 4 hidden nodes, whose locations are denoted by stars (The locations are not important in determining graph structures, so we just take the weighted average locations of observed nodes). Plot (b), (c) and (c) show the connections from hidden nodes to observed nodes (i.e., $\mathcal{G}_z$ in our context), and Plot (f), (g) and (h) show the connections among hidden nodes (i.e. $\mathcal{G}_y$). In all figures, the width of an arrow is proportional to the weight of the edge.

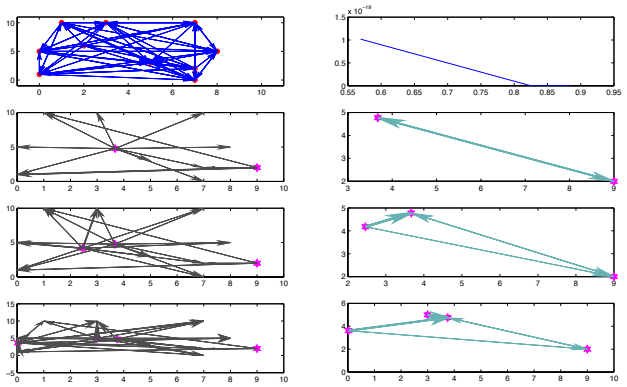

Fig. 4. The reduction results on a Markov model with 10 states, when the K-L divergence rate is adopted. Reduced graphs with 2, 3 and 4 meta-states are displayed.

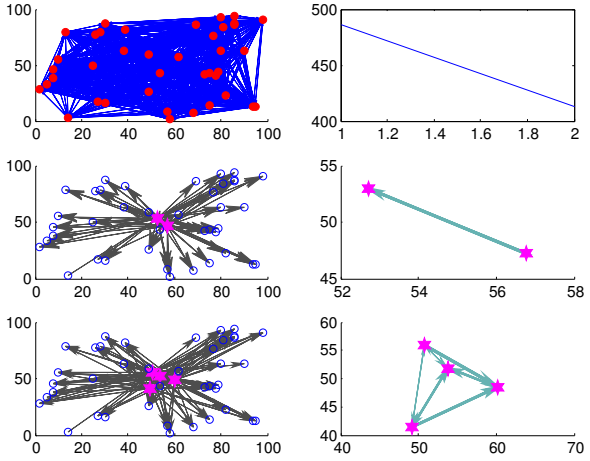

Fig. 5. Reduction results for a graph with 40 nodes when the squared Euclidean distance function is used.

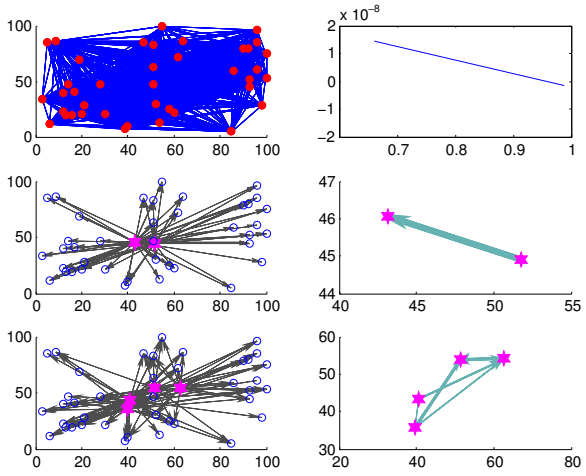

Fig. 6. Reduction results for a graph with 40 nodes when the K-L divergence is used.

VI. Analysis and Discussion

The maximum-entropy-principle based approach presented in this paper yields a tractable algorithm for (directed) graph reduction that identifies hierarchically the *natural* supernodes and that is designed to avoid local minima. This approach is flexible in terms of the distance functions that can be used in its formulation (e.g. squared Euclidean and K-L divergence used in Sections III and IV) and therefore accommodates many application areas (discussed in Section I). Also, the algorithm presented in this paper is for a simple graph-reduction problem; however the approach can easily incorporate dynamic, communication, and computational constraints by adapting the tools that we have developed for clustering/classification problems in our previous work [7]–[10]. For instance, we have applied this approach to simplification of influence diagrams obtained from neuroscientific community with good results, where we could account for the distance functions and constraints that are unique to that particular application area. Another advantage of this approach is that it is independent of the specific representation of the graph unlike many existing approaches whose solutions differ based on the permutations of the adjacency matrix associated with the graph.

At the outset, the algorithm presented in this paper can be thought to be computationally expensive, since it is not distributed - in fact, it uses computations that involve information from all the nodes to determine the reduced graph. However, the phase-transition property makes this algorithm progressively localized; which when exploited makes it computationally efficient [7], [9], [24]. Thus this algorithm makes use of the global information in its initial steps to avoid local minima while the localization of the latter iterations for reducing computational expense. In comparison to other annealing-based approaches (such as simulated annealing), the proposed approach is significantly faster (the annealing parameter is reduced geometrically as opposed to logarithmic rates for simulated annealing) [9].

References

[1] P. K. G. Thakurta, P. Sinha, N. Mallick, and S. Bandyopadhyay, "An approach towards reduction of routing paths for mobile networks," *AIP Conference Proceedings*, vol. 1298, no. 1, pp. 619–624, 2010.

[2] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle cooperative control," *Control Systems, IEEE*, vol. 27, no. 2, pp. 71 –82, april 2007.

[3] L. Mitiche, A. B. Adamou-Mitiche, and D. Berkani, "Low-order model for speech signals," *Signal Processing*, vol. 84, no. 10, pp. 1805 – 1811, 2004.

[4] D. Parikh and T. Chen, "Hierarchical semantics of objects (hsos)," *Computer Vision, IEEE International Conference on*, vol. 0, pp. 1–8, 2007.

[5] C. Quinn, T. Coleman, N. Kiyavash, and N. Hatsopoulos, "Estimating the directed information to infer causal relationships in ensemble neural spike train recordings," *Journal of Computational Neuroscience*, vol. 30, pp. 17–44, 2011.

[6] S. Samar and C. Beck, "Model reduction of heterogeneous distributed systems," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 5, dec. 2003, pp. 5271 – 5276 Vol.5.

[7] P. Sharma, S. Salapaka, and C. Beck, "A Scalable Approach to Combinatorial Library Design for Drug Discovery," *Journal of Chemical Information and Modeling*, vol. 48, no. 1, pp. 27–41, 2008.

[8] N. V. Kale and S. M. Salapaka, "Maximum entropy principle based algorithm for simultaneous resource location and multi-hop routing in multi-agent networks," *IEEE Transactions on Mobile Computing*, vol. 99, no. PrePrints, 2011.

[9] P. Sharma, S. Salapaka, and C. Beck, "Entropy-based framework for dynamic coverage and clustering problems," *Accepted to IEEE Transactions on Automatic Control*, 2011.

[10] Y. Xu, S. Salapaka, and C. L. Beck, "Dynamic maximum entropy algorithms for clustering and coverage control," in *Decision and Control, 2010. Proceedings of the 49th IEEE Conference on*, dec 2010, pp. 1836 –1841.

[11] P. Sharma, S. Salapaka, and C. Beck, "A Maximum Entropy Based Scalable Algorithm for Resource Allocation Problems," *Proceedings of American Control Conference*, pp. 516–521, 2007.

[12] K. Deng, P. Mehta, and S. Meyn, "A simulation-based method for aggregating markov chains," in *Decision and Control, 2009. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, dec 2009, pp. 4710 –4716.

[13] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine Learning*, vol. 56, pp. 9–33, 2004.

[14] K. M. Hall, "An r-dimensional quadratic placement algorithm," *Management Science*, vol. 17, pp. 219–229, nov 1970.

[15] W. E. Donath and A. J. Hoffman, "Lower bounds for the partitioning of graphs," *IBM J. Res. Dev.*, vol. 17, pp. 420–425, sep 1973.

[16] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, pp. 1299–1319, jul 1998.

[17] R. Phillips and P. Kokotovic, "A singular perturbation approach to modeling and control of markov chains," *Automatic Control, IEEE Transactions on*, vol. 26, no. 5, pp. 1087 – 1094, Oct. 1981.

[18] H. A. Simon and A. Ando, "Aggregation of Variables in Dynamic Systems," *Econometrica*, vol. 29, no. 2, pp. 111–138, 1961.

[19] C. Beck, S. Lall, T. Liang, and M. West, "Model reduction, optimal prediction, and the mori-zwanzig representation of markov chains," in *Decision and Control, 2009. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, 2009, pp. 3282 –3287.

[20] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–39, nov 1998.

[21] L. D. Landau and E. M. Lifshitz, *Statistical Physics, Part 1*, 3rd ed. Oxford, 1980, vol. 3.

[22] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, 1st ed. Kluwer, Boston, Massachusetts, 1991.

[23] Z. Rached, F. Alajaji, and L. Campbell, "The kullback-leibler divergence rate between markov sources," *Information Theory, IEEE Transactions on*, vol. 50, no. 5, pp. 917 – 921, May 2004.

[24] A. Kwok and S. Martinez, "A distributed deterministic annealing algorithm for limited-range sensor coverage," *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 4, pp. 792 –804, july 2011.