Multiway k-Cut in Static and Dynamic Graphs: A Maximum Entropy Principle Approach

Mayank Baranwal[1], Amber Srivastava[2a], and Srinivasa M. Salapaka[2b]

Abstract— This work presents a maximum entropy principle based algorithm for solving minimum multiway k-cut problem defined over static and dynamic *digraphs*. A multiway k-cut problem requires partitioning the set of nodes in a graph into k subsets, such that each subset contains one prespecified node, and the corresponding total cut weight is minimized. These problems arise in many applications and are computationally complex (NP-hard). In the static setting this article presents an approach that uses a relaxed multiway k-cut cost function; we show that the resulting algorithm converges to a local minimum. This iterative algorithm is designed to avoid poor local minima with its run-time complexity as $\sim O(kIN^3)$, where N is the number of vertices and I is the number of iterations. In the dynamic setting, the edge-weight matrix has an associated dynamics with some of the edges in the graph capable of being influenced by an external input. The objective is to design the dynamics of the controllable edges so that multiway k-cut value remains small (or decreases) as the graph evolves under the dynamics. Also it is required to determine the time-varying partition that defines the minimum multiway k-cut value. Our approach is to choose a relaxation of multiway k-cut value, derived using maximum entropy principle, and treat it as a control Lyapunov function to design control laws that affect the weight dynamics. Simulations on practical examples of interactive foreground-background segmentation, minimum multiway k-cut optimization for non-planar graphs and dynamically evolving graphs that demonstrate the efficacy of the algorithm, are presented.

I. INTRODUCTION

The *Multiway k-Cut* Problem [1] is a generalization of minimum *s-t* cut problem and has applications in parallel and distributed computing [2], as well as in chip design. Multiway cut also finds applications in several other problems of related interest such as extending a partial k-coloring of a graph [3]. Given a graph $G = (V, E)$ with vertex set $V, |V| = N \in \mathbb{N}$, set E of edges, edge weights $w : E \to \mathbb{R}^+$, and a set of terminals $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$, a *multiway k-cut* is a set of edges whose removal disconnects the terminals from each other. The goal of the minimum multiway k-cut problem is to find a minimum weight set $E' \subseteq E$ of edges such that removing E' from G separates all terminals. Fig. 1a shows a schematic of a minimum multiway 3-cut problem with set of terminals denoted by $S = \{s_1, s_2, s_3\}$. The objective is to obtain a partition of the vertex set V into disconnected
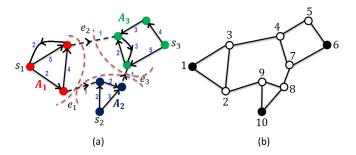
Fig. 1: (a) Schematic of a minimum multiway k-cut problem with $k = 3$ and $S = \{s_1, s_2, s_3\}$. Here A_1, A_2 and A_3 are the disconnected components. (b) Illustrative example of a multiway 3-cut problem with $S = \{1, 6, 10\}$ and unit edge-weights described in Sec. VI.

components $\{A_j : A_j \subset V, 1 \leq j \leq 3\}$, such that $s_j \in A_j$ for all j, and the total cut size, $\frac{1}{2} \sum_{j=1}^{3} w(A_j, \bar{A}_j)$ is minimized.

While the problem of computing a minimum *s-t* cut (i.e., $k = 2$) in static graphs is solvable in polynomial time, it is shown in [4] that the minimum multiway k-cut problem is not just **NP**-hard, but also **APX**-hard, i.e., there is a constant $\delta > 1$ such that it is **NP**-hard to even approximate the solution to within a ratio of less than δ to the optimal cost even when restricted to instances with three terminals ($k = 3$) and unit edge costs. The special case of the problem on planar graphs is also **NP**-hard if k is arbitrarily large, but can be solved in polynomial time for every fixed k [5]. The complexity of multiway k-cut problem arises from the *combinatorial* number of ways in which the vertex set V can be partitioned into k feasible sets.

In many application areas such as social networks, communication networks and epidemic spread networks, the corresponding graph models are time-varying in nature [6], [7]. For instance, a company wants to maintain its prominence over a certain user base represented by a graph $G(V, E)$. The set of vertices V includes all the existing and potential costumers and the set E of edges denotes the influence of one customer over the other. Such a graph is temporal in nature since the influence of two nodes over one another changes with time depending on several factors such as the number of text messages exchanged in a day. The company identifies k agents $\{s_1, \ldots, s_k\}$ in the set V of vertices and partitions the graph into k subsets such that there is exactly one agent in each subset and the total cut value is minimized. The agents help the company provide services to the customers, as well as, influence them in company's favor by providing customized schemes for each subset. Owing to the dynamic

nature of this setting it becomes imperative to find the time-varying minimum multiway k-cuts and also, if possible, influence consumer interactions such that cut value remains small (or decreases) with time. In this context the objective is to design control input that externally influence the edge-weight dynamics and determine the minimum multiway k-cut at each time instant.

In this paper, we first present an innovative relaxation heuristic for the minimum multiway k-cut problem on static digraphs (weighted and directed graphs). The heuristic is based on maximum-entropy-principle (MEP), which in turn, has close analogy to minimum free-energy principle in statistical physics [8]. Using this heuristic we introduce probability distributions on the space of associations between vertices and the terminal points; this constitutes the *soft-partitioning* of the graph. We then seek the fairest distribution (that is with maximum entropy) that guarantees the expected cut values to lie below a prescribed upper bound. A sequence of such problems are solved as the values of the upper bound are successively decreased. At the end of these iterations, the resulting distributions are concentrated about the mean values, and these mean values approximate the solutions to the original unrelaxed problem. In the context of data clustering this MEP based heuristic has resulted in the deterministic annealing (DA) algorithm [9]. DA has been successfully applied to a large class of optimization problems such as, pattern classification [10], image segmentation [11], graph aggregation [12], robust speech recognition [13], multiple travelling salesman [14] and coverage control [15]. However, DA is primarily a resource allocation algorithm and requires the precise knowledge of coordinates of data points that need to be clustered. An advantage of our proposed methodology is that it does not require the knowledge of coordinates of data points provided that a *similarity measure* (such as edge-weight matrix in the context of a graph) between the data points is known.

In this work, we propose an MEP based algorithm that does away with this limitation of the DA algorithm and is applicable to optimization problems on graphs. In particular, we develop an iterative algorithm for the minimum multiway k-cut problem on *weighted* and *directed* graphs and present its convergence analysis by exploiting the algebraic structure of the relaxed cost function and non-negativity of the Kullback-Leibler divergence. An important feature of our algorithm is that it is independent of the initialization and is designed to avoid poor local minima. In fact, the algorithm always starts with uniform distribution for the target probability distributions which evolve into appropriate concentrated distributions. For a graph with N vertices, the run-time complexity of the proposed algorithm is $\sim O(kIN^3)$, where N and I denote the number of vertices and iterations respectively. Empirical evaluations on planar and non-planar graphs suggest that our algorithm outperforms the approximation algorithms described in [4], [16], [17]. In fact, these approximation algorithms are known to result in highly suboptimal solutions on specific graph instances. We show that our approach returns optimal solutions on such otherwise challenging instances. We further demonstrate that our algorithm is also capable of handling very large graph instances (comprising of image pixels as nodes) for interactive foreground-background segmentation.

In the context of determining the minimum multiway k-cut on a dynamic graph there could be several simple approaches; one of the straightforward method is the *frame-by-frame* approach where we solve for the minimum cut at each time instant. However such a methodology is computationally expensive and non-viable for very large graphs. In our proposed methodology we first choose an energy function that captures the multiway k-cut value as well the external control effort. We then treat this function as a control Lyapunov function to design the control input, which determines the manipulable edge-weight dynamics and the time-varying minimum multiway k-cuts. We show that this approach is not conservative, that is, under the assumption of feasibility, it always results in a control law that ensures the non-positiveness of the time derivative of the energy function. Our simulations demonstrate reduction in computational times by over 25 times in comparison to the frame-by-frame approach.

II. PROBLEM FORMULATION IN STATIC GRAPHS

For a given weighted directed graph $G = (V, E, W)$ with vertex set $V, |V| = N \in \mathbb{N}$, set $E \subseteq V \times V$ of edges, edge weight matrix $W = [w_{lm}] \in \mathbb{R}_{\geq 0}^{N \times N}$ with w_{lm} as the weight of the edge from vertex l to m and $w_{lm} = 0 \ \forall \ (l, m) \notin E$, and a set of terminals $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$, the minimum multiway k-cut problem is defined as

$$\min_{\substack{\{A_1, A_2, \ldots, A_k\}, \\ A_j \subset V}} \frac{1}{2} \sum_{j_1=1}^{k} \sum_{\substack{j_2=1 \\ j_2 \neq j_1}}^{k} \sum_{\substack{l \in A_{j_1} \\ m \in A_{j_2}}} w_{lm}$$

$$\text{s.t.} \quad s_1 \in A_1, \ldots, s_k \in A_k, \quad \cup_{i=1}^{k} A_k = V,$$

$$\text{and} \quad A_j \cap A_l = \phi, \text{ for all } j \neq l. \tag{1}$$

Note that, a partition $\{A_1, \ldots, A_k\}$ of V results in k disconnected components (subgraphs) of graph G (see Fig. 1a). A component A_j contains all the vertices i in V that are in the same subgraph as the terminal s_j. Here a *multiway k cut* represents the set of edges whose vertices belong to distinct components (i.e. distinct A_j's). For instance the set $\{e_1, e_2, e_3\}$ is a multiway k cut for the graph in Fig. 1(a) where set of terminals is $\{s_1, s_2, s_3\}$. The above optimization problem seeks a partition that minimizes the total *cut weight*, i.e. the cumulative weight of all the edges in the multiway k cut.

We reformulate the optimization problem in (1) by introducing $N \times k$ *soft* decision variables $\{p(j|i)\}$. These variables describe soft partitions of the graph. Here $p(j|i) \in [0, 1]$ denotes *a probabilistic* association of the vertex i with terminal s_j. We also require that for each vertex i, $\sum_j p(j|i) = 1$, i.e., $\{p(j|i)\}$ ascribes a probability distribution over all feasible associations over vertex set V. In the case of hard partitions in (1) all the decision variables $\{p(j|i)\}$ will be taking values either 0 or 1. In this case a component A_j corresponds to a set

$\{i \in V : p(j|i) = 1\}$. A relaxation of the optimization problem (1) is given by

$$\min_{\substack{\{p(j|i)\} \\ 1 \le i \le N \\ 1 \le j \le k}} D \triangleq \frac{1}{2} \sum_{\substack{j_1, j_2 = 1 \\ j_2 \ne j_1}}^{k} \sum_{l,m=1}^{N} p(j_1|l) p(j_2|m) w_{lm}$$

$$\text{s.t.} \quad p(j_1|s_1) = \cdots = p(j_k|s_k) = 1, \qquad (2)$$

where we set $W \leftarrow W + \lambda \mathbb{I}_N$. The inclusion of a constant parameter λ or equivalently the *regularizer* term in D is explained as follows. Note that in formulation (1), self-loop edges with node weights $w(l,l)$ can not be in a multiway k-cut, and therefore the solution to this problem is *independent* of these weights, that is, independent of the diagonal entries of the matrix W. However in the proposed relaxation, since each vertex has partial membership with respect to different terminals s_j, the self-loop edges can become a part of the multiway k-cut; so we replace the edge-weight matrix W by $W + \lambda I_N$, i.e.

$$W \longleftarrow W + \lambda I_N, \qquad (3)$$

where I_N is a $N \times N$ identity matrix and $\lambda > 0$ is large enough to make sure that self-loop edges are not included in the multiway k-cut; the choice of λ is discussed in Section IV.

Remark 1: Note that without loss of generality we can assume the edge-weight matrix W to be symmetric. In fact, solution to the optimization problem in (2) for a given edge-weight matrix W is identical to solution of a similar optimization problem with edge-weight matrix $\tilde{W} := 0.5(W + W^T)$ which can be verified by substituting $\sum_{j_2 \ne j_1} p(j_2|m) = 1 - p(j_1|m)$ in expression of D in (2).

III. SOLUTION TO MULTIWAY k-CUT IN STATIC GRAPHS

In the proposed approach instead of directly solving (2) we use the Maximum Entropy Principle (MEP) [8] to determine the distributions $\{p(j|i)\}$ that ensure the relaxed cost function D in (2) is less than or equal to a constant $d_0 > 0$. More specifically, this principle states that of all the probability distributions that satisfy a given set of constraints on expected values of functions of a random variable, choose the one that maximizes the Shannon entropy $H(\{p(j|i)\})$. Accordingly in our case, the MEP would solve $\max H(\{p(j|i)\})$ under the constraint that $D \le d_0$, where D is given in (2) and the Shannon entropy term is given by

$$H \triangleq -\sum_{i=1}^{N} \sum_{j=1}^{k} p(j|i) \log p(j|i). \qquad (4)$$

The equivalent Lagrangian is thus defined as

$$L \triangleq D - d_0 - \frac{1}{\beta} H, \qquad (5)$$

where Lagrange multiplier β controls the trade-off between minimizing cost function D and maximizing entropy H.

In the expression of the Lagrangian (5) we refer to the Lagrange multiplier $1/\beta$ as temperature and L as *free energy* because of their close analogies to statistical physics (where free energy is enthalpy (D) minus the temperature times

entropy (TH)). Note that the free energy L can also be viewed as a relaxation of the cost function (2). In fact, as $\beta \to \infty$, we note that $L \to D$. Using the fact that $\sum_{j_2 \ne j_1}^{k} p(j_2|i) = 1 - p(j_1|i)$ and d_0 is a constant, the effective Lagrangian in (5) is given by

$$L = \frac{1}{2} \sum_{l,m=1}^{N} w_{lm} \left(1 - \sum_{j=1}^{k} p(j|l) p(j|m)\right)$$
$$+ \frac{1}{\beta} \sum_{i=1}^{N} \sum_{j=1}^{k} p(j|i)) \log p(j|i). \qquad (6)$$

The above reformulation is critical to the convergence analysis described in Section IV. By setting $\frac{\partial L}{\partial p(j|i)} = 0$ toward minimizing (local) L with respect to $p(j|i)$ yields

$$p(j|i) = \frac{\exp\left\{\beta \left(\sum_{m=1}^{N} p(j|m) w_{im}\right)\right\}}{Z_i}, \qquad (7)$$

where the normalizing constant Z_i is given as

$$Z_i = \sum_{j} \exp\left\{\beta \left(\sum_{m=1}^{N} p(j|m) w_{im}\right)\right\}. \qquad (8)$$

We then substitute the expression (7) of $p(j|i)$ in the free-energy L (6) to obtain

$$L \triangleq \frac{1}{2} \sum_{l,m=1}^{N} w_{lm} \left(1 - \sum_{j=1}^{k} p(j|l) p(j|m)\right)$$
$$- \frac{1}{\beta} \sum_{i=1}^{N} \log \left(\sum_{j=1}^{k} \exp \left(\sum_{m=1}^{N} p(j|m) w_{im}\right)\right). \qquad (9)$$

The essence of the MEP-based approach lies in successive evaluations of Gibbs distribution in (7). Note that from (9) minimizing L at small values of β is equivalent to maximizing entropy H, which in turn corresponds to uniform distribution. As β is gradually increased, minimization of Lagrangian in (5) puts more weight on minimization of the cost function (2) and as evident from (7) results in *hard* (0-1) associations $\{p(j|i)\}$ as $\beta \to \infty$. This process of gradual *cooling* is referred to as *annealing* in the statistical physics literature. Also observe that increasing the Lagrange parameter β is equivalent to solving the same MEP problem with a decreased value of d_0 [8]. Thus as $\beta \to \infty$, the algorithm seeks the minimum value for the cost D in (2). In the proposed algorithm (see Algorithm 1) we minimize the free-energy L through fixed-point iterations in (7) at successively increasing values of the annealing parameter β. Note that at each value of the annealing parameter β, the algorithm executes the following two steps to solve the equation (7)

Step 1: $\quad \sigma_{lj}^+ \leftarrow p(j|l),$

Step 2: $\quad p^+(j|l) \leftarrow \dfrac{\exp\left\{\beta \left(\sum\limits_{m=1}^{N} \sigma_{mj}^+ w_{lm}\right)\right\}}{\sum_j \exp\left\{\beta \left(\sum\limits_{m=1}^{N} \sigma_{mj}^+ w_{lm}\right)\right\}},$

$$\forall l \in \{1, \ldots, N\} \setminus S, j \in \{1, \ldots, k\} \qquad (10)$$

Fig. 2: Phase Transition in Algorithm 1 applied to problem stated in Fig. 1(b) with unit edge-weights.

Algorithm 1 Algorithm for minimum MultiwayCut

Input: $G = (V,E)$, $w : E \to \mathbb{R}^+$, $S = \{s_1, \dots, s_k\}$
Output: $\{p(j|i)\}$
   Initialization:
   $p(j|j) \leftarrow 1 \quad \forall s_j \in S, \quad p(j|i) \leftarrow \frac{1}{k} \quad \forall i \notin S, \forall j$
   $\beta \leftarrow \beta_{\min}$
   Annealing Process
   while $\beta < \beta_{\max}$ **do**
      Fixed-Point Iterations
      while until convergence **do**
         Update $\{p(j|i)\}$ as in (7) $\forall i \notin S$
      end while
      Increment β
   end while
   return $\{p(j|i)\}$

Time-complexity of the proposed algorithm: The main complexity of this algorithm stems from the matrix multiplication in the fixed point iteration scheme. For a graph $G = (V,E)$ with $|V| = N$, there is a total Nk association probability parameters $\{p(j|i)\}$, that need to be estimated at each β iteration. Note that the *batch* update equation in (7) requires multiplying the two matrices W and $[p(j|i)]_{N \times k}$. This multiplication operation runs in $O(N|E|k)$ time (total of $N|E|$ operations for each partition associated with multiplying non-zero elements of edge-weight matrix). Thus the run-time complexity of the proposed algorithm is $\approx O(kIN^3)$ where I accounts for the number of β iterations and the fixed-point iterations.

Remark : Usually the MEP-based heuristics developed for solving the combinatorial optimization problems undergo several *phase transitions* [9] as the parameter β increases from zero to a large number. In the case of the Algorithm 1, these phase transitions that occur at certain critical β_{cr}'s, correspond to an abrupt change in the weight of the partitions A_j defined as $p(A_j) := \sum_i p(j|i)/N$. Empirical evaluations suggest that $p(A_j)$ does not change much between two consecutive phase transitions. Hence in our simulations we anneal the parameter β geometrically, i.e. $\beta_t = \gamma^t \beta_0$ where $\gamma > 1$ and the effective number of β iterations is $\log \frac{\beta_{\max}}{\beta_{\min}}$ which is small. In fact in all our simulations $\log \frac{\beta_{\max}}{\beta_{\min}} \lessapprox 10$. Fig. (2) demonstrates the phase transition phenomenon observed in our simulation when Algorithm 1 is applied to the problem stated in Fig. 1(b) with unit edge-weights. In our ongoing work we are solving for analytical conditions that determine the critical β_{cr} at which the phase transition phenomenon occurs.

IV. CONVERGENCE ANALYSIS

In this section we provide a proof that the Lagrangian L in (6) converges to a local minimum under the two-step iterations specified by (10). More specifically we show that for a fixed value of the Lagrange parameter β, every successive iteration of (10) decreases the effective Lagrangian L and
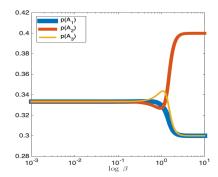
since L is lower bounded it converges to a local minimum. We use $P \in \mathbb{R}^{N \times k}$ to denote the matrix of associations $[p(j|i)]$.

Claim : The Lagrangian $L(P)$ in (6) converges to a local minimum under the fixed point iterations (7) (equivalently the two step iterations in 10) in Algorithm 1.

Proof. We first show that $L(P)$ is decreasing under the two-step iterations (10); that is we show that $\Delta(P, P^+) := L(P) - L(P^+) > 0$ where P^+ is obtained after successive executions of steps 1 and 2 in (10). Towards this end, we first construct a function $\Gamma(\zeta, \eta) : \mathbb{R}^{N \times k} \times \mathbb{R}^{N \times k} \to \mathbb{R}$ as

$$\Gamma(\zeta, \eta) \triangleq \frac{1}{2} \sum_{l,m,j=1}^{N,N,k} \left(\eta_{lj} - 2\zeta_{lj} \right) \eta_{mj} w_{lm} + \frac{1}{\beta} \sum_{i,j=1}^{N,k} \zeta_{ij} \log \zeta_{ij}; \quad (11)$$

here observe that $\Gamma(P,P) = L(P)$. Therefore

$$\Delta(P, P^+) = \Gamma(P,P) - \Gamma(P^+, P^+), \quad (12)$$

using the fact that $\Gamma(P,P) = L(P)$. To show that $\Delta(P, P^+) \geq 0$ we re-write it as

$$\Delta(P, P^+) = \underbrace{\Gamma(P^+, P) - \Gamma(P^+, P^+)}_{\Delta_1} + \underbrace{\Gamma(P,P) - \Gamma(P^+, P)}_{\Delta_2}, \quad (13)$$

and then show that both Δ_1 and Δ_2 are non-negative, i.e. $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$. We show that $\Delta_1 \geq 0$ by showing that $\Gamma(P^+, \sigma)$ (a quadratic function of σ with fixed P^+) achieves its minimum when $\sigma = P^+$. The stationary point is obtained by setting $\frac{\partial \Gamma}{\partial \sigma_{ij}}\big|_{(P^+, \sigma)} = 0$ in (11), which yields

$$\sigma_{ij} = p^+(j|i). \quad (14)$$

This solution $\sigma = P^+$ is a minimum if the Hessian $\frac{\partial^2 \Gamma(P^+, \sigma)}{\partial \sigma^2}\big|_{\sigma = P^+} = W$ is positive-definite. Choosing λ in (3) such that $\lambda = \sum_i \sum_{j \neq i} w_{ij}$, ensures positive-definiteness of the W matrix by the Gershgorin circle theorem [18]. Since P^+ is a minimizer, setting $\sigma = P \neq P^+$ results in $\Delta_1 \geq 0$.

To show that $\Delta_2 \geq 0$, note that from (11) we have

$$\Delta_2 = -\frac{1}{2} \sum_{lmj} \left(p(j|l) - p^+(j|l) \right) p(j|m) w_{lm}$$

$$+ \frac{1}{\beta} \sum_{lj} \left(p(j|l) \log p(j|l) - p^+(j|l) \log p^+(j|l) \right). \quad (15)$$

Also from taking logarithm on both sides of (7), we have

$$\sum_m p(j|m) w_{lm} = \frac{1}{\beta} \log Z_l + \frac{1}{\beta} \log p(j|l), \quad (16)$$

where Z_l is given by (8). Using (16) and the fact that $\sum_j p(j|l) = 1 \; \forall \; l \in V$ we obtain

$$\sum_{lj} \left(p(j|l) - p^+(j|l) \right) \sum_m p(j|m) w_{lm}$$

$$= \frac{1}{\beta} \sum_{lj} \left(p(j|l) - p^+(j|l) \right) \log p^+(j|l). \quad (17)$$

On substituting (17) in (15), we obtain

$$\Delta_2 = \frac{1}{\beta} \sum_{l,j=1}^{N,k} p(j|l) \log \frac{p(j|l)}{p^+(j|l)} = \frac{1}{\beta} \sum_{l=1}^{N} D_{KL}(P_l || P_l^+) \geq 0, \quad (18)$$

where P_l and P_l^+ are the lth rows of the matrices P and P^+ respectively, and $D_{KL}(P_l||P_l^+)$ represents the Kullback-Leibler measure. Consequently, $\Delta(P, P^+) \geq 0$. Thus we have shown that $L(P)$ decreases as a result of the two-step iteration in (10), and since $L(P)$ is bounded from below, the fixed point iterations in (7) converge to a local minimum of $L(P)$.

V. EXTENSION TO DYNAMIC GRAPHS

Here we consider a dynamic digraph $G(t) = (V, E, W_0, \Sigma)$ where V, $|V| = N \in \mathbb{N}$ denotes the set of vertices, $E \subseteq V \times V$ denotes the set of edges, $W_0 = [w_{0_{lm}}] \in \mathbb{R}^{N \times N}$ denotes the edge-weight matrix with $w_{0_{lm}} = 0 \; \forall \; (l,m) \notin E$, and Σ denotes the dynamics of the time-varying edge-weight matrix $W(t) = [w_{lm}(t)]$ given by $\dot{W} = \mathscr{F}(W,U)$ where $W(t_0) = W_0$ and $\dot{w}_{lm}(t) = 0 \; \forall \; (l,m) \notin E$. We assume that the matrix function $\mathscr{F} := [f_{ij}]$ is known a priori and defined as $f_{ij} : \mathbb{R}^{N \times N} \times \mathbb{R}^{N \times N} \rightarrow \mathbb{R}$ where f_{ij} belong to the class of continuously differentiable $C^1(\mathbb{R}^{N \times N})$ functions. Here $U(t) = [u_{ij}(t)] \in \mathbb{R}^{N \times N}$ denotes the external control input and is defined as $u_{ij} : \mathbb{R} \rightarrow \mathbb{R}$. The function u_{ij} belongs to the class of continuously differentiable $C^1(\mathbb{R})$ functions.

Remark 2: The matrix function $\mathscr{F}(W,U)$ encodes the information whether a particular edge $(l,m) \in E$ in the graph is manipulable or not.

The objective here is two-fold (a) design a control $U(t)$ such that a modified multiway k-cut value in (19) is minimized at every time instant t, and (b) determine the time-varying multiway k-cut for the dynamically evolving graph $G(t)$ given the set of terminal $S = \{s_1, \ldots, s_k\} \subseteq V$. The corresponding dynamic optimization problem is given by

$$\min_{\substack{\{A_j\}, A_j \subset V \\ \{u_{ij}(t) \in C^1\}}} \; \frac{1}{2} \sum_{j_1=1}^{k} \sum_{\substack{j_2=1 \\ j_2 \neq j_1}}^{k} \sum_{\substack{l \in A_{j_1} \\ m \in A_{j_2}}} \left(w_{lm}(t) \right)^2 + \mu \|U(t)\|_F^2$$

$$\text{s.t.} \quad s_j \in A_j \forall \; j \in \{1, \ldots, k\}, \cup_{j=1}^{k} A_j = V,$$

$$A_j \cap A_l = \phi \forall \; j \neq l,$$

$$\dot{W}(t) = \mathscr{F}(W,U), \quad W(t_0) = W_0, \quad (19)$$

where $\mu \in \mathbb{R}$ is a user-defined parameter, and $\| \cdot \|_F$ is the Frobenius norm. The first part of this objective function corresponds to the multiway k-cut cost, where each edge-weight

is replaced by its square, and the second part represents a penalty on the control effort. The user-defined parameter μ regulates the relative weight given to the control effort $\mu \|U(t)\|_F^2$. Observe that the edge-weight dynamics described by $\dot{W}(t) = \mathscr{F}(W,U)$ also allows for edge-weights to possibly become negative. This issue is addressed by modifying the cost function to have squared weight terms $\{w_{lm}^2\}$ instead of $\{w_{lm}\}$. In this way, the cost function penalizes only the magnitude (square) of the edge weights regardless of their signs and the resulting cuts include only those edges whose weights have small magnitude; a feature required in most applications. We discuss the case with the original (not modified) cost function in remark 4. The dynamic optimization problem in (19) inherits the computational complexity of the static-problem; which is further worsened by the dynamical aspect of the problem. Solving for $U(t)$ at each time-instant while ensuring that the function $U(t)$ is smooth is one of the main contributors to the additional complexity.

One straightforward method is the *frame-by-frame* approach which disregards this constraint. Here we set $U(t) = 0$ for all t and solve the minimum multiway k-cut problem using the Algorithm 1 (for the cost function in (19)) at every time instant t. A disadvantage of this approach is that, if the time interval Δt_p between two successive runs of the Algorithm 1 is short, then the overall approach is computationally expensive. Also the frame-by-frame analysis does not exploit the information available from the previous time instances to determine the minimum multiway k-cut at the current time instant. On the other hand, if the time interval Δt_p between two successive instances is large, then the algorithm cannot account for dynamics in this time interval and the resulting cut may be correspondingly large.

We propose an alternative method, where instead of solving directly the dynamic optimization problem in (19), we address its objectives of minimizing the multiway k-cut and the control effort. Here we consider an energy-like function $F(W(t), U(t))$ and design a dynamic control law $\dot{U}(t) = \mathscr{V}(W,U)$ such that $\dot{F}(W,U) \leq 0$. The function is given by

$$F(W(t), U(t)) := L_1(W(t)) + \mu \|U(t)\|_F^2, \quad (20)$$

where $L_1(W(t)) =$

$$\frac{1}{2} \sum_{l,m=1}^{N} w_{lm}^2 \left(1 - \sum_{j=1}^{k} p(j|l) p(j|m) \right) + \frac{1}{\beta} \sum_{i,j=1}^{N,k} p(j|i) \log p(j|i) \quad (21)$$

is the effective Lagrangian (6) modified with the squared edge-weights. Since this Lagrangian is a close approximation of the multiway k-cut value described in (19) (especially for high values of β as seen in Section III); its inclusion in $F(W,U)$ addresses the objective of having *small* multiway k-cut values.

The time-derivative $\dot{F}(W,U)$ is given by

$$\dot{F} = e_N^T \left(\Phi \circ W(t) \circ \mathscr{F}(W,U)(t) + 2\mu U(t) \circ \mathscr{V}(t) \right) e_N, \quad (22)$$

where $\dot{U}(t) = \mathscr{V}(W,U)$ represents a dynamic control law, $e_N \in \mathbb{R}^N$ is a column of all 1's, $\Phi = (\phi_{lm}) \in \mathbb{R}^{N \times N}$ such that $\phi_{lm} = 1 - \sum_{j=1}^{k} p(j|l) p(j|m)$ and $\circ$ denotes the Hadamard

product. We exploit the affine dependence of $\dot{F}$ on $\mathcal{V}$ in (22) to make $\dot{F}$ non-positive analogous to control based on control Lyapunov functions [19], [20], [21]. Specifically we choose

$$\mathcal{V} = -\left[C_0 + \frac{\alpha + \sqrt{\alpha^2 + (4\mu\|U(t)\|_F^2)^2}}{4\mu\|U(t)\|_F^2}\right] U(t), \qquad (23)$$

where $\alpha = 2e_N^T \Phi \circ W \circ \mathcal{F}(W,U) e_N$ and $C_0 > 0$. The following theorem summarizes the consequences of this design:

Theorem 1:

1) $F(W,U)$ is lower bounded; more specifically $F(W,U) + \frac{1}{\beta} N \log k \geq 0$.

2) If there exists a dynamic controller $\dot{U} = \bar{\mathcal{V}}(W,U)$ such that

   a) $\mathcal{V}(W,U)$ is locally Lipschitz,
   b) $\dot{F} = \frac{1}{2}\alpha + 2\mu U(t) \circ \bar{\mathcal{V}} \leq 0$, where $\alpha = 2e_N^T \Phi \circ W \circ \mathcal{F}(W,U)e_N$,

   then the control design $\dot{U} = \mathcal{V}(W,U)$ in (23) is such that

   a) $\bar{\mathcal{V}}(W,U)$ is bounded and locally Lipschitz
   b) $\dot{F} = \frac{1}{2}\alpha + 2\mu U(t) \circ \mathcal{V} \leq 0$, where $\alpha = 2e_N^T \Phi \circ W \circ \mathcal{F}(W,U)e_N$
   c) $\dot{F} \to 0$, $U \to 0$, and $\dot{L}_1 \to 0$ as $t \to \infty$.

Please refer to Appendix in Section VII for the proof of the above theorem.

Remark 3: From the above theorem we show that our control design $\dot{U} = \mathcal{V}$ achieves the objectives of (19), that is $\dot{F} \leq 0$, only when *there exists atleast one* Lipshitz control design $\bar{\mathcal{V}}$ such that $\dot{F} \leq 0$. The conditions under which such a $\bar{\mathcal{V}}$ exists is a difficult problem in itself and is part of our ongoing work. Essentially this theorem shows that our approach is not conservative; it guarantees $\dot{F} \leq 0$, whenever it is possible (by any control design) under the dynamics $\dot{W} = \mathcal{F}(W,U)$.

Remark 4: The proposed control design approach when applied to the case with the original (not modified) cost function, that is is with L in (6); yields similar results. In fact the resulting dynamic control law is Lipshitz and is such that $\dot{F} \leq 0$ under the assumption that there exists a $\bar{\mathcal{V}}$ that satisfies the conditions in the above theorem.

VI. ILLUSTRATIVE EXAMPLES

We demonstrate our algorithm for the example shown in Fig. 1b. The Algorithm 1 determines the partitioning $\{A_j\}_{j=1}^3$ as $A_1 = \{1,2,3\}$, $A_2 = \{8,9,10\}$, $A_3 = \{4,5,6,7\}$ and the resulting set of cut-edges as $\{(2,9),(3,4),(7,8)\}$. It is easy to verify that the above solution is optimal. Fig. 3 shows a typical run of our algorithm at increasing β values.

As seen in Fig. 3, at very low values of β, the association probability $p(j|i) \equiv 1/3$ for every $i \notin S$, which in turn corresponds to maximizing randomness (Shannon entropy) of the solution. However as the randomness is gradually decreased by increasing β, the probabilities start becoming *non-uniform* and exhibit preferential association to a specific terminal s_j. In the limiting case, i.e., at large values of

Illustrative Example of the proposed approach								
$\beta \approx 0$			$\beta \approx 0.96$			$\beta \approx 38.55$		
1.00	0	0	**1.00**	0	0	**1.00**	0	0
0.33	0.33	0.33	0.64	0.14	0.22	1.00	0	0
0.33	0.33	0.33	0.67	0.17	0.15	1.00	0	0
0.33	0.33	0.33	0.34	0.44	0.22	0	1.00	0
0.33	0.33	0.33	0.21	0.60	0.19	0	1.00	0
0	**1.00**	0	0	**1.00**	0	0	**1.00**	0
0.33	0.33	0.33	0.18	0.39	0.26	0	1.00	0
0.33	0.33	0.33	0.15	0.31	0.64	0	0	1.00
0.33	0.33	0.33	0.23	0.30	0.63	0	0	1.00
0	0	**1.00**	0	0	**1.00**	0	0	**1.00**

Fig. 3: Illustrative example of minimum 3-cut problem. The columns indicate the association matrices $p(j|i)$ at different β values. The 'bold' numbers indicate the associations of terminals, i.e., $p(j = 1|i = 1) = 1, p(j = 2|i = 6) = 1$ and $p(j = 3|i = 10) = 1$. As the algorithm progresses, the associations of remaining nodes harden, thereby resulting in optimal cut.

β, the algorithm results in hardened probabilities $(0 - 1$ associations). Thus an optimal cut is obtained. Note that in this example, β is increased *geometrically* from 0.01 to 40, i.e., the algorithm provides for very fast β scheduling.

Minimum multiway k cut for a non-planar graph: We now consider an instance of a 22-node *non planar* graph, shown in Fig. 4a with unit edge-weights. In this example we consider a 6-cut problem, whose set of terminals is specified as $S = \{1,4,8,11,15,19\}$. Our MEP algorithm results in a partition of the underlying graph, given by $\{1,2,3\}$, $\{4,5,6,7\}$, $\{8,9,10\}$, $\{11,12,13,14\}$, $\{15,16,17,18\}$ and $\{19,20,21,22\}$ with a cut value of 15; which is indeed optimal and can be easily verified. On the other hand, the isolating cut heuristic [4] for this randomly generated instance results in a cut solution with a value of 16. A similar observation is made on other randomly generated instances, where our algorithm results in optimal cut values (whenever verifiable). Moreover, the total run-time for the example in Fig. 4a for a naive implementation of the proposed approach in MATLAB is < 1s on an Intel i7-4790 CPU @ 3.60 GHz.

Non-unique optimal cuts: In both the examples described above, the resulting optimal cuts are indeed *unique*. We now therefore consider a scenario with more than one permissible optimal cuts. Our algorithm identifies the multiplicity of optimal cuts, which is reflected in the final association matrix $\{p(j|i)\}$. Fig. 4b shows an example of a 8-node graph with unit edge-weights and multiple permissible optimal 4-cuts . Executing the proposed MEP-based algorithm results in the following association matrix. As shown in Fig. 4(c), node 3 can be included in either A_2 or A_3 without affecting the value of the optimal cut, and therefore the association probabilities $p(j = 2|i = 3) = p(j = 3|i = 3) = 0.5$. Assigning node 3 to any of the two partitions still results in a feasible, yet optimal cut.

Comparison on challenging examples: Fig. 4d shows the performance of our MEP based approach on a toy-example (which can be generalized to any number of nodes. Here heuristic such as isolating cut [4], fails to identify an optimal cut for the simplest such scenario. In fact, it has been analytically shown that the isolating cut heuristic results can
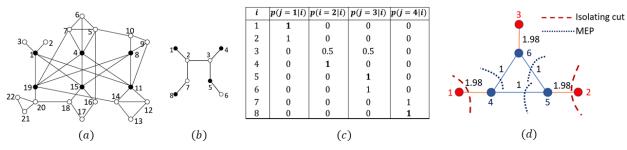
i	$p(j=1|i)$	$p(i=2|i)$	$p(j=3|i)$	$p(j=4|i)$
1	1	0	0	0
2	1	0	0	0
3	0	0.5	0.5	0
4	0	1	0	0
5	0	0	1	0
6	0	0	1	0
7	0	0	0	1
8	0	0	0	1

(a) (b) (c) (d)

Fig. 4: (a) Example of a 22 nodes non-planar graph with 6 terminals, $S = \{1,4,8,11,15,19\}$. The proposed algorithm results in an optimal cut. (b) An 8 node graph with multiple permissible optimal 4-cuts. (c) Final association matrix for the 4-cut problem shown in (b) part. (d) Toy-example with $2k = 6$ vertices where k of the vertices form a cycle where each edge weight is equal to 1, and each other vertex is connected to exactly one vertex on the cycle with an edge weight of 1.98. The isolating cut heuristic [4] results in a cut value of 3.96, where as our MEP-based algorithm results in optimal cut.

Fig. 5: Illustration of the proposed multiway k-cut approach to background-foreground segmentation. Here each pixel in the bounding box represents a node. The terminal nodes (corresponding to foreground and background) are interactively provided by the user once, and the algorithm successively evaluates minimum st-cut to refine foreground-background segmentation. Edge-weights are obtained as functions of pixels' RGB values.
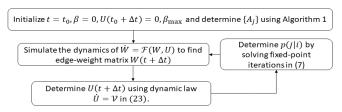

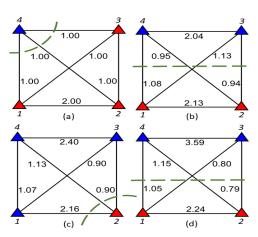Fig. 6: Algorithm to solve the multiway k-cut problem on Dynamic graphs.

Fig. 7: Illustration of the proposed approach for time varying minimum multiway 2-cut in an example graph with 4 nodes, where $s_1 = 2$ and $s_2 = 4$. The four plots indicate different stages of the evolution of edge-weights. Note that the edge $(1,2)$ is uncontrollable and evolves under its natural dynamics. On the other hand, other edges can be controlled to result in minimum cut values at each instant without having to explicitly compute cut values from the beginning. It is easy to verify that the cut values at each instant.

not give a solution within $2(1-1/k)$ of the optimal solution [4]. However, our algorithm finds the optimal multiway k-cut; thus achieving a cut which is impossible to obtain using the isolating cut method.

On Large Graphs: We also test our approach on large graphs with the number of nodes as large as ~ 25000 (corresponding to the size of the bounding box 150×160 pixels). Figure 4 shows the results of our implementation of the interactive foreground-background segmentation (GrabCut [22]) using the proposed MEP approach. In GrabCut, an image is represented as a graph of pixels (nodes), where edge-weights capture differences in intensities between neighboring pixels. Users are required to demarcate the approximate foreground region using a bounding box as shown in Fig. 5. The proposed approach is then employed to segment and refine the foreground through successive minimization of s-t cuts of the resulting graphs. The implementation results in effective segmentation of foreground and background

Dynamic Graphs: Fig. 6 summarizes the steps of our algorithm. Fig. 7 illustrates the time-varying minimum multiway 2-cut obtained on a weighted *undirected* dynamic graph using our proposed methodology. As stated in Section V the objective here is to find the time varying cuts A_1 and A_2 where the nodes s_1 and s_2 are the nodes 2 and 4 respectively. The edge-weight dynamics is given by $\dot{W}(t) = AW + BU$, which can be succinctly re-written as

$$\begin{pmatrix} \dot{w}(1,2) \\ \dot{w}(1,3) \\ \dot{w}(1,4) \\ \dot{w}(2,3) \\ \dot{w}(2,4) \\ \dot{w}(3,4) \end{pmatrix} = A' \begin{pmatrix} w(1,2) \\ w(1,3) \\ w(1,4) \\ w(2,3) \\ w(2,4) \\ w(3,4) \end{pmatrix} + B' \begin{pmatrix} u(1,2) \\ u(1,3) \\ u(1,4) \\ u(2,3) \\ u(2,4) \\ u(3,4) \end{pmatrix}, \text{where} \quad (24)$$

$$A' = \frac{5}{6}\text{diag}\left(e^{0.5},\ e^{1.8},\ e^{1.5},\ e^{-1.5},\ e^{-1},\ e^{3.5}\right), \text{ and}$$

$$B' = \text{diag}\left(0,\ 1,\ 1,\ 1,\ 1,\ 1\right). \quad (25)$$

Note that in the above system dynamics (25), the edge $(1,2)$ is incapable of being influenced by any external control input. We simulate the dynamical system (24) for a total of 0.075 seconds. Fig. 7(a) illustrates the minimum multiway 2-cut given by the Algorithm 1 at the initial time t_0 where $A_1 = \{1,2,3\}$, $A_2 = \{4\}$. As time progresses the partitioning changes to $A_1 = \{1,2\}$, $A_2 = \{3,4\}$ in Fig. 7(b), $A_1 = \{2\}$, $A_2 = \{1,3,4\}$ in Fig. 7(c) and $A_1 = \{1,2\}$, $A_2 = \{3,4\}$ in Fig. 7(d).

In the frame-by-frame approach we first discretize the entire time interval of 0.075 seconds with $\Delta t = 0.01$ seconds and set $U(t) = 0 \; \forall \; t$, then Algorithm 1 is used to obtain the minimum multiway 2-cut at each time instant. However this latter takes approximately 29 times more computational time than required by our proposed method thereby making the frame-by-frame approach unscalable even for a very short duration of time.

VII. CONCLUSION

In this paper, we formulate a statistical physics based algorithm for the minimum multiway k-cut problem on static as well as dynamic digraphs and present a convergence proof for the algorithm. The algorithm described in this paper is computationally efficient and has ability to avoid poor local minima through controlled randomness. We believe that a combination of good theoretical properties and experimental success of the proposed MEP-based algorithm makes it a suitable technique of choice for a wide variety of combinatorial optimization problems on graphs, such as, vertex coloring and finding independent sets.

APPENDIX

Proof of Theorem 1

1): The first part of $L_1(W(t))$ in (21) is positive since $\sum_j p(j|l)p(j|m) \leq (\sum_j p(j|l)^2)^{0.5}(\sum_j p(j|m))^{0.5} \leq (\sum_j p(j|l))^{0.5}(\sum_j p(j|m))^{0.5} = 1$, and the second part $\sum_{ij} p(j|i)\log p(j|i) \geq -N\log k$. Also since $\|U\|_{\mathrm{F}} \geq 0$ we have that $F + \frac{1}{\beta}N\log k \geq 0$.

2(a): Since $\bar{\mathscr{V}}$ is Lipschitz at $U = 0$, $\exists$ a δ neighborhood $B_\delta \triangleq \{U : \|U\| \leq \delta\}$ and $k > 0$ such that $\dot{F}(\bar{\mathscr{V}}) = \frac{1}{2}\alpha + 2\mu e_N^T U \circ \bar{\mathscr{V}} e_N \leq 0$, where $\alpha = 2e_N^T \Phi \circ W \circ \dot{W} e_N$, and $\|\bar{\mathscr{V}}\| \leq k\|U\| \; \forall \; U \in B_\delta$.

Case $\alpha > 0$: $\Rightarrow |\alpha| < |4\mu e_N^T U \circ \bar{\mathscr{V}} e_N| \leq \bar{k}\|U\|_{\mathrm{F}}^2$. From $\mathscr{V}$ in (23), we have

$$\|\mathscr{V}\|_{\mathrm{F}} \leq \left[C_0 + \frac{\bar{k}}{4\mu} + \frac{\sqrt{\bar{k}^2 + 16\mu^2}}{4\mu}\right]\|U\|_{\mathrm{F}} = \hat{k}\|U\|_{\mathrm{F}}$$

Case $\alpha < 0$: From (23)

$$\|\mathscr{V}\|_{\mathrm{F}} \leq (C_0 + \frac{\sqrt{\bar{k}^2 + 16\mu^2}}{4\mu})\|U\|_{\mathrm{F}} = \check{k}\|U\|_{\mathrm{F}}$$

2(b): Substituting the dynamic controller $\mathscr{V}(t)$ in (23) in the expression of $\dot{F}$ given by (22) we obtain $\dot{F} = -2\mu C_0\|U(t)\|_{\mathrm{F}}^2 - \sqrt{\alpha^2 + (4\mu\|U(t)\|_{\mathrm{F}}^2)^2}$ which is clearly non-positive.

2(c): $\dot{F} \to 0$ follows from Lasalle's Invariance Principle. From equation (23) $\frac{d(U^T U)}{dt} \leq -C_0 U^T U$ which implies

$U^T U \to 0$ exponentially using the Gronwall's inequality [23]. This in turn implies that $U \to 0$. Now since $\dot{L}_1 = \dot{F} - 2e_N^T \mu U(t) \circ \mathscr{V} e_N$ we have that $\dot{L}_1 \to 0$.

REFERENCES

[1] S. Lowen, "Multiway cut," 2011.

[2] H. S. Stone, "Multiprocessor scheduling with the aid of network flow algorithms," *IEEE transactions on Software Engineering*, no. 1, pp. 85–93, 1977.

[3] P. L. Erdős and L. A. Székely, "On weighted multiway cuts in trees," *Mathematical programming*, vol. 65, no. 1, pp. 93–105, 1994.

[4] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, "The complexity of multiterminal cuts," *SIAM Journal on Computing*, vol. 23, no. 4, pp. 864–894, 1994.

[5] D. Marx, "A tight lower bound for planar multiway cut with fixed number of terminals," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2012, pp. 677–688.

[6] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 27, no. 5, pp. 387–408, 2012.

[7] J. Thomas, D. Seo, and L. Sael, "Review on graph clustering and subgraph similarity based analysis of neurological disorders," *International journal of molecular sciences*, vol. 17, no. 6, p. 862, 2016.

[8] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

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

[10] P. V. Gehler and O. Chapelle, "Deterministic annealing for multiple-instance learning." in *AIStats*, 2007, pp. 123–130.

[11] S. Mitra, R. Castellanos, and S. Joshi, "An adaptive deterministic annealing approach for medical image segmentation," in *Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American*. IEEE, 2000, pp. 82–84.

[12] Y. Xu, S. M. Salapaka, and C. L. Beck, "Aggregation of graph models and markov chains by deterministic annealing," *Automatic Control, IEEE Transactions on*, vol. 59, no. 10, pp. 2807–2812, 2014.

[13] A. Rao, K. Rose, and A. Gersho, "Design of robust hmm speech recognizers using deterministic annealing," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 466–473.

[14] M. Baranwal, B. Roehl, and S. M. Salapaka, "Multiple traveling salesmen and related problems: A maximum-entropy principle based approach," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 3944–3949.

[15] Y. Xu, S. M. Salapaka, and C. L. Beck, "Clustering and coverage control for systems with acceleration-driven dynamics," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1342–1347, 2014.

[16] G. Călinescu, H. Karloff, and Y. Rabani, "An improved approximation algorithm for multiway cut," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 48–52.

[17] J. Naor and L. Zosin, "A 2-approximation algorithm for the directed multiway cut problem," *SIAM Journal on Computing*, vol. 31, no. 2, pp. 477–482, 2001.

[18] S. A. Gershgorin, "Uber die abgrenzung der eigenwerte einer matrix," *Proceedings of the Russian Academy of Sciences. Mathematical series*, no. 6, pp. 749–754, 1931.

[19] R. Sepulchre, M. Jankovic, and P. V. Kokotovic, *Constructive nonlinear control*. Springer Science & Business Media, 2012.

[20] E. D. Sontag, "A lyapunov-like characterization of asymptotic controllability," *SIAM Journal on Control and Optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[21] ——, "A universal construction of artstein's theorem on nonlinear stabilization," *Systems & control letters*, vol. 13, no. 2, pp. 117–123, 1989.

[22] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[23] T. H. Gronwall, "Note on the derivatives with respect to a parameter of the solutions of a system of differential equations," *Annals of Mathematics*, pp. 292–296, 1919.