Chapter 10
Dynamic Coverage and Clustering: A Maximum Entropy Approach

Carolyn Beck, Srinivasa Salapaka, Puneet Sharma and Yunwen Xu

Abstract We present a computational framework we have recently developed for solving a large class of dynamic coverage and clustering problems, ranging from those that arise in the deployment of mobile sensor networks to the identification of ensemble spike trains in computational neuroscience applications. This framework provides for the identification of natural clusters in an underlying dataset, while addressing inherent tradeoffs such as those between cluster resolution and computational cost. More specifically, we define the problem of minimizing an instantaneous *coverage* metric as a combinatorial optimization problem in a Maximum Entropy Principle framework, which we formulate specifically for the dynamic setting. Locating and tracking dynamic cluster centers is cast as a control design problem that ensures the algorithm achieves progressively better coverage with time.

Carolyn Beck
Coordinated Science Lab, University of Illinois at Urbana-Champaign & Department of Industrial and Enterprise Systems Engineering, University of Illinois, 117 Transportation Building MC-238, 104 S. Mathews Ave., Urbana, IL 61801, USA
e-mail: beck3@illinois.edu

Srinivasa Salapaka
Coordinated Science Lab, University of Illinois at Urbana-Champaign & Department of Mechanical Science and Engineering, 362c Mechanical Engineering Building, 1206 West Green Street, Urbana, IL 61801
e-mail: salapaka@illinois.edu

Puneet Sharma
Integrated Data Systems Department, Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540, USA
e-mail: sharma.puneet@gmail.com

Yunwen Xu
Coordinated Science Lab, University of Illinois at Urbana-Champaign & Department of Industrial and Enterprise Systems Engineering, University of Illinois, 1308 W Main Street Urbana, IL 61801-2307, USA
e-mail: xu27@illinois.edu

R. Johansson and A. Rantzer (Eds.): Distributed Decision Making and Control, LNCIS 417, pp. 215–243.
springerlink.com

10.1 Introduction

There has been a recent growing interest in the development of algorithms for deployment of mobile resources that continuously *cover* a set of mobile sites in a region, that is, algorithms related to determining clusters in an ensemble of moving objects. Such algorithms have numerous applications, such as in developing automatic deployment and tracking techniques for surveillance and military applications [4, 8], in clustering of the spatio-temporal dynamics of brain signals [18, 17, 1] and in routing traffic and detecting traffic jams by clustering traffic flow [30].

Clustering algorithms for *static data* have been widely studied in various areas, such as the minimum distortion problem in data compression [10], facility location assignments [5], optimal quadrature rules and discretization of partial differential equations [6], pattern recognition [33], drug discovery [28] and recently in coarse quantization [7, 16, 22]. However, the focus on clustering *dynamically evolving data* is new and many issues pertaining to quantification, analysis and design for coverage remain unsolved.

Although the static problems focus on seemingly unrelated goals, they, in fact, share a number of fundamental common attributes. They are typically formulated under a class of combinatorial optimization problems that searches for an optimal partition of the underlying domain, as well as an optimal assignment of elements, from a finite *resource* set to each *cell* in the partition. Since both the number of partitions as well as the number of element associations to partitions are combinatorial, these problems are computationally intractable (i.e., NP-hard [11]), which rules out exhaustive search methods. The cost functions in these optimization problems contain numerous local minima, and therefore it is crucial to design algorithms that do not get trapped in local minima [11].

The complexity of the combinatorial problem addressed in this chapter is further compounded by the *dynamics* associated with each data point, i.e., the moving objects. These moving objects could be, for example, mobile threat locations in a battlefield scenario, ensemble neuronal spike trains in brain signal data, formations of unmanned vehicles, or naturally occuring swarms or flocks. The task at hand is to design a dynamic control law for mobile resources such that they continuously identify and track cluster centers of groups of moving objects. Thus, locations of each data-point in the static case are replaced by velocity and/or acceleration fields in the dynamic case. From a naive computational viewpoint, the dynamic problem can be regarded as a time-indexed set of static problems. Adding dynamics also introduces new complexities to the notions of coverage due to the dynamic nature of cluster sizes, number of clusters, and relative distances between the individual elements.

Problems related to dynamic coverage were considered in [4, 8, 3, 30], in which distributed implementation approaches are used. Distributed schemes aim to address the issue of large impractical or nonviable computational efforts required for centralized schemes; however, these algorithms are prone to converge to one of the many local minima typically present in these problems. As a consequence, the performance of distributed algorithms is extremely sensitive to initial placement of the resource locations. At present, there is little research that addresses the development of tractable algorithms for dynamic problems of a nondistributed nature, that is, algorithms that aim simultaneously to attain global solutions and maintain low computational expense.

In this chapter, we present a general framework based on the maximum entropy principle (MEP) to formulate and solve dynamic clustering problems, which addresses both coverage and tracking. The framework we propose, which we refer to as the dynamic maximum entropy (DME) framework, adapts the notion of coverage to the dynamic setting and resolves the inherent trade-off between resolution of the clusters and computational cost, while avoiding shallow local minima. The algorithms we propose are hierarchical in that they progressively seek finer subclusters from larger clusters. An important feature of the proposed framework is its ability to detect natural cluster centers in the underlying data set, without the need to initialize or define the clusters a priori. The computational complexity of these algorithms is reduced by exploiting structure in the problem formulation. The algorithms, as they proceed, become more “local”, that is, the computation of clusters becomes less sensitive to distant sites. This feature is exploited to make these algorithms scalable and computationally efficient. Our simulation results, employing algorithms based in the DME framework, demonstrate improvements in computation time and illustrate the flexibility of this framework.

We note that an MEP-based approach has been developed in the static setting in the context of vector quantization (VQ) problems in the data-compression community. The resulting algorithm, known as the *deterministic annealing* (DA) algorithm, has been shown to avoid local minima, provide provably better clustering results than the popular *k*-means algorithms, and converge faster than other heuristics designed to avoid local minima such as the simulated annealing algorithm [21]. The DME framework we propose inherits the advantages of the DA algorithm and, further, addresses tracking as well as resolution issues originating from cluster dynamics.

This chapter is organized as follows. We discuss the general setting of the dynamic coverage problem in Sec. 10.2 and highlight key issues and challenges that must be resolved so such problems can be solved. We also provide an overview of the DA algorithm developed in [21] for clustering static data in this section and analyze those features that are relevant to the dynamic problem. The proposed DME framework for solving the clustering problem for dynamic data is presented in Sec. 10.3. Scalability issues are discussed in Sec. 10.4. Implementation and simulation results for a variety of data sets are presented in Sec. 10.5. Conclusions and directions for future work are discussed in Sec. 10.6.

10.2 Dynamic versus Static Clustering

Although the static clustering problem has been extensively studied, in this chapter we focus on the scenario where the problem is to detect and track a group of moving objects (henceforth referred to as *sites*) in a given area; that is, the *dynamic clustering problem*.

10.2.1 Problem Formulation

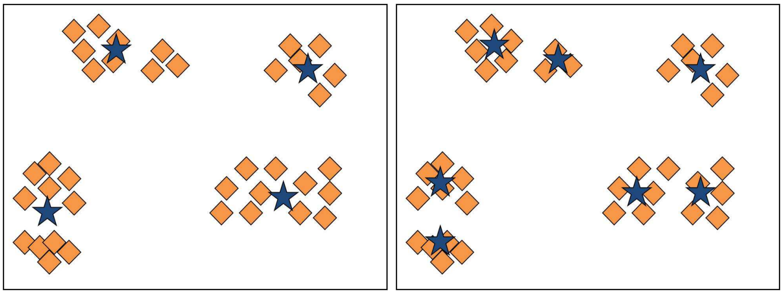
The mobile sites we study may move in clusters and change cluster associations, and the clusters themselves can split and rejoin over time. This problem can be posed as a coverage problem, where the aim is to successively identify representative object locations (in the sequel referred to as *resource locations*, or *cluster centers*) such that the resources provide adequate *coverage* of the moving sites at all times. The number of resources is assumed to be far fewer than the number of moving sites. Each resource can then be thought of as a cluster center.

For notational convenience, we consider a domain $\Omega \subset \mathbb{R}^2$ with N mobile sites and M resources, where $N \gg M$, on a time horizon in $[0,\infty)$. (Note that the approach we develop in this chapter is applicable to domains $\Omega \subset \mathbb{R}^k$, for any $k \in \mathbb{N}$; however, we will restrict our discussion herein to $k = 2$ for ease of exposition.) The location of the ith mobile site ($i \leq N$) and the jth resource ($j \leq M$) at time instance $t \in [0,\infty)$ is represented by $x_i(t) = [\xi_i(t)\ \eta_i(t)]^T \in \mathbb{R}^2$ and $y_j(t) = [\rho_j(t)\ \omega_j(t)]^T \in \mathbb{R}^2$, respectively. We will sometimes simply use x_i and y_j in place of $x_i(t)$ and $y_j(t)$, where the time dependence is assumed and is clear from the context. The dynamics are given by the continuously differentiable velocity fields, $\phi_i(x,y,t) \in \mathbb{R}^2, i \leq N$, for the ith site and $u_j(t) \in \mathbb{R}^2, j \leq M$, for the jth resource, where x and y represent the locations of all sites and resources, respectively. More precisely, we have a domain Ω with N sites $\{x_i\}$ and M resource locations $\{y_j\}$, whose dynamics are given by

$$\begin{aligned} \dot{x}(t) &= \phi(x(t), y(t), t), \quad x(0) = x_0 \\ \dot{y}(t) &= u(t), \quad y(0) = y_0 \\ &\Updownarrow \\ \dot{\zeta} &= f(\zeta, t), \end{aligned} \tag{10.1}$$

where $x(t) = [x_1(t)\ x_2(t)\ \cdots\ x_N(t)]^T$, $y(t) = [y_1(t)\ y_2(t)\ \cdots\ y_M(t)]^T$, $\phi(t) = [\phi_1(t)\ \phi_2(t)\ \cdots\ \phi_N(t)]^T$, $u(t) = [u_1(t)\ u_2(t)\ \cdots\ u_M(t)]^T$, and $\zeta(t) = [x^T\ y^T]^T$. This system can be viewed as a control system where the control field u is to be determined for the M mobile resources such that a notion of coverage is maintained. We summarize the main challenges and our objectives in addressing dynamic coverage problems in the following:

- One of the main challenges in dynamic coverage problems is fundamental: quantifying the performance objectives. We adopt the concept of *distortion* and its variants from the data compression literature (which deals with static coverage problems) as a metric for coverage and modify this to make it suitable to a dynamic setting. Distortion, in a static coverage problem, is a measure of the (weighted) average distance of the site locations to their nearest resource location. For a static problem ($\phi(t) \equiv 0, u(t) \equiv 0$), the distortion measure is given



(a) (b)

Fig. 10.1 The clustering solution in (a) identifies 4 coarse clusters centers $y_j, 1 \leq j \leq 4$ (shown by stars) in the underlying data $x_i, 1 \leq i \leq 37$ (shown by squares). On the other hand, the solution in (b) identifies 7 finer clusters at a higher resolution on the same underlying data. The solution in (b) has lower distortion D than in (a), but at the expense of higher computation time.

by

$$D(x,y) = \sum_{x_i \in \Omega} p_i \left\{ \min_{1 \leq j \leq M} d(x_i, y_j) \right\}, \tag{10.2}$$

where p_i represents the weight or relative importance of the site x_i and $d(x_i, y_j)$ is a metric defined on Ω, which represents a distance function; this is typically given by a squared Euclidean distance function $d(x_i, y_j) = \|x_i - y_j\|^2$. Thus, for a given set of site locations $\{x_i\}$, $1 \leq i \leq N$, the set of resource locations $\{y_j\}$, $1 \leq j \leq M$ that achieves lower distortion results in better coverage.

- A second challenge arises in defining clusters. This challenge primarily stems from the fact that clusters are hierarchical in nature. Each cluster can be thought of as smaller subclusters (Fig. 10.1), and in the limiting case comprising every moving site can be thought of as a distinct cluster. On the other hand, the entire domain can be thought of as a cluster. Thus, we consider assigning a notion of *resolution* for a cluster. Iterative algorithms that identify finer and finer (higher resolution) clusters progressively reduce the coverage cost function, but at the expense of increased computation.

  Modeling the coverage function and defining clusters is even more challenging in the dynamic case, since the coverage function must adjust to and appropriately reflect the expanding, shrinking, splitting and/or coalescing components of clusters, adding further variability to the control design.
- Computational complexity is a third important challenge confronting the solution of coverage problems, arising from the inherent nonconvex nature of the distortion function (10.2). The dynamic distortion function contains numerous local minima, just as it does in the static case. Consequently, the problem necessitates designing algorithms that do not get trapped at local minima, while also avoiding expensive divide-and-search strategies. This complexity is further aggravated

by the additional time component in the dynamic setting. These challenges are addressed by the framework proposed in this section. More specifically we formulate a coverage problem and propose algorithms that determine the resource velocity fields $u(t)$, such that they *track* each cluster over time. If a cluster splits, the resource locations mimic this behavior, thereby maintaining coverage.

10.2.2 The Static Resource Allocation Approach

Dynamic data may be viewed as a time-indexed series of static data, thus the simplest approach conceptually is to perform static clustering periodically. However, if the elapsed time between two successive clustering events is small, the algorithm is unnecessarily expensive because the spatial clustering from previous time steps is not exploited for clustering at current and future time steps. On the other hand, if the time period is long the clustering obtained at the previous time step does not provide adequate tracking during the interval to the current time step. In short, such a simplistic approach ignores the spatio-temporal aspects of clustering moving data. Nevertheless, we provide a brief overview of this approach to provide insights into the challenges that are inherited by the dynamic problems. In this *frame-by-frame approach*, at each instant of time t, we solve the following static resource allocation problem:

Given N sites $x_i(t), 1 \leq i \leq N$, in a domain Ω with relative weights p_i, find the set of M resource locations $y_j(t), 1 \leq j \leq M$, that minimizes the distortion (at fixed time t); that is, find

$$\underset{y_j,\ 1\leq j\leq M}{\arg\min} \left(\sum_{i=1}^{N} p_i \left\{ \min_{1\leq j\leq M} d(x_i,y_j) \right\} \right) \tag{10.3}$$

Again, $d(x_i,y_j)$ represents an appropriate distance function between the resource location y_j and the site x_i, for example, $d(x_i,y_j) = \|x_i - y_j\|^2 + \|\phi_i - u_j\|^2$, where velocity terms as well as location terms are included in the Euclidean distance function. Minimizing (10.3) is akin to finding a velocity field for the resources such that the coverage condition is satisfied at time t.

This problem is equivalent to finding an *optimal* partition of the domain space Ω at time t into M cells ($R_j, j = 1 \ldots M$) and assigning to each cell R_j a resource location y_j that minimizes the partition cost, given by $\sum_j \sum_{x_i \in R_j} d(x_i,y_j)p_i$. Solving this partitioning problem at a fixed time t is equivalent to the vector quantization problem in data compression, which is addressed by the DA algorithm [21, 20].

We now describe relevant features of the DA algorithm [19], which provide a basis for our DME algorithm.

10.2.3 The Deterministic Annealing Algorithm: Clustering in the Static Setting

One of the main objectives of the DA algorithm is to avoid local minima. This algorithm can be viewed as a modification of Lloyd's algorithm [15, 10], in which the initial step consists of randomly choosing resource locations and then successively iterating between the steps of: (1) forming Voronoi partitions and (2) moving the resource locations to respective centroids of cells until the sequence of resource locations converges. The solution depends substantially on the initial placement of resources, as in successive iterations the locations are influenced only by nearby points of the domain and are virtually independent of distant points. As a result, Lloyd's algorithm typically converges to local minima.

The DA algorithm overcomes the local influence of domain elements by allowing each element $x_i \in \Omega$ to be associated with every resource location y_j through a weighting parameter $p(y_j|x_i)$ (without loss of generality, $\sum_j p(y_j|x_i) = 1$ for every $1 \leq i \leq N$) [21, 20, 26, 27]. Thus, the DA algorithm eliminates the hard partitions of Lloyd's algorithm and seeks to minimize a modified distortion term given by

$$D(x,y) = \sum_{i=1}^{N} p_i \sum_{j=1}^{M} d(x_i,y_j)p(y_j|x_i). \tag{10.4}$$

Note that the instantaneous weighting term $p(y_j|x_i)$ (at time t) can also be viewed as an association probability between the mobile site x_i and the mobile resource y_j. The choice of weights $\{p(y_j|x_i)\}$ determines the trade-off between decreasing effects of local influence and deviation of the distortion (10.4) from the original cost function (10.2). For instance, a uniform weighting function, where $p(y_j|x_i) = 1/M$ for all $i \leq N, j \leq M$, makes the minimization of (10.4) with respect to y_j independent of initial placement of y_j; however, the corresponding distortion function is considerably different from the cost function in (10.2). At the other extreme, setting $p(y_j|x_i) = 1$ when $d(x_i,y_j) = \min_k d(x_i,y_k)$, but otherwise setting $p(y_j|x_i) = 0$, makes the distortion term in (10.4) identical to the cost function (10.2) but retains the "local influence effect" when minimizing with respect to y_j.

The MEP provides a systematic way to determine a weighting function that achieves a specific feasible value of distortion, and thereby achieves a prespecified tradeoff in the above context [12, 13]. More specifically, we seek a weight distribution $p(y|x)$ that maximizes the Shannon entropy [25]

$$H(y|x) = -\sum_{i=1}^{N} p_i \sum_{j=1}^{M} p(y_j|x_i)\log(p(y_j|x_i)), \tag{10.5}$$

at a given (feasible) level of coverage $D(x,y) = D_0$. The entropy term quantifies the level of randomness in the distribution of association weights, so maximizing this term causes the weights to be maximally noncommittal toward any single cluster. In this framework, we first determine the weighting functions by maximizing the unconstrained Lagrangian $L = H(y|x) - \beta(D(x,y) - D_0)$ with respect to $\{p(y_j|x_i)\}$, where $H(y|x)$ and $D(x,y)$ are given by (10.5) and (10.4) respectively, D_0 is the desired value of distortion and β is a Lagrange multiplier. This is equivalent to the minimization problem:

$$\min_{\{p(y_j|x_i)\}} \underbrace{D(x,y) - TH(y|x)}_{\stackrel{\triangle}{=} F}, \tag{10.6}$$

where the Lagrange multiplier, denoted by $T = 1/\beta$, and the term F are called *temperature* and *free energy*, respectively, analogously to quantities in statistical physics [14]. The MEP theorem (see [12, 13]) gives an explicit solution for the weights, given by the Gibbs distribution

$$p(y_j|x_i) = \frac{\exp\{-\beta d(x_i,y_j)\}}{\sum_{k=1}^{M} \exp\{-\beta d(x_i,y_k)\}} \tag{10.7}$$

On substituting (10.7) into the Lagrangian (10.6) we have

$$F(x,y) = -\frac{1}{\beta}\sum_{i=1}^{N} p_i \log \sum_{k=1}^{M} \exp\{-\beta d(x_i,y_k)\}. \tag{10.8}$$

This function plays an important role in representing the coverage function (as will be shown later). The resource locations $\{y_j\}$ are specified by setting $\partial F/\partial y_j = 0$, which yields

$$y_j = \sum_{i=1}^{N} p(x_i|y_j)x_i, \ \forall j = 1,2,\ldots,M, \text{ with } p(x_i|y_j) = \frac{p_i p(y_j|x_i)}{\sum_{m=1}^{N} p_m p(y_j|x_m)} \tag{10.9}$$

where $p(x_i|y_j)$ denotes the posterior association weight calculated using Bayes's rule. The above equations clearly convey the "centroid" aspect of the solution.

The temperature variable $T = 1/\beta$ is fixed by the constraint value D_0 of the distortion. A simple sensitivity analysis shows that lower values of D_0 correspond to lower values of the temperature variable [13]. Clearly, for small values of β (i.e., large values of T) in (10.6), we are mainly maximizing the entropy. Thus a choice of weights corresponding to a high value (near infinity) of T leads to algorithms that are insensitive to the initial allocation of resource locations, since their subsequent locations are affected almost equally by all sites. As β is increased, we trade entropy for the reduction in distortion, and as β approaches infinity (i.e., T approaches zero), we minimize distortion D directly to obtain a *hard* (nonrandom) solution. An *annealing* process is incorporated where the minimization problem (10.6) is repeatedly solved at different values $\beta = \beta_k$, where $\beta_{k+1} > \beta_k$. Solving the implicit equation (10.9) to determine y_j entails the most computationally expensive step for each value of β_k. This equation is solved by evolving the following dynamical system to convergence:

$$y_j^k(n+1) = \sum_{i=1}^{N} p(x_i|y_j^k(n))x_i, \ \ 1 \le j \le M, \ n \ge 0, \tag{10.10}$$

where $y_j^k(n)$ represents the value of the estimate of y_j (when the temperature value is given by $\beta = \beta_k$) at the nth step of this iterative procedure. At each k, the initial value $y_j^k(0)$ is set to the final value for $k-1$, that is $y_j^k(0) = y_j^{k-1}(\infty)$. Note that the iterative process (10.10) is equivalent to a Newton descent method and, accordingly, this procedure inherits the convergence properties of a Newton descent method.

The annealing process itself exhibits a *phase transition* property. That is, there are critical values of β at which the number of *distinct* resource locations abruptly change. At $\beta = 0$, there is only one resource at the weighted centroid of all site locations x_i. As the parameter β is increased, there exists some critical value β_c such that the number of distinct resource locations that minimize free energy jumps from $m = 1$ for $\beta < \beta_c$ to some $m > 1$ for $\beta > \beta_c$. This critical value, β_c, can be computed explicitly from the distribution of the sites x_i and resource locations y_j. As β is increased further, successive critical values are reached that also satisfy the phase transition condition. At each such critical value, there is an increase in the number of the distinct resource locations. A detailed version of the implementation steps is presented [21].

10.2.4 Properties of the DA Algorithm

The features developed for the static coverage algorithm form the basis for our DME framework. In this formulation, various cluster attributes, such as relative size and shape, arise naturally, thereby preemting the need for new characterization variables. The weights $\{p(x_i|y_j)\}$ characterize soft clusters; a high value of $p(x_i|y_j)$ implies the site x_i predominantly belongs to jth cluster, which is covered by the resource at location y_j. The weight $p(x_i|y_j)$, when viewed as a function of x_i for a fixed y_j, determines the shape of the jth cluster. Cluster mass is characterized by the set of weights $\{p(y_j)\}$. Since $p(y_j) = \sum_i p_i p(y_j|x_i)$, this represents the *total mass* associated with the resource y_j. That is, $p(y_j)N$ is an estimate of the number of sites that determine the jth cluster. In this work, we assume $\min_j\{p(y_j)\} > \nu > 0$ to avoid modeling degenerate (or zero-mass) clusters. As the annealing parameter $\beta \to \infty$, the weights $\{p(y_j|x_i)\}$ become binary valued and the resulting clusters become "hard".

Notation: We define matrices

$$P_x \overset{\triangle}{=} \mathrm{diag}(p(x_i)) \in \mathbb{R}^{N\times N}, \ P_{y|x} \overset{\triangle}{=} [p(y_j|x_i)]$$

and

$$P_{x|y} \overset{\triangle}{=} [p(x_i|y_j)] \in \mathbb{R}^{N\times M}, \quad P_y \overset{\triangle}{=} \mathrm{diag}(p(y_j)) \in \mathbb{R}^{M\times M}$$

. In this notation, $P_{x|y}$ and P_y contain information about the relative shapes and sizes of the clusters. Also note that

$$P_x P_{y|x} = P_{x|y} P_y = P_{xy} = [p(x_i, y_j) = p_i p(y_j|x_i)]$$

The expression for cluster centers, given by (10.9), can be written concisely as $y = \check{P}^T_{x|y} x$ where

$$\check{P}_{x|y} \triangleq (I_2 \otimes P_{x|y})$$

I_2 is the 2×2 identity matrix, and $\otimes$ represents the matrix Kronecker product. Similarly, we define matrices

$$\check{P}_{y|x} \triangleq (I_2 \otimes P_{y|x}), \;\; \check{P}_x \triangleq (I_2 \otimes P_x), \;\; \check{P}_y \triangleq (I_2 \otimes P_y), \text{ and } \check{P}_{xy} \triangleq (I_2 \otimes P_{xy})$$

The *radius* of each cluster can be inferred from the magnitudes of the vectors $x_i - y_{c[i]}$ $(1 \leq i \leq N)$, where $y_{c[i]}$ denotes the resource location closest to the site x_i. We define

$$\bar{x} \triangleq x - \check{P}_{y|x} \check{P}^T_{x|y} x$$

which determines the weighted average distance of x_i from the cluster centers $\{y_j\}$; that is,

$$\bar{x}_i = x_i - \sum_j p(y_j|x_i) y_j$$

Additional important features of the DA algorithm are summarized in the following.

Theorem 10.1. *The following properties hold for the Deterministic Annealing algorithm:*

1. Centroid property [21]: $\lim_{\beta \to 0} y_j = \sum_{i=1}^N p_i x_i$.
2. Phase transition Property [21]: *The resource locations* $\{y_j\}$ *given by (10.9) give a local minimum for free energy* F *at every value of* β *except at* critical temperatures *when* $\beta = \beta_c$*, given by* $\beta_c^{-1} = 2\lambda_{max}(C_{x|y_j})$ *for some* $1 \leq j \leq M$*, where*

$$C_{x|y_j} = \sum_{i=1}^N p(x_i|y_j)(x_i - y_j)(x_i - y_j)^T \tag{10.11}$$

   and $\lambda_{max}(\cdot)$ *represents the largest eigenvalue. Moreover, the number of distinct locations in* $\{y_j(\beta)\}$ *for* $\beta > \beta_c$ *is greater than for* $\beta < \beta_c$*.*
3. Sensitivity-to-temperature property [29]: *If the parameter value* β *is far from the critical values* β_c*, that is the minimum eigenvalue of* $(I - 2\beta C_{x|y_j}) \geq \Delta$ *for some* $\Delta > 0$ *and* $1 \leq j \leq M$*, then*

$$\|\frac{dy}{d\beta}\| \triangleq \left(\sum_j \|\frac{dy_j}{d\beta}\|^2 \right)^{\frac{1}{2}} \leq \frac{c(\beta)}{\Delta}$$

where $c(\beta)$ monotonically decreases to zero with β and is completely determined by β and the size of the space Ω.

Sketch of Proof: Proofs for properties 1 and 2 can be found in [21]). Herein we provide a sketch of a proof for property 3; full details can be found in [29].

We obtain $dy_j/d\beta$ by differentiating (10.9) with respect to β. Premultiplying $dy_j/d\beta$ by

$$p(y_j)\frac{dy_j}{d\beta}^T$$

and summing over j gives

$$\underbrace{\sum_j p(y_j)\frac{dy_j}{d\beta}^T\left[I-2\beta C_{x|y_j}\right]\frac{dy_j}{d\beta}}_{T_1}$$

$$\begin{aligned} &= \underbrace{\sum_i p_i \sum_j p(y_j|x_i)(x_i-y_j)^T\frac{dy_j}{d\beta}\left\{\sum_k p(y_k|x_i)d(x_i,y_k)-d(x_i,y_j)\right\}}_{T_2} \\ &- \underbrace{2\beta\sum_i p_i\sum_j p(y_j|x_i)(y_j-x_i)^T\frac{dy_j}{d\beta}\sum_k p(y_k|x_i)(y_k-x_i)^T\frac{dy_k}{d\beta}}_{T_3}. \end{aligned}$$

Since T_3 is nonnegative, and $T_1+T_3=T_2$, we have $T_2 \geq T_1$, which in turn implies

$$T_2 \geq \min_j\{p(y_j)\}\Delta\sum_j\frac{dy_j}{d\beta}^T\frac{dy_j}{d\beta} = \min_j\{p(y_j)\}\Delta\left\|\frac{dy}{d\beta}\right\|^2. \qquad (10.12)$$

Also, since $\sum_k p(y_k|x_i)=1$, T_2 can be rewritten as

$$T_2 = \sum_i p_i\sum_j\sum_k p(y_j|x_i)p(y_k|x_i)\left[d(x_i,y_k)-d(x_i,y_j)\right](y_j-x_i)^T\frac{dy_j}{d\beta}. \quad (10.13)$$

To obtain an upper bound on T_2, we first obtain a bound on the association weights,

$$p(y_j|x_i) = \frac{e^{-\beta(d(x_i,y_j)-d(x_i,y_k))}}{1+\sum_{m\neq k}e^{-\beta(d(x_i,y_m)-d(x_i,y_k))}} \leq e^{-\beta(d(x_i,y_j)-d(x_i,y_k))}. \qquad (10.14)$$

Since $\|x-y_j\| \leq 2w_1 \overset{\triangle}{=} 2$ diameter(Ω) and

$$|e^{-\beta(d(x_i,y_m)-d(x_i,y_k))}(d(x_i,y_m)-d(x_i,y_k))| \leq \frac{e^{-1}}{\beta},$$

(since $\theta e^{-\gamma\theta} < \frac{e^{-1}}{\gamma}$ for $\gamma > 0$), we can infer a bound on T_2 given by

$$T_2 \leq \frac{e^{-1}}{\beta}(2w_1)\sum_j \left\| \frac{dy_j}{d\beta} \right\| \leq \left(\frac{e^{-1}}{\beta}(2w_1)\sqrt{M} \right) \left\| \frac{dy}{d\beta} \right\| \tag{10.15}$$

From (10.12) and (10.15), we have that

$$\min_j \{p(y_j)\} \Delta \| \frac{dy}{d\beta} \|^2 \leq |T_2| \leq \left(\frac{e^{-1}}{\beta}(2w_1)\sqrt{M} \right) \left\| \frac{dy}{d\beta} \right\|,$$

which implies that

$$\left\| \frac{dy}{d\beta} \right\| \leq \frac{c(\beta)}{\Delta}, \text{ where } c(\beta) = \left(\frac{e^{-1}}{\nu\beta}(2w_1)\sqrt{M} \right)$$

is completely determined by the value of β and the bound of the size of the space Ω. Here ν is a lower bound on $\min_{y_j}\{p(y_j)\}$, the minimum cluster size and w_1 is the diameter of space Ω. □

The centroid property implies that for small values of β the DA algorithm places all resources at the weighted centroid of the sites, that is, at $\beta = 0$ the cost function (10.6) achieves the global minimum with $p(y_j|x_i) = 1/M$ and with all resource locations $\{y_j\}$ being placed at the centroid of the data set. The annealing process deforms the free energy F from the entropy function at $\beta = 0$ to the distortion function at $\beta = \infty$, which allows us to track the evolution of the global minimum as β is increased.

The phase transition property guarantees that the resource locations obtained at noncritical β values are local minima of F. In this sense, the performance of this algorithm is as good as or better than related algorithms such as Lloyd's algorithm. As $\beta \to \infty$, the DA algorithm essentially converges to Lloyd's algorithm, with the choice of initial placement of locations (through the annealing process) designed to achieve better minima. The phase transition property also obviates the hierarchical nature of the algorithm. When the value of β crosses a critical threshold value, β_c, the number of *distinct* resource locations jumps; this is referred to as a *splitting* process and is used explicitly to identify *natural* clusters in the data.

The sensitivity-to-temperature property implies that in the static algorithm—the rate of change of the resource locations between two critical temperatures—can be bounded above. The condition on the minimum eigenvalue of $I - 2\beta C_{x_i|y_j}$ for $1 \leq j \leq M$ being bounded away from zero corresponds to non-critical temperature values. The bounds given in this theorem are conservative; better bounds can be obtained by imposing additional assumptions on the data [29]. This property has important consequences—namely the rate at which we change temperature values, that is the *cooling rate*, can be high—and typically we increase β geometrically, that is

$$\beta_k = \gamma^k \beta_0 \text{ for some } \gamma > 1. \tag{10.16}$$

It is this property that forms the basis for extending the MEP-based framework to the dynamic setting.

Remark 10.1. Recall that the static problem is NP-hard. The cooling law for the annealing process in the DA algorithm is typically geometric, as is indicated in (10.16), and therefore avoiding local minima comes at a cost of only a few iteration steps (in comparison, other annealing procedures such as simulated annealing require much slower ($O(\log N)$) cooling rates [9]). The MEP-based framework has additional flexibility, for example, we have proposed adoptings constraints that reflected computational expense and obtained weight functions that accommodated for this constraint. The resulting scalable algorithms were $75\% - 80\%$ more efficient in terms of computation time than the original algorithm.

10.3 The Dynamic Maximum Entropy Framework

In this section, we present our general framework for formulating the dynamic clustering problem and determining computationally efficient algorithms that resolve issues of numerous local minima, quantifying coverage and cluster resolution, spatio-temporal smoothening, achieving trade-offs between computational cost and resolution and tracking dynamically evolving clusters. As noted earlier, using a frame-by-frame approach is not computationally viable as it requires multiple iterations at each time step.

We use free energy as the metric for coverage, which together with an annealing process provides a basis for an algorithm that is independent of the initial choice of resource locations and avoids local minima. The instantaneous cluster center z^c, which is determined solely by site locations x_i (see (10.9)), is given by the recursion $z^c = \check{P}^{cT}_{x|y}x$, where $P^c_{x|y} = [p(x_i|z^c_j)] \in \mathbb{R}^{N\times M}$ (see Sec. 10.2.4 for details on notation). Note we represent the cluster center by z^c to distinguish it from the instantaneous resource location y. Since cluster centers, shapes, and mass are specified by

$$z^c = \check{P}^{cT}_{x|y}x,\ \check{P}^c_{x|y} \text{ and } \check{P}^c_y$$

respectively (see Sec. 10.2.4), the cluster-drift, intra-cluster, and inter-cluster-interaction dynamics are quantified by ϕ, $\dot{\check{P}}_{x|y}{}^{c}$, and $\dot{\check{P}}_{y}{}^{c}$, respectively.

In adapting the static concepts for the dynamic setting we must determine appropriate cooling rates (as in (10.16)) relative to the time scales of the site dynamics. For example, if the rate of cooling is much faster than the given dynamics of the sites, the resulting algorithm is similar to the frame-by-frame approach. However, from the sensitivity-to-temperature property in Theorem 10.1, we know the resource locations are insensitive to temperatures between two successive critical temperatures. Critical temperatures are indicative of splits and resolution, thus decreasing temperature values matter only for forcing these splits or obtaining higher resolution. Note that in the dynamic setting, the cluster splits at time t are still identified by critical temperatures given by $\beta_c^{-1} = 2\lambda_{\max}(C_{x(t)|y_j(t)})$ (once the resource locations $y_j(t)$ are at cluster centers). However, unlike the static case, these splitting conditions can be reached due to dynamics of x and y, in addition to decreasing values of β.

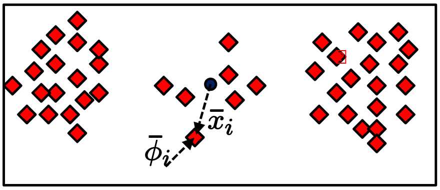


Fig. 10.2 The condition on the velocity field ϕ for consistent clusters. The solid circle represents the weighted cluster centroid x_i^c that is closest to the ith site with velocity ϕ_i, i.e., $\bar{\phi}_i = p_i\phi_i$. The condition $\bar{\phi}^T\bar{x}^c < 0$ implies that the angle between $\bar{x}_i^c$ and $\bar{\phi}_i$ is obtuse (in an average sense) and hence the individual sites are directed toward the cluster and not away from it.

This interpretation of the sensitivity-to-temperature property allows us to separate the dynamic clustering problem into two subproblems: (1) tracking cluster centers; and (2) monitoring splitting conditions. In our implementations, we split the resource locations only after they have reached the cluster centers. We monitor the cluster splits that result due to site dynamics or the cooling law. The resulting algorithm, as well as the incorporation of user-specified decisions on splits and resolutions, are presented later in this section.

Our main tracking results are briefly summarized in the following.

1. Under the assumption that the site dynamics given by $\phi(\zeta,t)$ in (10.1) are continuously differentiable, we provide a control law $u(\zeta)$ such that the resource locations asymptotically track the cluster centers (see Theorem 10.3).
2. We show this control law is nonconservative, that is, if $u(\zeta)$ is not bounded then there *does not exist any* Lipschitz control law that achieves asymptotic tracking for the system given by (10.1) (see Theorem 10.4).
3. If we further assume the clusters are *consistent* (that is, the average cluster size is nonincreasing and clusters are separated), then the prescribed control that achieves asymptotic tracking is *bounded* (see Theorem 10.5).

The results stated in 1 and 2 require only mild assumptions on the site dynamics. For the result in 3 we require the clusters to be consistent, that is, the distance from a site location x_i to the closest cluster center z_j^c does not increase with time. We impose this constraint, albeit in an average sense, by assuming that ϕ satisfies

$$\bar{\phi}^T\bar{x}^c \le 0, \text{ where } \bar{\phi} = \check{P}_x^c\phi \text{ and } \bar{x}^c = x - \check{P}_{y|x}^c\check{P}_{x|y}^{cT}x$$

Since the cluster center $z^c = \check{P}_{x|y}^{cT}x$, $\bar{x}^c$ denotes the average weighted vector from each site location x_i to these cluster centers (with the nearest cluster center weighted heavily for large β), consequently, $\bar{\phi}^T\bar{x}^c \le 0$ implies that velocity ϕ is such that magnitudes of these vectors are nonincreasing (see Fig. 10.2). We first elaborate on properties of the free energy term and its time derivative, which form the basis for addressing both the problems of tracking cluster centers and monitoring splitting conditions.

10.3.1 The Free Energy Term

The crux of the DA algorithm for the static setting is that it replaces the notion of distortion in (10.2) by the free energy in (10.8) as a metric for coverage. In order to achieve the objective of tracking the cluster centers without resorting to the frame-by-frame approach, we formulate a control problem, where we design the velocity $u(t)$ for the resource locations such that the time derivative $\dot{F}$ of the free energy function is nonpositive along the trajectories of $x(t)$ and $y(t)$. Such a formulation not only addresses the drawbacks of the frame-by-frame clustering approach, but also preserves the advantages of the DA algorithm found with static data. Moreover, the condition under which control authority is lost, that is, where $\partial F/\partial y = 0$, provides explicit connections to the splitting conditions and cooling laws encountered in the static setting. We summarize the properties of free energy and its time derivative in the following theorem. For a proof of these properties see [29].

Theorem 10.2. *Let F given by (10.8) be the free energy for the sites and resources x_i, $1 \leq i \leq N$, and y_j, $1 \leq j \leq M$, whose dynamics are defined by (10.1). Then*

1. Positivity: $F(\zeta) + \frac{1}{2\beta}\log M > 0$ *for all* ζ *in* $\left(\mathbb{R}^2\right)^{N+M}$.
2. Structured derivative: *The derivative of the free energy term has the following structure:*

$$\dot{F} = 2\zeta^T \Gamma(\zeta) f(\zeta), \ \ \Gamma = \begin{pmatrix} \check{P}_x & -\check{P}_{xy} \\ -\check{P}_{xy}^T & \check{P}_y \end{pmatrix}. \tag{10.17}$$

   The matrix Γ is a symmetric positive semidefinite matrix for all ζ, which can be decomposed as $\alpha(I - W)$ where $\alpha > 0$, I is the identity matrix and W is a symmetric doubly stochastic matrix with spectral radius $\rho(W) = 1$.
3. Loss of dynamic control authority at cluster centers: *The derivative $\dot{F}$ becomes independent of the control, that is, $\partial(\dot{F})/\partial u = 0$ (or equivalently the partial derivative $\partial F/\partial y = 0$) only at those time instants t_c when the resource locations $y_j(t_c)$ are coincident with the cluster centers, that is, only when $y_j(t_c) = \sum_{i=1}^{N} p(x_i(t_c)|y_j(t_c))x_i(t_c)$ for $1 \leq j \leq M$.*

10.3.2 Control Design: Tracking Cluster Centers

The resource locations coincide with the cluster centers only when $y_j - \sum_i p(x_i|y_j)x_i$ is zero for each j, that is, when

$$\bar{y} \overset{\triangle}{=} y - \check{P}_{x|y}^T x = 0$$

For design of u, we transform the coordinates $\zeta = (x,y)$ to $\bar{\zeta} = (x,\bar{y})$, in which the dynamics in (10.1) and $\dot{F}$ given by (10.17) are rewritten as

$$\left.\begin{array}{rcl}\dot{x}(t) & = & \phi,\\ \dot{y}(t) & = & u-\dot{\check{P}}_{x|y}^{\;T}x-\check{P}_{x|y}^{T}\phi\end{array}\right. \iff \dot{\bar{\zeta}}=\bar{f}(\bar{\zeta},t) \text{ and } \dot{F}=\bar{x}^{T}\bar{\phi}+\bar{y}^{T}\check{P}_{y}\bar{u} \qquad (10.18)$$

where $\bar{u}=u-\check{P}_{x|y}^{T}\phi$. We exploit the affine dependence of $\dot{F}$ in (10.18) on $\bar{u}$ to make $\dot{F}$ nonpositive analogous to control design based on control Lyapunov functions [24, 31, 32]. More specifically we choose $\bar{u}$ from sets of the form:

$$\bar{\mathcal{U}}(\alpha)=\{\bar{u}:\left(\mathbb{R}^{2}\right)^{N+M}\rightarrow\left(\mathbb{R}^{2}\right)^{M}\mid \bar{u}(\bar{\zeta})=-[K_{0}+\frac{\alpha(\bar{\zeta})+\theta(\bar{\zeta})}{\bar{y}^{T}\check{P}_{y}\bar{y}}]\bar{y},\ \bar{y}\neq 0\} \qquad (10.19)$$

for some $\theta(\bar{\zeta})\geq 0$ and $K_{0}>0$, which are parameterized by functions

$$\alpha(\cdot):\left(\mathbb{R}^{2}\right)^{N+M}\rightarrow\mathbb{R}$$

The following theorem establishes that the assumption of continuously differentiable ϕ in (10.1) (which ensures $\dot{F}$ is of bounded variation) is adequate for the control design to achieve asymptotic tracking of clusters. We first state the following lemma, which is used in the proof of the theorem; a proof for this lemma can be found in [29].

Lemma 10.1. *For a nonnegative function* $g:\mathbb{R}\rightarrow\mathbb{R}$ *of bounded variation, if* $\int_{0}^{\infty}g(\tau)d\tau<\infty$*, then* $\lim_{t\rightarrow\infty}g(t)=0$.

Theorem 10.3 (Tracking). *For the site-resource dynamics given by (10.18), if*

$$u=\bar{u}+\check{P}_{x|y}^{T}\phi$$

where $\bar{u}\in\bar{\mathcal{U}}(\bar{x}^{T}\bar{\phi})$*, then* $\dot{F}\leq 0$ *for all* $t\geq 0$ *and the resource locations asymptotically track the cluster centers, that is,* $\lim_{t\rightarrow\infty}\bar{y}(t)=0$.

Proof. *For* $\bar{u}\in\bar{\mathcal{U}}(\bar{x}^{T}\bar{\phi})$*, it is straightforward to show that* $\dot{F}$ *given by (10.18) reduces to* $\dot{F}=-K_{0}\bar{y}^{T}\check{P}_{y}\bar{y}-\theta(\bar{\zeta})$ *for some* $\theta(\bar{\zeta})\geq 0$ *and* $K_{0}>0$*. Therefore* $\dot{F}\leq 0$*. Since* $F(t)$ *(which denotes the time-dependence of free energy* $F(\zeta(t))$*) is bounded below (from Theorem 10.2(1)) and* $\dot{F}\leq 0$*, this implies* $F(t)\rightarrow F_{\infty}$ *for some* $|F_{\infty}|<\infty$ *as* $t\rightarrow\infty$*. Thus,*

$$\int_{0}^{\infty}|\dot{F}(\tau)|d\tau=-\int_{0}^{\infty}\dot{F}(\tau)d\tau=F(0)-F_{\infty}<\infty$$

Therefore, since $\dot{F}$ *is of bounded variation, we can deduce that* $\lim_{t\rightarrow\infty}|\dot{F}(t)|=0$ *from Lemma 10.1. Since*

$$\dot{F}=-K_{0}\bar{y}^{T}\check{P}_{y}\bar{y}-\theta(\bar{\zeta})\Rightarrow K_{0}\bar{y}^{T}\check{P}_{y}\bar{y}\leq|\dot{F}| \text{ and } \check{P}_{y} \text{ is positive definite}$$

with elements bounded away from zero (since $\min_{j}\{p(y_{j})\}\geq\nu>0$*), we have* $\bar{y}\rightarrow 0$ *as* $t\rightarrow\infty$*.* □

Clearly, resource velocities of the form $u = \bar{u} + \check{P}^T_{x|y}\phi$ can take very large values near $\bar{y} \neq 0$. The following theorem emphasizes that this control design is not conservative, that is, if there exists a control design that achieves $\dot{F} \leq 0$ without growing unbounded near $\bar{y} = 0$, then there necessarily exists a bounded element from $\bar{\mathcal{U}}(\bar{x}^T\bar{\phi}) + \check{P}^T_{x|y}\phi$ that guarantees $\dot{F} \leq 0$.

Theorem 10.4 (Lipschitz property of control). *If there exists a $\hat{u} : \left(\mathbb{R}^2\right)^{N+M} \to \left(\mathbb{R}^2\right)^M$ such that $\hat{u}$ is Lipschitz at $\bar{\zeta} = 0$ and*

$$\dot{F} = \bar{x}^T\bar{\phi} + \bar{y}^T\check{P}_y(\hat{u} - \check{P}^T_{x|y}\phi) \leq 0$$

then $u = \bar{u}_S + \check{P}^T_{x|y}\phi$ is also Lipschitz at $\bar{\zeta} = 0$, where

$$\bar{u}_S(\bar{\zeta}) = -[K_0 + \frac{\bar{x}^T\bar{\phi} + \sqrt{|\bar{x}^T\bar{\phi}|^2 + (\bar{y}^T\check{P}_y\bar{y})^2}}{\bar{y}^T\check{P}_y\bar{y}}]\bar{y} \in \bar{\mathcal{U}}(\bar{x}^T\bar{\phi}). \tag{10.20}$$

i.e., there exists a $\delta > 0$ and a constant c_0 such that $||\bar{u}_s(\bar{\zeta})|| \leq c_0||\bar{\zeta}||$ for $||\bar{\zeta}|| \leq \delta$.

Theorem 10.4 can be proven in a manner analogous to the proof for Proposition 3.43 in [24]); see [29] for details. Theorems 10.3 and 10.4 are general in the sense that they do not impose conditions other than smoothness on ϕ. We can obtain better bounds on the control effort needed if we assume additional conditions on the cluster dynamics. We use the following lemma, the proof of which can be found in [29].

Lemma 10.2. *Let $x \in \left(\mathbb{R}^2\right)^N$ and y in $\left(\mathbb{R}^2\right)^M$ satisfy*

$$\frac{||y_{c[i]} - x_i||^2}{||y_j - x_i||^2} \leq \delta < 1,\ \forall 1 \leq j \leq M,\ j \neq c[i]$$

where $y_{c[i]}$ is the unique closest resource location to x_i, that is,

$$||y_{c[i]} - x_i|| = \min_{1 \leq j \leq M} ||y_k - x_i|| \textit{ for } 1 \leq i \leq N.$$

Then, there exists a constant $\bar{k}_2 < \infty$ such that

$$||\check{P}_x\check{P}_{y|x}\check{P}^T_{x|y} - \check{P}_x\check{P}^c_{y|x}\check{P}^{cT}_{x|y}|| < 3k_2||\bar{y}||(1 + k_2||\bar{y}||^2 + k_2^2||\bar{y}||^2)$$

Theorem 10.5 (Bounded control). *Assume the cluster dynamics are size-consistent, that is*

$$\bar{\phi}^T\bar{x}^c \leq 0 \textit{ and } \frac{||y_{c[i]} - x_i||^2}{||y_j - x_i||^2} \leq \delta < 1\ \ \forall j,\ \ 1 \leq j \leq M,\ \ j \neq c[i],$$

where $y_{c[i]}$ is the unique closest resource location to x_i. Consider the control law

$$u(\bar{\zeta}) = \check{P}_{x|y}^{T}\phi + [K_0 + \frac{\alpha + \sqrt{\alpha^2 + (\bar{y}^T \check{P}_y \bar{y})^2}}{\bar{y}^T \check{P}_y \bar{y}}]\bar{y},$$

where $\alpha = \bar{\phi}^T \bar{x} - \bar{\phi}^T \bar{x}^c$ *and* $K_0 > 0$*. Then, for some quadratic function* $c_1(\cdot)$*,*

$$\lim_{t\to\infty} \bar{y} = 0 \textit{ and } \|u(\bar{\zeta})\| \leq (2\nu^{-1}c_1(\|\bar{y}\|)\|\|x\| + 1)\|\phi\| + (K_0+1)\|\bar{y}\|$$

Therefore, if $\|\phi\|, \|x\| \leq c_2 < \infty$*, then* $\|u(\bar{\zeta}(t))\|$ *is bounded.*

Proof. *Note that* $\dot{F} = \bar{\phi}^T \bar{x} + \bar{y}^T P_y(u - \check{P}_{x|y}^T \phi) = \bar{\phi}^T \bar{x}^c + \alpha + \bar{y}^T P_y(u - \check{P}_{x|y}^T \phi)$*. After substituting for u, we obtain*

$$\dot{F} = \bar{\phi}^T \bar{x}^c - K_0 \bar{y}^T \check{P}_y \bar{y} - \sqrt{\alpha^2 + (\bar{y}^T \check{P}_y \bar{y})^2} \leq 0$$

By following the same arguments as in the proof for Theorem 10.3, we can show that $\lim_{t\to\infty} \bar{y} = 0$*. From Lemma 10.2 we have*

$$\|\check{P}_x \check{P}_{y|x} \check{P}_{x|y}^T - \check{P}_x \check{P}_{y|x}^c \check{P}_{x|y}^{cT}\| < \|\bar{y}\| c_1(\|\bar{y}\|)$$

for some quadratic function $c_1(\cdot)$*. Since* $\alpha = x^T(\check{P}_x \check{P}_{y|x} \check{P}_{x|y}^T - \check{P}_x \check{P}_{y|x}^c \check{P}_{x|y}^{cT})\phi$*, we have*

$$|\alpha\|\bar{y}\|/\bar{y}^T \check{P}_y \bar{y}| \leq \nu^{-1} c_1(\|\bar{y}\|)\|x\|\|\phi\|$$

Further

$$\|u(\bar{\zeta})\| \leq \|\phi\| + (2|\alpha|\|\bar{y}\|/\bar{y}^T \check{P}_y \bar{y}) + (1 + K_0\|\bar{y}\|)$$

implies $\|u(\bar{\zeta})\|$ *is bounded by*

$$\|\phi\| + 2\nu^{-1}c_1(\|\bar{y}\|)\|x\|\|\phi\| + (K_0+1)\|\bar{y}\| \leq (2\nu^{-1}c_1(\|\bar{y}\|)\|c_2 + 1)c_2 + (K_0+1)\|\bar{y}\|$$

where $\bar{y} \to 0$ *from above.* □

□

This design procedure provides great flexibility, with the free energy F viewed as a control Lyapunov function and u designed to make $\dot{F} \leq 0$. Alternative computationally efficient control laws can be devised by exploiting the properties of Γ. For instance, we can guarantee exponential convergence of $\bar{y} \to 0$ by appropriately choosing the function $\theta(\bar{\zeta})$ in the control design [29].

Note that the size-consistency assumption on cluster dynamics is required only in the interim period when resource locations are far from the cluster centers (i.e., when $\bar{y} \neq 0$). Once the resource locations track the cluster centers, these assumptions are not required, and instead monitoring cluster splits is required.

10.3.3 Cluster Evaluation

As the resource locations begin to track the cluster centers, cluster evaluation is completed based on the following decision choices.

Splitting: The decision to split can be enforced when the parameter β satisfies the condition

$$\beta^{-1} = 2\lambda_{\max}(C_{x(t_c)|y_j(t_c)})$$

If time is fixed at $t = t_c$, this splitting condition implies that y is at a local maxima (or inflection point) of F (i.e., the Hessian of F is no longer positive definite).

The set of resource locations after splitting is determined by the same procedure used as in the static case described in [28]. The new locations are denoted by

$$\mathbf{y}_{new} = q_{\text{split}}(\mathbf{y})$$

Splitting does not pose problems for the dynamic implementation of the algorithm as the computation time to determine $q_{\text{split}}(\mathbf{y})$ is negligible compared to computation of $u(t)$.

Tracking: At time t if $y_j, 1 \leq j \leq M$, are at the cluster centers and the parameter β does not satisfy the splitting condition, then one can continue to track the cluster centers by assigning $y_j(t) = \sum_i p(x_i(t)|y_j(t))x_i(t)$. Alternatively, one can choose to improve coverage resolution by increasing cooling rates, eventually leading to the splitting condition being satisfied, resulting in finer clusters and therefore higher resolution.

10.4 Scalability of the DME Algorithm

The framework presented in this chapter aims at avoiding local minima. As a consequence the computations are global, or centralized, in the sense that the computation of *each* resource location uses the values of *all* the site locations. However, the contribution of distant site locations becomes progressively smaller as the parameter β is increased. In fact, the partitions are hard as $\beta \to \infty$ and consequently the contribution of site locations that are not nearest neighbors goes to zero. In this sense, the computation of resource locations changes from being *truly global* to *truly local*, or distributed, as β is increased from zero to infinity.

In the case of static sites, we have exploited this tendency toward distributedness to develop a scalable algorithm that provides a close approximation to the original algorithm. The basic idea is straightforward: quantify and exploit inherent cluster separations based on cluster interaction information, then apply the MEP-based algorithm separately to each cluster. This approach leads to a hierarchical application of this algorithm. More specifically, at each phase transition, we compute the *level of interaction* between clusters by

$$\varepsilon_{ji} = \sum_{x \in R_i} p(y_j|x)p(x), \tag{10.21}$$

which represents the level of interaction between sites in cluster R_i with cluster R_j. We then determine subsets of strongly interacting clusters based on these ε-values and apply the DA to each subset individually, ignoring the other clusters. Obviously the trade-off in this is that the resulting distortions will be larger than in the original algorithm. We have shown we can bound the difference in the resulting cluster centers found by the scalable algorithm as follows:

$$|y_i - \hat{y}_i| < \varepsilon_i \left(\frac{1 - \sum_{x \in R_i} p(x)}{1 - \varepsilon_i \sum_{x \in R_i} p(x)} \right) \quad \text{with } \varepsilon_i = \sum_{j \neq i} \varepsilon_{ij}, \tag{10.22}$$

where y_i represent the cluster center for cluster R_i determined from the original DA algorithm, and $\hat{y}_i$ represent the cluster center for cluster R_i determined from the scalable algorithm.

In simulations on data sets containing from 5000 to $15,000$ static sites, the scalable algorithm has converged to a solution approximately six to seven times faster on average than the DA algorithm, with an increase in the final distortion value obtained of only about 5%. On data sets containing from $15,000$ up to $50,000$ static sites, the scalable algorithm converges to a solution where the original DA algorithm fails; see [28] for further details.

To extend this approach to the dynamic setting, we first note that clusters can interact even when β values are high, due to site dynamics. Thus we propose to monitor values of the norm of the Hessian $\partial^2 F / \partial y^2$ and estimate the *effective* radius around each cluster center y_j, beyond which the site locations can be ignored.

10.4.1 Incorporating Scalability into the Cost Function

An additional approach for improving scalability in our DME algorithm is to consider incorporating computational cost into the algorithm's objective function. More specifically, we consider computing the association weights $p(x_i,y_j)$ and resource locations y_j such that the entropy term

$$H(x,y) = -\sum_j \sum_i p(x_i,y_j) \log(p(x_i,y_j))$$

is maximized subject to constraining the distortion, or coverage cost, as before by D_0, with

$$D_0 = \sum_j \sum_i p(x_i)p(y_j|x_i)d(x_i,y_j)$$

and also constraining the *cluster interaction* and the *computational cost* by values C_I and C_C, respectively, where

$$C_I = \sum_j \varepsilon_{ji} d(x_i, y_j) M_{ij}, \text{ with } M_{ij} = \begin{cases} 0 & x_i \in R_j \\ 1 & x_i \notin R_j \end{cases} \tag{10.23}$$

and

$$C_C = \sum_j \varepsilon_{ij} N_j^2, \text{ with } N_j = \sum_i (1 - M_{ij}). \tag{10.24}$$

That is, we now solve the modified Lagrangian

$$\max_{y_j, p(y_j|x_i)} H - \beta_1 D_0 - \beta_2 C_I - \beta_3 C_C \tag{10.25}$$

To further improve tractability of this algorithm, we consider a relaxation of the problem where M_{ij} and N_j are approximated by the differentiable functions

$$M_{ij} = 1 - e^{-\frac{d(x_i, y_j)}{\sigma_j}} \text{ and } N_j = \sum_i e^{-\frac{d(x_i, y_j)}{\sigma_j}}. \tag{10.26}$$

Implementation and simulations using these approaches are ongoing.

10.5 Simulations

In this section, we present simulation results for a variety of data sets with different underlying dynamics. These test cases are specifically designed to highlight key features of our DME framework and to allow for performance analysis and discussion.

10.5.1 The Basic Algorithm

We summarize in the following outline the main steps and flow of the the algorithm. This implementation applies to the basic version of the algorithm, having no external user-based directives:

- **Step 0:** Initialize the algorithm: $t = t_0$, $\beta = 0$, and $u = 0$
- **Step 1:** Determine the resource locations (10.9) together with the association probabilities (10.7)
- **Step 2:** Simulate motion of sites x_i under (10.1) and determine resource velocities from (10.1) for time interval Δt
- **Step 3a:** If $\bar{y} = 0$, then
  - Evaluate phase transition condition:
    - Satisfied:
      - If more resolution required, split y_j, redistribute weights, determine new y_j values and return to **Step 2**; otherwise check stopping-time criteria and exit if met, else return to **Step 2**

- Not satisfied:
  - If more resolution required, increase β and return to **Step 2**; otherwise set y_j to respective centroids and return to **Step 2**

- **Step 3b:** If $\bar{y} \neq 0$ then compute control u using (10.20) and return to **Step 2**

All simulations were carried out in MATLAB. For these simulations the dynamics in (10.1) were completed by discretizing using a fourth-order Runge–Kutta method (RK-4) [2]. In the RK-4 method, the error per step is $O((\Delta t)^5)$, while the total accumulated error is $O((\Delta t)^4)$. The time steps were chosen so that the solution converges. Note that the time required to compute $q_{\text{split}}(y)$ is comparable to that required to compute $u(t)$.

10.5.2 Natural Cluster Identification and Tracking

In this primary simulation, we use a data set with 160 mobile sites. A velocity field $\phi(t,x)$ is assigned to each of the mobile sites such that natural clusters emerge within 8 seconds.

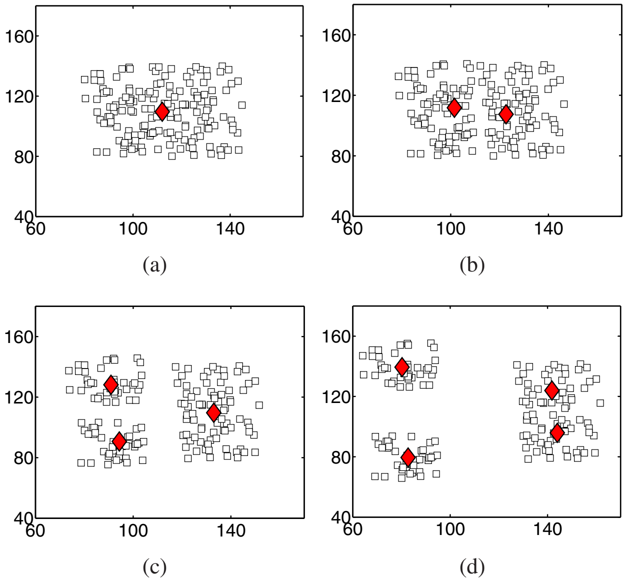


Fig. 10.3 Simulation results showing cluster identification and tracking. Snapshots (a), (b), (c) and (d) show the locations of mobile sites $x_i, 1 \leq i \leq 160$ (shown by squares) and resources $y_j, 1 \leq j \leq M$ (shown by diamonds) at various phase transition instances. All sites are initially concentrated at the center of the domain area and then slowly drift apart. Four natural clusters emerge at the end of the time horizon, shown in (d).

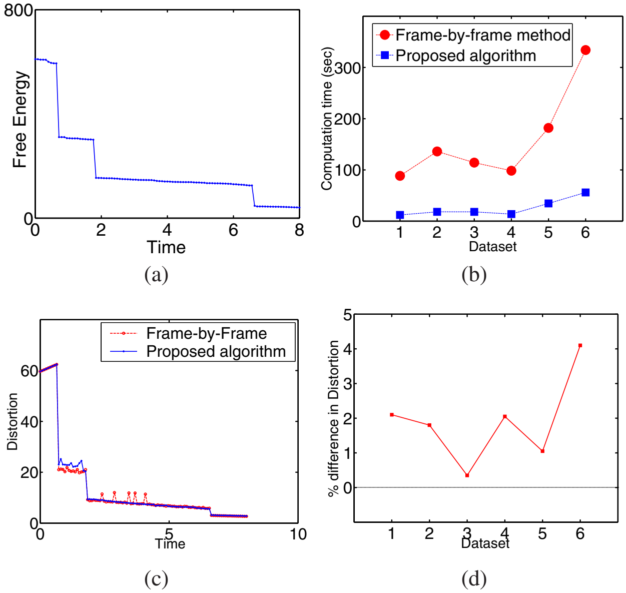


Fig. 10.4 (a) Free energy F with respect to time. The control value u ensures that $dF/dt \leq 0$. Sharp decreases in F are due to phase transitions. Progressively decreasing free energy results in improved coverage and tracking throughout the time horizon. (b) Comparison of the computation times for the frame-by-frame approach and the DME algorithm. (c) Comparison of the distortion achieved by the frame-by-frame method and the proposed framework. (d) Percentage error in final distortion achieved by the proposed DME algorithm versus the frame-by-frame method.

The algorithm is initiated with one resource placed at the centroid of the sites (at $t = 0$) (a diamond denotes this location and squares represent the sites, Fig. 10.3(a)). As the site dynamics evolve, the site locations move according to the equation $\dot{x} = \phi(x,t)$. The DME algorithm progressively updates the association probabilities and resource locations, and determines control values (i.e., the resource velocities from (10.20)) in order to track cluster centers. Figures 10.3(b), (c) and (d) show the locations of the sites and the resources in the interim instants. The number of resource locations increases progressively due to successive phase transitions, and as seen in the figure, the resource locations identify and track natural clusters in the underlying data. At the end of the time horizon, all natural clusters are identified. The algorithm successfully avoids several local minima and provides progressively better coverage. Figure 10.4(a) shows a plot of the coverage function F with respect to time. Note that at each phase transition, there is a sharp decline in F. This decline is due to the fact that at every phase transition, one or more resource locations are added, which results in lower free energy, thereby providing better coverage.

Depending on the distribution of the underlying data, the DME algorithm is five to seven times faster than the frame-by-frame method. A comparison of the instantaneous distortion value $\sum_i p_i \min_j d(x_i, y_j)$ obtained by the two algorithms is presented in Fig. 10.4(c). The proposed algorithm identifies and tracks the clusters hierarchically such that the distortion steadily decreases, with sharp decreases at splits. As seen in the figure, the frame-by-frame method for the same number of resources achieves slightly lower distortion, but with frequent spikes due to the spatio-temporal effects. Moreover, the frame-by-frame method uses five times the computation time required for the proposed algorithm. Figure 10.4(d) shows a comparison of the final distortion achieved by the proposed algorithm and the frame-by-frame method (for the same time step and same number of resources). As is seen in the figure, the proposed algorithm achieves a distortion similar to the frame-by-frame approach (within 0.5% to 4.3%), however it uses considerably less computation time as shown in (b).

In the absence of spatio-temporal smoothening, the clustering solutions obtained at two successive time instants might be considerably disparate, even though the number of natural clusters in the data set remains the same. Figure 10.5 shows the clustering solution obtained by the frame-by-frame approach at two successive time instants. Three clusters are identified by the algorithm at each instant, but at considerably different spatial locations. This occurs because no information from the previous clustering solution is used for determining the solution at the next time instant. On the other hand, the proposed framework overcomes this problem by using a smooth control value everywhere except during cluster splits via phase transition. In order to speed up the frame-by-frame approach, we increase the time step between successive frames. During such an implementation, the resource locations obtained at the previous time instant are used in the interim between successive frames. This results in a clustering solution that deteriorates in the interim because of the lack of new information. Figures 10.5(c) and (d) show the distortion obtained by a three fold and six fold increase in the time steps. As is seen in the figures, the distortion obtained from the frame-by-frame approach deteriorates considerably with respect to the proposed algorithm. Note that even for a six fold increase in the time step, the computation time for the frame-by-frame is slightly higher than that of the proposed algorithm. In most of the applications, such a phenomenon would not be desirable.

Once natural clusters are identified and tracked, the user may desire higher resolution clustering and/or simultaneous tracking. User-based directives are easily incorporated in our DME framework, providing flexibility in the modes of operation. Higher resolution clusters can be achieved by increasing the annealing parameter value until the phase transition condition is satisfied and the eventual splitting of resources is obtained.

10.6 Ongoing Work and Conclusions

In this section we discuss ongoing work and further extensions of the framework presented herein.

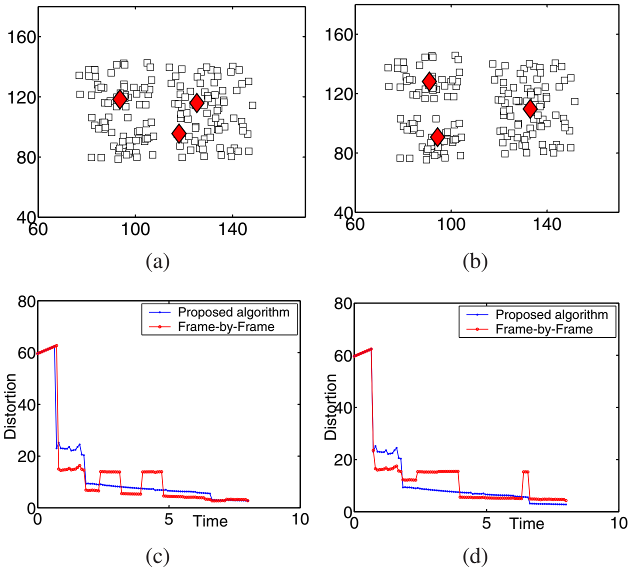


Fig. 10.5 (a) and (b): Clustering results from two successive frames using the frame-by-frame approach. In both instances, three resource locations were identified by the algorithm, but at considerably different positions. Such a solution violates the spatio-temporal requirement of the dynamic clustering algorithm. This happens because none of the clustering information from previous frame is employed in order to determine the solution at future frames. (c) and (d): Comparison of distortion obtained by the proposed algorithm and the frame-by-frame approach under different time steps. (c) Proposed algorithm : $\Delta t = 0.08$ s, frame-by-frame approach: $\Delta t = 0.24$ s. (d) Proposed algorithm : $\Delta t = 0.08$ s, frame-by-frame approach: $\Delta t = 0.48$ s.

10.6.1 General Extensions of the Coverage Problem

Constraints on resources: The resources in the framework presented are assumed to be identical. However, depending on the underlying problem domain, these resources may be heterogeneous, where different constraints apply to different resources; for example, vehicles may be of different sizes with different coverage capacities. Resources can be made nonidentical in our framework by introducing weights λ_j to each resource location y_j. This interpretation yields a modified free energy function,

$$F = -\frac{1}{\beta}\sum_i p_i \log \sum_j \lambda_j e^{-\beta d(x_i, y_j)}. \tag{10.27}$$

Constraints on resources are implemented by specifying constraints on λ_j. For example, constraining the λ_j by constants W_j yields resource locations that have weights in the same ratios as $\{W_j\}$. More details on this formulation for static problems in the context of facility location, drug discovery and vector quantization are presented, respectively, in [23, 28, 21]. The same formulation easily extends to the dynamic case, where static constraints can be incorporated by redefining the free energy in (10.27), and minimizing the appropriate Lagrangian $L = F + \sum_j \mu_j(\lambda_j - W_j)$ with respect to $\{y_j\}$ and $\{\lambda_j\}$.

Inertial forces in vehicle dynamics: In this chapter, we have focused on tracking of cluster centers when velocity fields (one state per direction for each vehicle) are given for sites. Procedures that are applicable when higher order (i.e., multiple-state) differential equations are given are presented in [34]. For example, in the context of vehicle systems, the dynamics of autonomous mobile agents are often controlled by thrust actuators, the control term being the amount of thrust in each direction. The corresponding model for a domain with N mobile agents becomes $x_i = [\xi_i \;\; \eta_i]^T \in \mathbb{R}^2$ and M resource locations $y_j = [\rho_j \;\; \omega_j]^T \in \mathbb{R}^2$ as before, whose dynamics are now given by

$$\ddot{x}(t) = \boldsymbol{\Upsilon}(x(t), \dot{x}(t), y(t), \dot{y}(t), t), \;\; x(0) = x_0, \;\; \dot{x}(0) = \dot{x}_0, \tag{10.28}$$

$$\ddot{y}(t) = u(t), \;\; y(0) = y_0, \; \dot{y}(0) = \dot{y}_0. \tag{10.29}$$

Our task is to determine the accelerations ($\ddot{y}$) for the M mobile resources such that they track cluster centers. However by writing the above second order differential equations for each vehicle into first order vector differential equations, these equations take the same form as (10.1), albeit with additional algebraic structure. Under this scenario, the Euclidean distance metric is of the form

$$d(x_i, y_j) = (x_i - y_j)^2 + \theta(\dot{x}_i - \dot{y}_j)^2,$$

where θ is a constant. Choosing high values of θ thus gives relatively more importance to velocities and hence yields cluster centers for instantaneous headings (see [34]).

10.6.2 Estimating Data Dynamics

In the framework presented, the velocity fields ϕ of sites are assumed to be known. In the case of noisy dynamics, where ϕ is given by perturbations $n(t)$ about a nominal function $\bar{\phi}$, that is $\phi = \bar{\phi}(x, y, t) + n(t)$, and where the measurements of site and resource locations are noisy, control designs based on estimated values of (x, y) have provided satisfactory performance. This is primarily due to the fact that the shapes of the Gibbs distribution functions are insensitive to slight perturbations in x and y. When site dynamics are completely realized by velocity fields, if these fields are not known a priori, we can estimate them using system identification methods, and design $u(t)$ based on the estimated fields. The inherent robustness in the algorithm due to the properties of the Gibbs distribution again yield satisfactory performance.

10.6.3 Robustness to Modeling Uncertainties

The effect of noisy channel transmissions can also be accommodated in our framework, by modifying the instantaneous distortion term so that

$$\tilde{D} = \sum_i \sum_j p_i p(y_j|x_i) d'(x_i, y_j)$$

where $d'(x_i, y_j)$ reflects the probability of the event that x_i was actually sent given that the received message is interpreted as x_k.

Our DME framework can address robustness issues to a variety of transmission uncertainties. We can accommodate resource-dependent site-location uncertainties, for example, where the probability of receiving the location x_k when the location x_i is sent now depends on the receiver at the resource y_j. Additionally, transmission errors in resource locations can also be addressed; this case is analogous to the noisy vector quantization problem addressed in [21].

We can also address communication link failures; to do so, we introduce a binary random variable $\chi_{ij} \in \{0,1\}$, with a given probability distribution $p_{\chi_{ij}}$, where $\chi_{ij} = 1$ (or 0) implies that the ith site is (or is not) in communication with the jth resource. Again, we modify the instantaneous distortion term such that

$$\hat{D} = \sum_i \sum_j p_i p(y_j|x_i) p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j) \tag{10.30}$$

Appending the entropy to include the link failure probability distribution gives $\hat{H} = -\sum_i \sum_j p_i p(y_j|x_i) p_{\chi_{ij}}(\chi_{ij}) \log p(y_j|x_i)$, and the association probabilities then take the form

$$p(y_j|x_i) = \frac{\exp(-\beta(\sum_{\chi_{ij}} p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j)))}{Z_i} \tag{10.31}$$

$$\text{where } Z_i = \sum_j \exp(-\beta(\sum_{\chi_{ij}} p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j))) \tag{10.32}$$

The instantaneous free energy is then rewritten as

$$\hat{F} = -\frac{1}{\beta} \sum_i p_i \log \sum_j [-\beta(\sum_{\chi_{ij}} p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j))] \tag{10.33}$$

This term can now be used as a metric for instantaneous coverage under link failures. An analysis similar to that of the proposed algorithm in Sec. 10.2 can be employed to obtain dynamic clustering.

10.6.4 Conclusions

In this chapter, we proposed the dynamic maximum entropy (DME) framework for formulating and solving the dynamic coverage problem. As shown in the simulations, the proposed framework resolves both the coverage as well as the tracking aspects of the dynamic coverage problem. Using a control-theoretic approach to determine the velocity field for the cluster centers, we achieve progressively better coverage with time, which is shown to be five to seven times faster than the frame-by-frame method. The hierarchical aspect of the proposed algorithm enables us to identify natural clusters in the underlying data and characterize the notion of cluster resolution. Notions of coverage and clusters based on the MEP naturally allow for quantification of inter cluster and intra cluster dynamics, and greatly facilitate the simultaneous design process for the coverage and tracking objectives of the underlying problems.

Acknowledgements This work is partially supported by NSF label grants ECS-0725708 and ECS-0449310 CAREER.

References

1. Blankertz, B., Dornhege, G., Krauledat, M., Müller, K., Curio, G.: The non-invasive berlin brain–computer interface: Fast acquisition of effective performance in untrained subjects. Neuroimage **37**(2), 539–550 (2007)
2. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. John Wiley, New York (2004)
3. Chudova, D., Gaffney, S., Mjolsness, E., Smyth, P.: Translation-invariant mixture models for curve clustering. In: Proc. 9th ACM SIGKDD, pp. 79–88 (2003)
4. Cortés, J., Martínez, S., Karatas, T., Bullo, F.: Coverage Control for Mobile Sensing Networks. IEEE Trans. Robotics and Automation **20**(2), 243–255 (2004)
5. Drezner, Z.: Facility Location: A Survey of Applications and Methods. Springer Verlag, New York (1995)
6. Du, Q., Faber, V., Gunzburger, M.: Centroidal Voronoi Tessellations: Applications and Algorithms. SIAM Review **41**(4), 637–676 (1999)
7. Elia, N., Mitter, S.: Stabilization of Linear Systems with Limited Information. IEEE Trans. Automatic Control **46**(9), 1384–1400 (2001)
8. Frazzoli, E., Bullo, F.: Decentralized algorithms for vehicle routing in a stochastic time-varying environment. Proc. IEEE Conf. Decision and Control (CDC2004) pp. 3357–3363 (2004)
9. Geman, S., Geman, D.: Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. IEEE Trans. Pattern Analysis and Machine Intelligence **6**, 721–741 (1984)
10. Gersho, A., Gray, R.: Vector Quantization and Signal Compression, 1st edn. Kluwer, Boston, MA (1991)
11. Gray, R., Karnin, E.D.: Multiple local minima in vector quantizers. IEEE Trans. Information Theory **IT-28**, 256–361 (1982)
12. Jaynes, E.T.: Information theory and statistical mechanics. Physical Review **106**(4), 620–630 (1957)
13. Jaynes, E.T.: Probability Theory—The Logic of Science. Cambridge University Press (2003)
14. Landau, L.D., Lifshitz, E.M.: Statistical Physics, Part 1, vol. 3, 3rd edn. Oxford (1980)

15. Lloyd, S.: Least-squares quantization in PCM. IEEE Trans. Information Theory **28**(2), 129–137 (1982)
16. Mitter, S.K.: Control with limited information. Plenary lecture at International Symposium on Information Theory, Sorrento, Italy (2000)
17. Nunez, P.L., Srinivasan, R.: Electric Fields of the Brain: The Neurophysics of EEG. Oxford University Press, USA (2006)
18. Omar, C., Vaidya, M., Bretl, T.W., Coleman, T.P.: Using feedback information theory for closed-loop neural control in brain-machine interfaces. In: 17th Ann. Computational Neuroscience Meeting (CNS). Portland, OR (2008). Invited Session on Methods of Information Theory in Computational Neuroscience
19. Rose, K.: Deterministic annealing, clustering, and optimization. Ph.D. thesis, California Institute of Technology, Pasadena (1991)
20. Rose, K.: Constrained Clustering as an Optimization Method. IEEE Trans. Pattern Anal. Machine Intell. **15**, 785–794 (1993)
21. Rose, K.: Deterministic annealing for clustering, compression, classification, regression and related optimization problems. Proc. IEEE **86**(11), 2210–39 (1998)
22. Salapaka, S.: Combinatorial optimization approach to coarse control quantization. In: Proc. 45th IEEE Conf. Decision and Control (CDC2005), pp. 5234–5239. San Diego, CA (2006)
23. Salapaka, S., Khalak, A.: Locational optimization problems with constraints on resources. In: Proc. 41st Allerton Conference, pp. 1240–1249 (2003)
24. Sepulchre, R., Jankovic, M., Kokotovic, P.: Constructive Nonlinear Control. Springer-Verlag (1997). URL http://www.montefiore.ulg.ac.be/services/stochastic/pubs/1997/SJK97a
25. Shannon, C.E., Weaver, W.: The mathematical theory of communication. Univ. of Illinois Press, Urbana IL (1949)
26. Sharma, P., Salapaka, S., Beck, C.: A maximum entropy based scalable algorithm for resource allocation problems. Proc. American Control Conference (ACC2007) pp. 516–521 (2007)
27. Sharma, P., Salapaka, S., Beck, C.: Entropy based algorithm for combinatorial optimization problems with mobile sites and resources. In: Proc. American Control Conference (ACC2008), pp. 1255–1260 (2008)
28. Sharma, P., Salapaka, S., Beck, C.: A scalable approach to combinatorial library design for drug discovery. Journal of Chemical Information and Modeling **48**(1), 27–41 (2008)
29. Sharma, P., Salapaka, S., Beck, C.: Entropy-based framework for dynamic coverage and clustering problems. IEEE Trans. Automatic Control (Accepted; to Appear 2011)
30. Sheu, J.B.: A fuzzy clustering-based approach to automatic freeway incident detection and characterization. Fuzzy Sets Syst. **128**(3), 377–388 (2002)
31. Sontag, E.D.: A Lyapunov-like characterization of asymptotic controllability. SIAM J. Control Optim. **21**, 462–471 (1983)
32. Sontag, E.D.: A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization. System and Control Letters **13**(2), 117–123 (1989)
33. Therrien, C.W.: Decision, Estimation and Classification: An Introduction to Pattern Recognition and Related Topics, vol. 14, 1st edn. Wiley, New York (1989)
34. Xu, Y., Salapaka, S., Beck, C.: Dynamic maximum entropy algorithms for clustering and coverage control. In: Proc. IEEE Conf. Decision and Control (CDC2010), pp. 1836–1841. Atlanta, GA (2010)