Entropy-Based Framework for Dynamic Coverage and Clustering Problems

Puneet Sharma, Srinivasa M. Salapaka, *Member, IEEE*, and Carolyn L. Beck, *Senior Member, IEEE*

***Abstract*—We propose a computationally efficient framework to solve a large class of dynamic coverage and clustering problems, ranging from those that arise from deployment of mobile sensor networks to classification of cellular data for diagnosing cancer stages. This framework provides the ability to identify natural clusters in the underlying data set. In particular, we define the problem of minimizing instantaneous coverage as a combinatorial optimization problem in a Maximum Entropy Principle (MEP) framework that we formulate specifically for the dynamic setting, and which allows us to address inherent tradeoffs such as those between the *resolution* of the identified clusters and computational cost. The proposed MEP framework addresses both the coverage and the tracking aspects of these problems. Locating cluster centers of swarms of moving objects and tracking them is cast as a control design problem ensuring that the algorithm achieves progressively better coverage with time. Simulation results are presented that highlight the features of this framework; these results demonstrate that the proposed algorithm attains target coverage costs five to seven times faster than related frame-by-frame methods.**

***Index Terms*—Maximum entropy principle (MEP).**

I. Introduction

With recent advances in geographic information systems, geopositioning and wireless sensor networks, there is a growing interest in developing algorithms for deployment of mobile resources that continuously *cover* a set of mobile sites in a region. In particular, problems related to determining clusters in an ensemble of moving objects have received considerable attention lately. Such problems have numerous applications, such as in developing automatic deployment and tracking algorithms for surveillance and military applications [1], [2], in clustering of the spatio-temporal dynamics of brain signals [3]–[5], in routing traffic by clustering traffic flow [6], in finding dynamic clusters in groups of animals in studies of their migration patterns [7], and in weather forecasting by clustering cyclone patterns [8].

The emphasis on finding clusters in *dynamically evolving data* is fairly recent and many issues pertaining to quantification, analysis, and design for coverage remain unsolved. However, clustering problems on *static data* have been widely studied in various areas, such as the minimum distortion problem in data compression [9], facility location assignments [10], optimal quadrature rules and discretization of partial differential equations [11], pattern recognition [12], drug discovery [13], neural networks [14], and clustering analysis [15]. Recently they have appeared in the control literature, especially in coarse quantization [16]–[18].

Although the static problems focus on different and seemingly unrelated goals, in fact, they have 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 (for instance, a decomposition of the chemical property space in the development of a library of compounds for drug discovery, or for determining subregions of interest for coverage control), as well as an optimal assignment of values, or elements, from a finite *resource* set to each cell in the partition (for instance, subsets of drug properties [13], or vehicle locations [19]). Since both the number of partitions as well as the number of element associations to partitions are combinatorial, these problems are computationally complex (i.e., NP-hard [20]), which rules out exhaustive search methods. For instance, in the drug discovery problem, determining 30 representative compounds from an array of 1000 compounds results in approximately 2.43×10^{57} possibilities. The cost functions in these optimization problems are riddled with local minima, and therefore it is crucial to design algorithms that do not get trapped in local minima [20]. Since even the static resource allocation problem is NP-hard, we cannot expect to find optimal solutions to the dynamic problem. As a result, the proposed algorithms are either based on heuristics, or solve approximations of the original problem.

The complexity of the combinatorial problem addressed in this paper is further compounded by the inherent dynamics associated with each data point, i.e., the moving objects. These objects could be mobile threat locations in a battlefield scenario, forest fires, or unmanned vehicles, depending on the context of the problem. The task at hand is to design a velocity field 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 fields* in the dynamic case, and 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.

Manuscript received September 04, 2008; revised September 10, 2009; accepted April 04, 2011. Date of publication October 03, 2011; date of current version December 29, 2011. This paper was presented in part at the American Control Conference, Seattle, WA June 2008. This work was supported in part by NSF ITR Collaborative Research grant ECS-0426831 and NSF ECS 0449310 CAR grant. Recommended by Associate Editor I. Paschalidis.

P. Sharma is with Siemens Corporate Research, Princeton, NJ 08540 USA (e-mail: sharma.puneet@gmail.com).

S. M. Salapaka is with the Department of Mechanical Science and Engineering, Coordinated Science Lab, University of Illinois at Urbana Champaign, Urbana, IL 61801 USA. (e-mail: salapaka,beck3@illinois.edu).

C. L. Beck is with the Coordinated Science Lab, University of Illinois at Urbana Champaign, Urbana, IL 61801 USA.

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

Digital Object Identifier 10.1109/TAC.2011.2166713

Problems related to dynamic coverage are considered in [1], [2], [6], [8], where the emphasis is on distributed implementations, i.e., under limited information flow between individual elements. Distributed schemes have arisen as the underlying computational costs incurred for centralized schemes are impractical and inviable for implementation in some applications. For instance, limited sensor ranges prohibit centralized schemes in many sensor network scenarios. However, the distributed algorithms are prone to converge to one of the many local minima that typically riddle the coverage functions. As a result, their performance is very sensitive to initial placement of the resource locations. Further there is scant research that addresses the development of algorithms for problems of a non-distributed nature, or that aim simultaneously to attain global solutions and maintain low computational expense.

In spite of growing interest and wide-ranging applications, dynamic clustering problems are still not characterized or defined in a general framework. The terms *cluster* and *coverage* are subjective and usually have different interpretations, depending on the context of the problem. In some cases, a natural quantification of coverage is not easy to obtain, which results in added computational difficulty. For instance, in [21], the authors propose the concept of a micro-moving cluster (MMC), which denotes a group of objects that are not only close to each other at the current time, but also likely to move together for a while. Each MMC maintains a bounding box, which is a measure of its size. If the size of the bounding box exceeds a certain threshold, the MMC is split. Different clustering algorithms (for static data) can then be implemented on the MMC, instead of the individual points. However, because of the periodic maintenance of bounding boxes, the number of maintenance events dominates the computation time of the algorithm. The approach in [22] relies on a tradeoff between cluster quality and efficiency, but often results in clusters with large radii [23]. In [24], methods are proposed that automatically determine clusters from the historical trajectories of moving objects. These methods compare clusters at consecutive snapshots and determine the moving clusters. However, the comparison costs for such techniques are high. In [25], a histogram based clustering paradigm is proposed, however the authors observe that the task of updating the histograms severely slows down the computation.

In this paper, we provide a general framework based on the maximum entropy principle (MEP) to formulate and solve dynamic clustering problems, and which addresses both coverage and tracking aspects. The framework we propose, which we refer to as the Dynamic-Maximum-Entropy (DME) framework, adapts the notion of coverage to the dynamic setting, resolves the inherent trade-off between the *resolution* of the clusters and the computational cost, and provides flexibility to incorporate various dynamic specifications, while avoiding shallow local minima. The algorithms we propose are hierarchical in that they progressively seek finer subclusters from larger clusters. This hierarchical-clustering property of algorithms is developed heuristically to avoid local minima and exhibit insensitivity to initial placement of the resources. 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. More specifically, we define the notion of instantaneous coverage in a combinatorial optimization framework, and propose algorithms that achieve progressively better coverage with time. The computational complexity of these algorithms is reduced by exploiting the structure of the problem. 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. Simulations that employ algorithms based on this framework show improvements in computation times by as much as eight times over other methods. The simulations further demonstrate the flexibility of the framework, where coverage issues such as resolution of clusters are addressed, that other existing methods do not address.

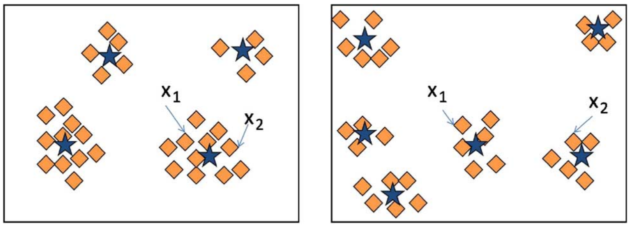
In fact, the MEP-based approach, when applied to the *static* setting, is equivalent to the *law of minimum free energy* from physical chemistry and thermodynamics. In certain chemical systems [26], nature employs the same mechanism in allocating distributions, called Gibbs distributions, over the set of all possible energy configurations. This law tells us that the energy states (i.e., cluster centers) should be such that the free energy (equivalently the Lagrangian) is minimized. Thus, this heuristic, besides being validated by the simulation results, is conceptually motivated by the laws of physics that apply to similar combinatorial problems in natural phenomena. It should be noted that the MEP-based approach in the static setting has been developed in the context of vector quantization (VQ) problems in data-compression. The resulting algorithm, known as the *deterministic annealing* (DA) algorithm, performs excellently in avoiding local minima, provides provably better clustering results than the popular k-means algorithms, and typically converges faster than other heuristics designed to avoid local minima such as the simulated annealing algorithm [26]. In this paper, the DME framework inherits the advantages of the DA algorithm and resolves the tracking and resolution issues originating from cluster dynamics.

This paper is organized as follows. We discuss the general setting of the dynamic coverage problem in Section II and highlight the key issues and challenges. In Section II-D, we present the DA algorithm developed in [26] for clustering static data and analyze those features that are relevant to the dynamic problem. Our DME framework for solving the clustering problem for dynamic data is presented in Section III. Implementation and simulation results for a variety of data sets are presented in Section IV. Directions for increasing the computational efficiency of the algorithm, improving its robustness, and additional future work are discussed in Section V. Finally we conclude the paper by revisiting the most important results obtained herein and identifying future goals.

II. Dynamic Coverage and Clustering Problems

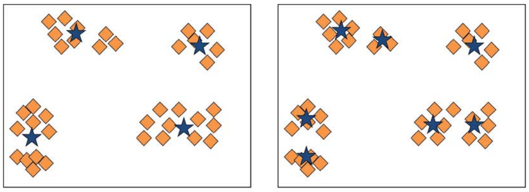
A. Problem Setting

Consider a scenario where the problem is to detect and track a group of moving objects (henceforth referred to as *sites*) in a given area. An illustration is given in Fig. 1. The objects may move in clusters and change cluster associations, and the clusters themselves can split and rejoin over time. For instance, sites x_1 and x_2 are in the same cluster at the time instance shown in



(a) (b)

Fig. 1. Clustering moving objects in a given area: in (a) and (b) two snapshots are shown (each at a distinct time instance) of an area with dynamic sites and resources. The squares and stars denote the positions of the sites and resources respectively. In a typical coverage problem, the task is to identify clusters (within this domain) in real time and track them over time. Objects in different clusters may move away from one another, form new clusters or exhibit other such movement, thereby resulting in an increase or decrease in the number of clusters. Sites x_1 and x_2 reside in the same cluster in (a). A split causes them to be in different clusters at the time instance shown in (b).



(a) (b)

Fig. 2. Clustering solution in (a) identifies four coarse clusters centers $y_j,\ 1 \leq j \leq 4$ (blue colored stars) in the underlying data $x_i,\ 1 \leq i \leq 37$ (orange colored squares). On the other hand, the solution in (b) identifies 7 finer clusters at a higher resolution on the same data. The solution in (b) has lower distortion D than (a), but at the expense of higher computation time.

(a). Both x_1 and x_2 change their *resident* clusters between the time instances (a) and (b). Alternatively, this problem can also be posed as a coverage problem, where the aim is to successively identify representative object locations (henceforth referred to as *resource locations*), such that these 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 sites. Each resource can be thought of as a cluster center.

In this paper, we consider a domain $\Omega \subset \mathbb{R}^2$ with N mobile sites and M resources, $N \gg M$, on a time horizon in $[0\ \infty)$ (note that the approach we develop in this paper 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 $t \in \mathbb{R}$ 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. For notational convenience, we sometimes use x_i and y_j in place of $x_i(t)$ and $y_j(t)$, where the time dependence is inherently 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

$$\begin{aligned} \dot{x}(t) &= \boldsymbol{\phi}(x(t), y(t), t), x(0) = x_0\\ \dot{y}(t) &= u(t), y(0) = y_0\\ &\Updownarrow\\ \dot{\zeta} &= f(\zeta, t) \end{aligned} \tag{1}$$

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 (yet to be specified) of coverage is maintained.

B. Challenges and Objectives

One of the main challenges in addressing dynamic coverage problems stems from the difficulty in 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 adapt it to make it suitable to a dynamic setting. Distortion is a measure of the (weighted) average distance of the site locations to their nearest resource location. Thus assuming a static problem ($\phi(t) \equiv 0$, $u(t) \equiv 0$), the distortion is given by

$$D(x, y) = \sum_{x_i \in \Omega} p_i \left\{ \min_{1 \leq j \leq M} d(x_i, y_j) \right\} \tag{2}$$

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 typically represents a distance function; for instance this can be given by a 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.

Another challenge is clearly defining clusters, especially their size. The difficulty in defining clusters primarily stems from the fact that clusters are hierarchical in nature. Each cluster can be thought of as being comprised of smaller subclusters (Fig. 2), and in the limiting case, 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. This motivates assigning a notion of *resolution* of a cluster. Iterative algorithms that identify finer and finer (higher resolution) clusters, progressively reduce the coverage cost function, but do so at the expense of increased computation. Modeling the coverage function and defining clusters become even more challenging in a dynamic setting. In contrast to the static setting, the coverage function must adjust to, and appropriately reflect the expanding, shrinking, splitting, coalescing, and other such dynamic components of clusters, adding further variability in the control design.

Computational complexity, another major challenge confronting the solution of coverage problems, stems from the inherent non-convex nature of the distortion function (2). The dynamic distortion function is riddled with numerous local minima just as 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 resolved by the framework proposed in this section. We formulate a coverage problem and propose algorithms that determine the velocity fields $u(t)$, such that they *track* each cluster over time. If a cluster splits, the resources mimic this behavior, thereby maintaining coverage.

C. Frame-by-Frame Approach

Since the dynamic data is a time-indexed series of static data, the simplest approach conceptually is to perform static clustering periodically. However, if the time between two successive clustering events is short, 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 period is long the clustering obtained at the previous time-step will not provide adequate coverage, or tracking, in the interim to the current time-step. However, we discuss this approach here to provide insights into the challenges that are inherited by the dynamic problems. In the frame-by-frame approach, at each time t, we solve the following static resource allocation problem:

1) [P1]: Given N sites $x_i(t),\ 1 \le i \le N$ in a domain Ω with relative weights p_i, find the set of M resource locations $y_j(t),\ 1 \le j \le M$ that minimizes the distortion (at fixed time t), that is, find

$$\underset{y_j,\,1\le j\le M}{\arg\min}\left(\sum_{i=1}^{N} p_i\left\{\min_{1\le j\le M} d(x_i,y_j)\right\}\right). \tag{3}$$

Here, $d(x_i, y_j)$ represents an appropriate distance metric between the resource location y_j and the site x_i. For example, $d(x_i,y_j) = \|x_i - y_j\|^2 + \|\phi_i - u_j\|^2$, includes the velocities and the locations in the Euclidean distance. Minimizing (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 which 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 [26], [27].

In Section II-D, we describe the salient features of the DA algorithm [28] that provide a basis for our DME algorithm, which we introduce in Section III.

D. MEP-Based Approach for Clustering in a Frozen Frame – The DA Algorithm

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 [9], [29]. In Lloyd's algorithm, 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. It should be noted that the solution depends substantially on the initial allocation of resource locations, as in successive iterations the locations are influenced only by 'proximal' points of the domain and are virtually independent of 'distant' points. As a result, the solutions from this algorithm typically get trapped at local minima.

The DA algorithm overcomes the local influence of domain elements by allowing each $x_i \in \Omega$ to be associated with every resource location y_j through a weighting parameter $p(y_j|x_i)$ (without loss of generality, these satisfy $\sum_j p(y_j|x_i) = 1$ for every $1 \le i \le N$) [19], [26], [27], [30]. Thus, the DA algorithm eliminates the hard partitions of Lloyd's algorithm, and seeks to minimize a modified distortion term given by

$$D(x,y) = \sum_{i=1}^{N} p_i \sum_{j=1}^{M} d(x_i,y_j)p(y_j|x_i). \tag{4}$$

We note that the instantaneous weighting term $p(y_j|x_i)$ (at time t) can alternatively be viewed as an association probability between the mobile site x_i and the mobile resource y_j. The choice of the weights $p(y_j|x_i)$ is crucial in determining the trade-off between the diminishing effect of the local influence and the deviation of the distortion (4) from the original cost function (2). For instance, a uniform weighting function $\{p(y_j|x_i)\} = 1/M\ \forall\, i \le N, j \le M$ makes the minimization of (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 (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 (4) identical to the cost function (2) but retains the "local-influence-effect" when minimizing with respect to y_j (see Fig. 3). 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 [31], [32]. More specifically, we seek a distribution $p(y|x)$ that maximizes the Shannon Entropy [33]

$$H(y|x) = -\sum_{i=1}^{N} p_i \sum_{j=1}^{M} p(y_j|x_i)\log(p(y_j|x_i)) \tag{5}$$

at a given (feasible) level of coverage $D(x,y) = D_0$ indicated by the distortion D. The entropy term quantifies the level of randomness in the distribution of association weights, and thus maximizing this term causes the weights to be maximally noncommittal toward any single cluster. Therefore 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 (5) and (4) respectively, D_0 is the distortion value that the algorithm aims, and β is a Lagrange multiplier. This is equivalent to the minimization problem

$$\min_{\{p(y_j|x_i)\}} \underbrace{D(x,y) - TH(y|x)}_{\triangleq F} \tag{6}$$

where the Lagrange multiplier denoted by $T = 1/\beta$ and the term F are called *temperature* and *free energy* respectively, due to a close analogy to quantities in statistical thermodynamics [34]. The MEP theorem (see Appendix and [31], [32]) gives an explicit solution for the weights, given by the Gibbs distribution

$$p(y_j|x_i) = \frac{\exp\{-\beta d(x_i,y_j)\}}{\sum\limits_{k=1}^{M}\exp\{-\beta d(x_i,y_k)\}}. \tag{7}$$

On substituting this distribution of weights into the Lagrangian (free energy), we obtain the following cost function:

$$F(x,y) = -\frac{1}{\beta}\sum_{i=1}^{N} p_i \log \sum_{k=1}^{M} \exp\{-\beta d(x_i,y_k)\}. \tag{8}$$

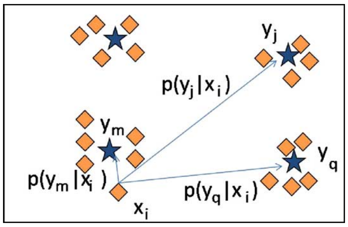


Fig. 3. Diminishing local influence of association weights. The site x_i is in close vicinity of the resource at location y_m, and considerably farther away from the resources at locations y_j and y_q. As a result, $p(y_m|x_i) \gg p(y_j|x_i)$ and $p(y_m|x_i) \gg p(y_q|x_i)$.

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

$$y_j = \sum_{i=1}^{N} p(x_i|y_j)x_i \text{ with } p(x_i|y_j) = \frac{p_i p(y_j|x_i)}{\sum_{m=1}^{N} p_m p(y_j|x_m)} \quad (9)$$

for $j = 1, 2, \ldots, M$ 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. Simple sensitivity analysis demonstrates that lower values of D_0 correspond to lower values of the temperature variable [32]. Clearly for small values of β (i.e., large values of T) in (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 the 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* (non-random) solution. An *annealing* process is incorporated where the minimization problem (6) is repeatedly solved at different values $\beta = \beta_k$ where $\beta_{k+1} > \beta_k$. Solving the implicit (9) to determine y_j forms the most computationally expensive step for each value of β_k. This equation is solved by evolving the following dynamical system to convergence:

$$y_j^k(n+1) = \sum_{i=1}^{N} p(x_i|y_j^k(n))x_i \quad 1 \le j \le M,\ n \ge 0 \quad (10)$$

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, we represent the converged value $y_j^k(\infty)$ by $y_j(\beta_k)$ when the parameter β needs to be emphasized and simply by y_j otherwise. Note that the iterative process (10) is equivalent to a Newton's descent method and accordingly, this procedure inherits the convergence properties of Newton's 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. For instance, it can be shown that there is only one resource location at $\beta = 0$ 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 for $\beta < \beta_c$ is one, while for $\beta > \beta_c$ it is more than one. This critical value β_c (that is, the condition for phase-transition) can be computed explicitly from the distribution of the sites x_i and resource locations y_j (the details are given later in this section). 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 in Appendix B and [26].

E. Features of the DA Algorithm

The concepts developed for the static coverage problem form the basis for our DME framework. In this formulation, various cluster features (such as relative sizes and shapes) arise naturally, thereby avoiding the need for new variables to characterize them. The weights $\{p(x_i|y_j)\}$ in (9) characterize soft clusters in this formulation. A high relative value of $p(x_i|y_j)$ implies that 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. The cluster mass is characterized by the 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 $\beta \to \infty$, the weights $\{p(y_j|x_i)\}$ become binary valued and the resulting clusters are no longer soft.

Remark: We define matrices $P_x \triangleq \operatorname{diag}(p(x_i))$ in $\mathbb{R}^{N\times N}$, $P_{y|x} \triangleq [p(y_j|x_i)]$ and $P_{x|y} \triangleq [p(x_i|y_j)]$ in $\mathbb{R}^{N\times M}$, and $P_y \triangleq \operatorname{diag}(p(y_j))$ in $\mathbb{R}^{M\times M}$. In this notation, $P_{x|y}$ and P_y carry the information about the relative shapes and sizes of the clusters. Also note that these matrices satisfy $P_x P_{y|x} = P_{x|y}P_y = P_{xy} = [p(x_i,y_j) = p_i p(y_j|x_i)]$. The expression for cluster centers given by (9) can be written concisely as $y = \check{P}_{x|y}^T x$ 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 $\check{P}_{y|x} \triangleq (I_2 \otimes P_{y|x})$, $\check{P}_x \triangleq (I_2 \otimes P_x)$, $\check{P}_y \triangleq (I_2 \otimes P_y)$, and $\check{P}_{xy} \triangleq (I_2 \otimes P_{xy})$). The *radius* of each cluster can be inferred from the magnitudes of the vectors $x_i - y_{c[i]}$ $(1 \le i \le N)$, where $y_{c[i]}$ denotes the resource location closest to the site x_i. We define $\bar{x} \triangleq x - \check{P}_{y|x}\check{P}_{x|y}^T x$ which determines the weighted average distance of x_i from the cluster centers $\{y_j\}$; that is, $\bar{x}_i = x_i - \sum_j p(y_j|x_i)y_j$. We summarize additional key features of the DA algorithm [26] in the following theorem and discuss their importance below.

Theorem 1: In the DA algorithm, the following hold:

1) *Centroid Property:* $\lim_{\beta\to 0} y_j = \sum_{i=1}^{N} p_i x_i$.
2) *Phase-Transition Property:* The resource locations $\{y_j\}$ given by (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 $1 \le j \le M$, where

$$C_{x|y_j} = \sum_{i=1}^{N} p(x_i|y_j)(x_i - y_j)(x_i - y_j)^T \quad (11)$$

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:* If the parameter value β is far from the critical values β_c, that is the minimum eigenvalue of $(I - 2\beta C_{x|y_j}) \geq \Delta > 0$ for $1 \leq j \leq M$, then $\|dy/d\beta\| \triangleq \left(\sum_j \|dy_j/d\beta\|^2\right)^{1/2} \leq c(\beta)/\Delta$, where $c(\beta)$ monotonically decreases to zero with β and is fully determined by β and the size of the space Ω.

Proof: See Appendix C for proof of (3) (the proofs for (1) and (2) can be found in [26]).

From the Centroid Property, it is clear that for very small values of β the algorithm places all resources at the weighted centroid of the sites. Furthermore, at $\beta = 0$, the cost function (6) achieves the global minimum with $p(y_j|x_i) = 1/M$, which results in all resource locations $\{y_j\}$ being placed at the centroid of the data set. The main rationale for the annealing process, which deforms the free energy F from the entropy function at $\beta = 0$ to the distortion function at $\beta = \infty$, is to obtain the global minimum at low values of β and then track the evolution of this minimum as β is increased. This heuristic is conceptually supported by the free energy principle in statistical thermodynamics [34], which motivates the inclusion of the entropy term as an appropriate choice to add to the distortion term for this homotopy.

The Phase-transition property guarantees that the resource locations obtained at non-critical β values of parameters are local minima of F. In this sense, the performance of this algorithm is at least as good as other algorithms such as Lloyd's algorithm. As $\beta \rightarrow \infty$, the DA algorithm becomes equivalent to Lloyd's algorithm, albeit that the choice of initial placement of locations (through the annealing process) is designed to achieve better minima. The phase transition property also explains the hierarchical nature of this algorithm. When the parameter β crosses a critical value β_c as β is increased, the number of *distinct* resource locations increase, which can also be viewed as a *splitting* process, where a single resource location splits into two or more resource locations. This process serves the purpose of identifying *natural* clusters where critical temperatures can be seen as indicators of cluster resolution: at $\beta = 0$ the algorithm identifies one natural cluster; and at the next critical value of β_c, subclusters of finer resolution are identified and successively after at each critical temperature finer and finer sub-clusters are identified.

From the sensitivity-to-temperature property, we conclude 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 $I - 2\beta C_{x_i|y_j}$, $1 \leq j \leq M$ being bounded away from zero corresponds to non-critical temperature values (Appendix C). The bounds given in this theorem are conservative; better bounds can be obtained by imposing additional assumptions on the data. For example, for data where the smallest distance between two distinct resource locations is μ, and there are no sites in the annulus around each resource described by $\{x|\bar{\rho}\mu < \|x - y_j\| \leq (1 - \bar{\rho})\mu\}$, a conservative bound proportional to $e^{-\beta\mu^2|1-2\bar{\rho}|}$ is obtained (Appendix D). In simulations, there is almost no change in resource locations as temperatures decrease between two successive critical temperatures, emphasizing the conservativeness of the bounds and avenues for analysis in making them stricter. This property has important consequences – namely the rate at which we change temperature values, that is the *cooling rate*, can be high. In simulations we typically change β geometrically, that is $\beta_k = \gamma^k\beta_0$ for some $\gamma > 1$. Another consequence of this property forms the basis for extending this MEP-based framework to the dynamic setting, which we discuss later.

Remark: Note that the static problem itself is NP-hard and the DA algorithm performs at least as well and typically much better than Lloyd's algorithm and its variants [26]. The cooling law for the annealing process is typically geometric and therefore the capability to avoid 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 [35]). Besides achieving better minima, the MEP-based framework has additional flexibility, which makes inclusion of problem-specific constraints easy. Notably, we adapted the framework in [13], [30] by adding a constraint 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, while preserving all other advantages of the DA algorithm.

III. Dynamic Clustering Problem and the DME Framework

In this section, we provide a general framework for formulating the dynamic clustering problem and determining computationally efficient algorithms that resolve issues of numerous local minima, quantifying coverage cost and cluster resolution, spatio-temporal smoothening, achieving trade-offs between computational cost and resolution, and flexibility to account for tracking clusters, their splits and real-time specifications of cluster-resolutions. Broadly stated, the aim of the DME framework is to successively identify and track clusters of moving objects. As noted earlier, using a frame-by-frame approach is not computationally viable, since it requires multiple iterations at each time step.

We consider free-energy as a metric for coverage, which together with the annealing process provides a basis for an algorithm that is independent of the initial choice of resource locations and better at avoiding local minima. A distinguishing feature of the free-energy based formulation is that it provides a method for quantifying cluster dynamics. The instantaneous cluster center z^c, which is determined solely by site-locations x_i (see (9)), is given by the recursion $z^c = \check{P}_{x|y}^{cT}x$, where $P_{x|y}^c = [p(x_i|z_j^c)] \in \mathbb{R}^{N \times M}$ (see Section II-E for details on notation). Here, we have deliberately represented the cluster center by z^c to distinguish it from the instantaneous resource location y. Since cluster centers, shapes, and mass are specified by $z^c = \check{P}_{x|y}^{cT}x$, $\check{P}_{x|y}^c$ and $\check{P}_y^c$ respectively (see Section II-E), 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.

The main conceptual hurdle in adapting the static concepts for the dynamic setting comes from choosing appropriate cooling rates relative to the time-scales of the site dynamics. For example, we can design the rate of cooling to be much faster than the given dynamics of the sites, resulting in an algorithm which is similar to the frame-by-frame approach. The sensitivity-to-temperature property in Theorem 1 leads to a solution to this problem. Since the resource locations are insensitive to temperatures between two successive critical temperatures, the design of the cooling laws is not critical between these critical temperatures. Critical temperatures are indicative of splits and resolution, thus decreasing temperature values matter only for forcing splits or obtaining higher resolution. In a 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, splitting conditions can also be reached due to dynamics of x and y, in addition to changing the values of β.

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 which 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.

The main results presented in this section for tracking are briefly summarized below.

P1 Under the assumption that the site-dynamics given by $\phi(\zeta, t)$ in (1) are continuously differentiable, we provide a control law $u(\zeta)$ such that the resource locations asymptotically track the cluster centers (Theorem 3).

P2 This control law is non-conservative, 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 (1) (see Theorem 4).

P3 If we further assume the clusters are *consistent* (that is, the average cluster size is non-increasing and clusters are separated), then the prescribed control that achieves asymptotic tracking is *bounded* (see Theorem 5).

Note that achieving the results stated in P1 and P2 requires only mild assumptions on the site-dynamics. For P3 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 $\bar{\phi}^T\bar{x}^c \leq 0$ where $\bar{\phi} = \check{P}_x^c\phi$ and $\bar{x}^c = x - \check{P}_{y|x}^c\check{P}_{x|y}^{cT}x$ (see Section II-E for notation). 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), and consequently, $\bar{\phi}^T\bar{x}^c \leq 0$ implies that velocity ϕ is such that magnitudes of these vectors are nonincreasing (see Fig. 4).

We first elaborate on properties of the free energy term, which form the basis for addressing both the problems of tracking cluster centers and monitoring splitting conditions.

A. Free Energy Properties

The crux of the DA algorithm for the static setting is that it replaces the notion of distortion in (2) by the free energy in (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 non-positive 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 conditions under which the control authority is lost, that is where $\partial F/\partial y = 0$, provide direct connections to the splitting conditions and cooling laws that we encountered in the static setting. We summarize the properties of free energy and its time derivative in the following theorem.

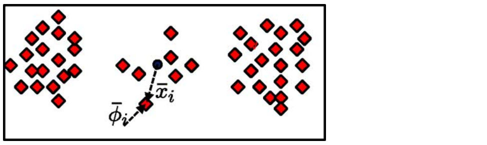


Fig. 4. Condition on the velocity field ϕ for consistent clusters. The solid circle represents the weighted cluster centroid x_i^c that is closest to the ith site with velocity ϕ_i, i.e., $\bar{\phi}_i = p_i\phi_i$. The condition $\bar{\phi}^T\bar{x}^c < 0$ imply that the angle between $\bar{x}_i^c$ and $\bar{\phi}_i$ is obtuse (in average sense) and hence the individual sites are directed toward the cluster and not away from it.

Theorem 2: Let F given by (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 (1). Then

1) *Positivity:* $F(\zeta) + (1/2\beta)\log M > 0$, for all ζ in $(\mathbb{R}^2)^{N+M}$.

2) *Structured derivative:* The derivative of the free energy term has the following structure:

$$\dot{F} = 2\zeta^T\Gamma(\zeta)f(\zeta),\ \ \Gamma = \begin{pmatrix} \check{P}_x & -\check{P}_{xy} \\ -\check{P}_{xy}^T & \check{P}_y \end{pmatrix}. \tag{12}$$

The matrix Γ is a symmetric positive semidefinite matrix for all ζ and it 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) *Lack of dynamic control authority at cluster centers:* The derivative $\dot{F}$ becomes independent of the control, that is, $(\partial/\partial u)(\dot{F}) = 0$ (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 $1 \leq j \leq M$.

Proof: See Appendix F.

B. Control Design for Tracking Cluster Centers

The resource locations are at cluster centers only when $y_j - \sum_i p(x_i|y_j)x_i$ is zero for each j, that is, only when $\bar{y} \triangleq y - \check{P}_{x|y}^T x = 0$. For design of u, we transform the coordinates $\zeta = (x, y)$ to $\bar{\zeta} = (x, \bar{y})$, in which the dynamics in (1) and $\dot{F}$ given by (12) are rewritten as

$$\begin{matrix} \dot{x}(t) = \phi, \\ \dot{\bar{y}}(t) = u - \dot{\check{P}}_{x|y}^T x - \check{P}_{x|y}^T\phi \end{matrix} \ \Leftrightarrow\ \begin{matrix} \dot{\bar{\zeta}} = \bar{f}(\bar{\zeta}, t) \\ \therefore \dot{F} = \bar{x}^T\bar{\phi} + \bar{y}^T\check{P}_y\bar{u} \end{matrix} \tag{13}$$

where $\bar{u} = u - \check{P}_{x|y}^T\phi$. We exploit the affine dependence of $\dot{F}$ on $\bar{u}$ in (13) to make $\dot{F}$ nonpositive analogous to control based on control Lyapunov functions [36]–[38]. Specifically we choose $\bar{u} : \mathbb{R}^{2(N+M)} \rightarrow \mathbb{R}^{2M}$ from sets of the form

$$\bar{\mathcal{U}}(\alpha) = \left\{ \bar{u}(\bar{\zeta}) = -\left[K_0 + \frac{\alpha(\bar{\zeta}) + \theta(\bar{\zeta})}{\bar{y}^T \check{P}_y \bar{y}}\right] \bar{y}, \quad \bar{y} \neq 0, \text{ for some } \theta(\bar{\zeta}) \geq 0 \text{ and } K_0 > 0 \right\} \quad (14)$$

which are parameterized by functions $\alpha(\cdot) : \mathbb{R}^{2(N+M)} \rightarrow \mathbb{R}$. The following theorem establishes that the assumption of continuously differentiable ϕ in (1) (which ensures $\dot{F}$ is of bounded variation) is adequate for the control design to achieve asymptotic tracking of clusters.

Theorem 3: Tracking cluster centers: For the site-resource dynamics given by (13), if $u = \bar{u} + \check{P}_{x|y}^T \phi$, where $\bar{u} \in \bar{\mathcal{U}}(\bar{x}^T \bar{\phi})$, then $\dot{F} \leq 0 \; \forall \, t \geq 0$ and the resource locations asymptotically track the cluster centers, i.e., $\lim_{t \rightarrow \infty} \bar{y}(t) = 0$.

Proof: Since $\bar{u} \in \bar{\mathcal{U}}(\bar{x}^T \bar{\phi})$, this implies $\dot{F}$ given by (13) 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 represents the time dependence of free energy $F(\zeta(t))$ with a slight abuse of notation) is bounded below (from Theorem 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 E.1. Since $\dot{F} = -K_0 \bar{y}^T \check{P}_y \bar{y} - \theta(\bar{\zeta}) \Rightarrow K_0 \bar{y}^T \check{P}_y \bar{y} \leq |\dot{F}|$, and $\check{P}_y$ 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 $t \rightarrow \infty$. ■

It is evident from (14), that resource velocities of the form $u = \bar{u} + \check{P}_{x|y}^T \phi$ can take very large values near $\bar{y} \neq 0$. The following theorem emphasizes that this control design is not conservative, that is, it is impossible for any control design to achieve $\dot{F} \leq 0$ without growing unbounded near $\bar{y} = 0$ unless there exists a bounded element from $\bar{\mathcal{U}}(\bar{x}^T \bar{\phi}) + \check{P}_{x|y}^T \phi$ that guarantees $\dot{F} \leq 0$.

Theorem 4: Lipschitz property of control: If there exists a $\hat{u} : (\mathbb{R}^2)^{N+M} \rightarrow (\mathbb{R}^2)^M$ such that $\hat{u}$ is Lipschitz at $\bar{\zeta} = 0$ and $\dot{F} = \bar{x}^T \bar{\phi} + \bar{y}^T \check{P}_y (\hat{u} - \check{P}_{x|y}^T \phi) \leq 0$, then $u = \bar{u}_S + \check{P}_{x|y}^T \phi$ is also Lipschitz at $\bar{\zeta} = 0$, where $\bar{u}_S(\bar{\zeta}) \in \bar{\mathcal{U}}(\bar{x}^T \bar{\phi})$ given by

$$\bar{u}_S(\bar{\zeta}) = -\left[K_0 + \frac{\bar{x}^T \bar{\phi} + \sqrt{|\bar{x}^T \bar{\phi}|^2 + (\bar{y}^T \check{P}_y \bar{y})^2}}{\bar{y}^T \check{P}_y \bar{y}}\right] \bar{y}. \quad (15)$$

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

Proof: (Note: The following proof is analogous to the proof for Proposition 3.43 in [36]). Since $\check{P}_{x|y}$ and ϕ are Lipschitz, it is enough to show that $\bar{u}_S$ is Lipschitz at the origin. Since $\underline{\hat{u}} \triangleq \hat{u} - \check{P}_{x|y}^T \phi$ is Lipschitz at $\bar{\zeta} = 0$, there exists a δ-neighborhood $B_\delta \triangleq \{\bar{\zeta} \text{ s.t. } \|\bar{\zeta}\| \leq \delta\}$ and $\bar{k} > 0$ such that $\bar{x}^T \bar{\phi} + \bar{y}^T \check{P}_y \underline{\hat{u}} < 0$ and $\|\underline{\hat{u}}(\bar{\zeta})\| < \bar{k} \|\bar{\zeta}\| \; \forall \bar{\zeta}$ in B_δ. So if $\bar{x}^T \bar{\phi} > 0$, $|\bar{x}^T \bar{\phi}| \leq |\bar{y}^T \check{P}_y \underline{\hat{u}}| \leq \bar{k} \|\bar{y}\| \|\bar{\zeta}\|$, then $\|\bar{u}_S\| \leq (2\bar{k} + K_0 + 1)\|\bar{\zeta}\|$. So when $\bar{x}^T \bar{\phi} < 0$, we have $\|\bar{u}_S\| \leq (1 + K_0)\|\bar{y}\| \leq (1 + K_0)\|\bar{\zeta}\|$.
■

Theorems 3 and 4 are very general in the sense that they do not impose any conditions other than smoothness on ϕ. We obtain better bounds on the size of control if we assume additional conditions on the cluster dynamics.

Theorem 5: Bounded control: Let the cluster dynamics be such that they are size-consistent, that is $\bar{\phi}^T \bar{x}^c \leq 0$ and satisfy $\|y_{c[i]} - x_i\|^2 / \|y_j - x_i\|^2 \leq \delta < 1$ for all $1 \leq j \leq M$, $j \neq c[i]$, where $y_{c[i]}$ is the unique closest resource location to x_i, that is, $\|y_{c[i]} - x_i\| = \min_{1 \leq j \leq M} \|y_k - x_i\|$ for $1 \leq i \leq N$. Consider the control design $u(\bar{\zeta}) = \check{P}_{x|y}^T \phi + \left[K_0 + (\alpha + \sqrt{\alpha^2 + (\bar{y}^T \check{P}_y \bar{y})^2}) / \bar{y}^T \check{P}_y \bar{y}\right] \bar{y}$, where $\alpha = \bar{\phi}^T \bar{x} - \bar{\phi}^T \bar{x}^c$ and $K_0 > 0$. Then $\lim_{t \rightarrow \infty} \bar{y} = 0$ and $\|u(\bar{\zeta})\| \leq (2\nu^{-1} c_1(\|\bar{y}\|) \|x\| + 1)\|\phi\| + (K_0 + 1)\|\bar{y}\|$ for some quadratic function $c_1(\cdot)$. Therefore, if $\|\phi\|, \|x\| \leq c_2 < \infty$, then $\|u(\bar{\zeta}(t))\|$ is bounded.

Proof: Note that $\dot{F} = \bar{\phi}^T \bar{x} + \bar{y}^T P_y (u - \check{P}_{x|y}^T \phi) = \bar{\phi}^T \bar{x}^c + \alpha + \bar{y}^T P_y (u - \check{P}_{x|y}^T \phi)$. On substituting for u, we obtain $\dot{F} = \bar{\phi}^T \bar{x}^c - K_0 \bar{y}^T \check{P}_y \bar{y} - \sqrt{\alpha^2 + (\bar{y}^T \check{P}_y \bar{y})^2} \leq 0$. Using the same arguments in the proof for Theorem 3, we get $\lim_{t \rightarrow \infty} \bar{y} = 0$.

From E.3 $\|\check{P}_x \check{P}_{y|x} \check{P}_{x|y}^T - \check{P}_x \check{P}_{y|x}^c \check{P}_{x|y}^{cT}\| < \|\bar{y}\| c_1(\|\bar{y}\|)$ 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 $|\alpha \|\bar{y}\| / \bar{y}^T \check{P}_y \bar{y}| \leq \nu^{-1} c_1(\|\bar{y}\|) \|x\| \|\phi\|$. Since $\|u(\bar{\zeta})\| \leq \|\phi\| + (2|\alpha| \|\bar{y}\| / \bar{y}^T \check{P}_y \bar{y}) + (1 + K_0 \|\bar{y}\|)$, therefore $\|u(\bar{\zeta})\|$ is bounded by

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

where $\bar{y} \rightarrow 0$. ■

There is great flexibility in this design procedure where the free energy F is treated as a control Lyapunov function and u is designed to make $\dot{F} \leq 0$. Additional computationally efficient control laws can be devised by exploiting the properties of Γ[39]. For instance, we can guarantee exponential convergence of $\bar{y} \rightarrow 0$ by appropriately choosing the function $\theta(\bar{\zeta})$ in the control design (see Appendix G).

Remark: Note that the assumptions on cluster-dynamics in [P3] are 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; this is discussed in the next section.

C. Monitoring Cluster Centers

As the resource locations begin to track the cluster centers, cluster-monitoring is completed based on the following decision choices:

1) Decision to Split: A decision to split can be taken when the parameter β satisfies the condition $\beta^{-1} = 2\lambda_{\max}(C_{x(t_c)|y_j(t_c)})$. If time is frozen at $t = t_c$, this splitting condition implies that y is at a local maxima (or inflection point) of F, that is, 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 (Appendix B). The new locations are given by $\mathbf{y}_{\text{new}} = q_{\text{split}}(\mathbf{y})$ as defined in (24). Splitting does not pose problems for the dynamic implementation as the computation time to determine $q_{\text{split}}(\mathbf{y})$ is small when compared to computation of $u(t)$ (Section B).

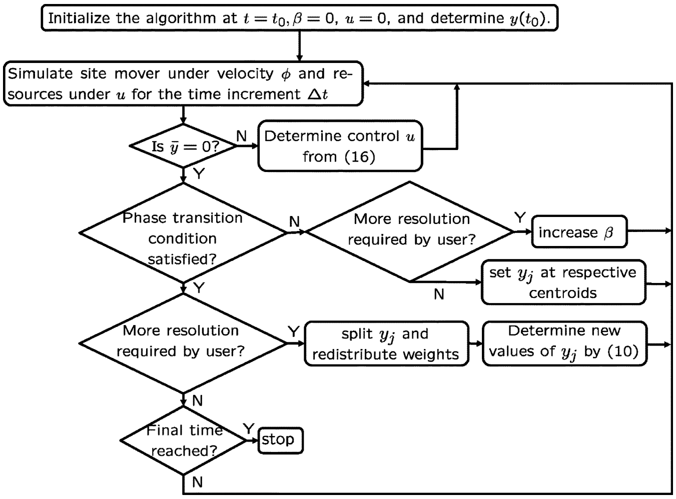

Initialize the algorithm at $t=t_0, \beta=0$, $u=0$, and determine $y(t_0)$.
Simulate site mover under velocity ϕ and resources under u for the time increment Δt
Is $\bar{y}=0$?
N
Determine control u from (16)
Y
Phase transition condition satisfied?
N
More resolution required by user?
Y
increase β
set y_j at respective centroids
N
Y
More resolution required by user?
Y
split y_j and redistribute weights
Determine new values of y_j by (10)
N
Final time reached?
Y
stop
N

Fig. 5. Flowchart showing the implementation of the proposed algorithm for tracking cluster centers and monitoring resolution. The vector $\|\bar{y}\|$ denotes the distance of resource locations to cluster centers. If $\bar{y} \neq 0$, control action (15) is applied to make the resource locations track the cluster centers. Once all resource locations are at cluster centers, the phase transition condition is evaluated (Theorem 1 (2) is satisfied). If no splitting is called for, the annealing parameter β is increased till the phase transition condition is satisfied when higher-resolution clustering is desired by the user; otherwise y_j are kept at cluster centers without changing β. When the phase-transition condition is satisfied, the resources are duplicated and new resource locations are determined by following the procedure in Section B of Appendix. The algorithm is terminated when an appropriate end condition is reached.

2) Decision to Track Cluster Centers With or Without Increasing Resolution: 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 $y_j(t) = \sum_i p(x_i(t)|y_j(t))x_i(t)$, or improve coverage-resolution by increasing cooling rates eventually leading to the splitting condition being satisfied, resulting in finer clusters, and therefore higher resolution.

D. Flexibility of the Framework

The distinct procedures and individual control over the decision choices for tracking cluster centers with or without splitting, results in an extremely flexible framework. This flexibility is essential, since as discussed in Section I, the task of efficiently clustering moving objects occurs in a variety of problems from different application areas, each of which have different constraints and requirements. The variability of constraints is accommodated by a user-based layer in our clustering framework, which enables the user to address the issues such as stopping criterion, resolution of the clustering solution, and number of clusters. For instance, a user can decide in real time to investigate clusters with more resolution; this can be effected by changing temperature values after tracking the clusters. The proposed framework can easily accommodate cases where different clusters need to be monitored with different resolutions by allowing multiple cooling laws – one for each resource location. Related simulations are discussed in Section IV. The flowchart in Fig. 5 gives a detailed description of the framework.

IV. Simulation Results and Discussion

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 allow for the performance analysis.

A. Implementation of the Basic Algorithm

We have summarized in Fig. 5, the main steps the algorithm traces to solve the coverage problem. This implementation is for the basic version of the algorithm, without any external user-based directives. All simulations were carried out in MATLAB. For simulations, the dynamics in (1) were run by discretizing them using a 4th-order Runge-Kutta method (RK-4) [40]. 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.

B. Hierarchical Natural Cluster Identification and Tracking

For the purpose of this simulation, we chose a scenario with 160 mobile sites. A velocity field $\phi(t,x)$ is chosen for each of the mobile sites, such that natural clusters emerge within a time horizon of 8 seconds.

The algorithm begins by placing one resource at the centroid of the sites (at $t = 0$) (a diamond denotes this location and squares represent the sites in Fig. 6(a)). As the site dynamics evolve, the site locations move according to the equation $\dot{x} = \phi(x,t)$. Our DME algorithm progressively updates the association probabilities and resource locations, and determines control values (i.e., the resource velocities from (15)) in order to track cluster centers. Fig. 6(b)–(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. Fig. 7(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. As expected, the proposed algorithm is considerably faster than the frame-by-frame approach. The computation times for five data sets are shown in Fig. 7(b). Depending on the distribution of the underlying data, the proposed 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. 7(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 5 times the computation time required for the proposed algorithm. Fig. 7(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).

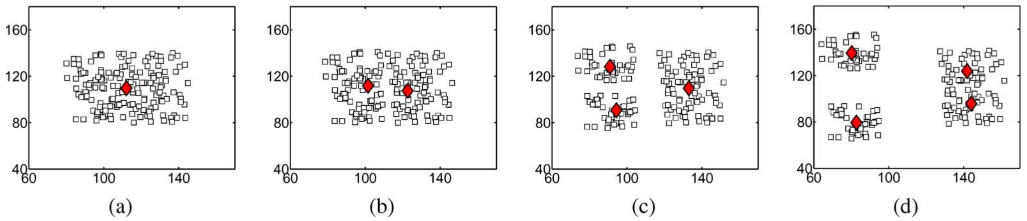


Fig. 6. Simulation results for data set 6 showing cluster identification and tracking during the time horizon. Snapshots (a), (b), (c) and (d) show the locations of mobile sites $x_i,\ 1 \leq i \leq 160$ (shown by squares) and resources $y_j,\ 1 \leq j \leq M$ (shown by diamond) at various phase transition 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, as seen in (d). The algorithm progressively identifies and tracks the clusters hierarchically.

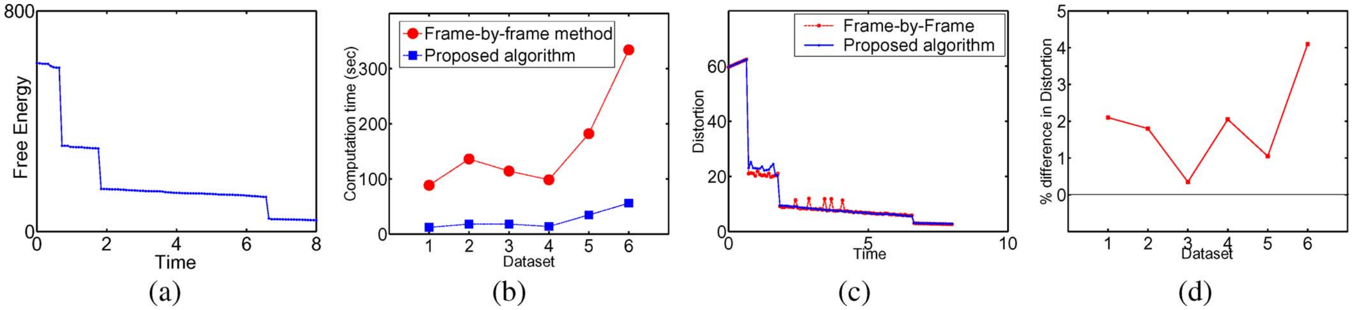


Fig. 7. (a) Free Energy F with respect to time. The control value u ensures that $dF/dt \leq 0$. Sharp decreases in F are due to phase transitions. Progressively decreasing Free Energy results in improved coverage and tracking throughout the time horizon. (b) Comparison of the computation times for the 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.

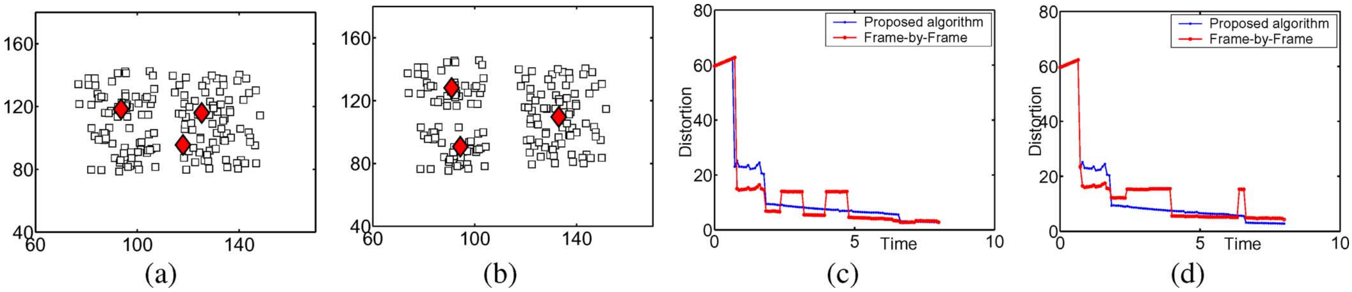


Fig. 8. (a) and (b): Clustering results from two successive frames using the frame-by-frame approach. In both instances, three resource locations were identified by the algorithm, but at considerably different positions. Such a solution violates the spatio-temporal requirement of the dynamic clustering 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 : $\Delta t = 0.08$ sec, frame-by-frame approach: $\Delta t = 0.24$ sec. (d) Proposed algorithm : $\Delta t = 0.08$ sec, frame-by-frame approach: $\Delta t = 0.48$ sec.

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. Fig. 8 shows the 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. Fig. 8(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. Simulation results for another data set is depicted in Fig. 9.

C. User-Based Directives

Once the natural clusters are identified and are being tracked, the user may desire higher resolution clustering and/or simultaneous tracking. Such 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 till the phase-transition condition is satisfied and the eventual splitting of resources is obtained. Depending on the user's desire, the annealing parameter may continue to increase to obtain further resource-locations splits and higher resolution. Fig. 10 shows the simulation results for the same data set as in Fig. 9, but under an additional user-based directive on the final clustering resolution. Both were

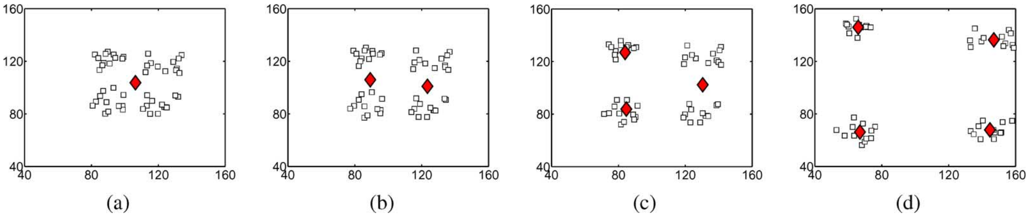


Fig. 9. Simulation results for data sets 2 and 3, showing cluster identification and tracking during the time horizon. The locations of mobile sites $x_i,\ 1 \leq i \leq N$ are shown by squares and the mobile resources $y_j,\ 1 \leq j \leq M$ are shown by diamonds, at various phase transition instances. Snapshots (a), (b), (c), and (d) show the clustering solution for the data set. Four natural clusters emerge at the end of the time horizon, as seen in (d). The algorithm progressively identifies and tracks the clusters in a hierarchical manner.

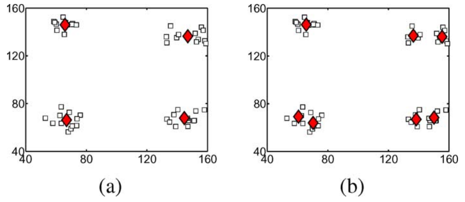


Fig. 10. Comparison of results obtained (a) without user-based resolution directives and (b) with user-based resolution directives. In (b), the user specified a higher final resolution and thus the algorithm identified and tracked finer clusters (and subclusters) during the same time horizon as that in (a).

executed for the same time-horizon, but resulted in different clustering resolutions.

V. Extensions and Future Work

A. Robustness to Modeling Uncertainties

1) Coverage Under Transmission Errors: In the DME framework, we have considered ideal transmission whereby the exact locations of the sites x_i are relayed to the resources. However, the effect of noisy channel transmissions can also be accommodated in our framework.

2) Communication-Link Failures: To address these failures, 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 jth resource. This is incorporated into the DME framework by modifying the instantaneous distortion term (2) to

$$\hat{D} = \sum_i \sum_j p_i p(y_j|x_i) p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j) \qquad (16)$$

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 take the form

$$p(y_j|x_i) = \frac{\exp\left(-\beta\left(\sum\limits_{\chi_{ij}} p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j)\right)\right)}{Z_i} \qquad (17)$$

where $Z_i = \sum_j \exp(-\beta(\sum_{\chi_{ij}} p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j)))$. The instantaneous free energy can now be rewritten as

$$\hat{F} = -\frac{1}{\beta} \sum_i p_i \log \sum_j \left[-\beta \sum_{\chi_{ij}} p_{\chi_{ij}}(\chi_{ij}) \chi_{ij} d(x_i, y_j)\right]. \qquad (18)$$

This term can now be used as a metric for instantaneous coverage under link failures. An analysis similar to that in Section II can be employed to obtain dynamic clustering.

3) Noisy Communication Channels: To address noisy transmission via a communication channel, we define a transition probability $p_{r|s}(x_k|x_i)$, to be the probability that the location information of site x_k is received (by all the resources) when actually the location of site x_i is sent. This is incorporated by modifying the instantaneous distortion term (2) as

$$\tilde{D} = \sum_i \sum_j p_i p(y_j|x_i) d'(x_i, y_j) \qquad (19)$$

where $d'(x_i, y_j) = \sum_k p_{s|r}(x_i|x_k) d(x_i, y_j)$ and $p(x_i|x_k) = p_{r|s}(x_k|x_i) p_i / \sum_l p_{r|s}(x_k|x_l) p_l$. That is, the average weighted distance replaces the notion of distance in the distortion term. Note that, $p_{s|r}(x_i|x_k)$ represents the probability of the event that x_i was actually sent given that the received message is interpreted as x_k. Now using the same procedure as above, the instantaneous free energy for noisy transmission can now be rewritten as

$$\tilde{F} = -\frac{1}{\beta} \sum_i p_i \log \sum_j \exp\left[-\beta\left(\sum_k p_{s|r}(x_i|z_k) d(x_i, y_j)\right)\right]. \qquad (20)$$

The DME framework can address robustness issues to a variety of transmission uncertainties. For instance, we can accommodate resource-dependent site-location uncertainties, where the probability of receiving the location x_k when actually the location x_i is sent, now depends on the receiver at the resource y_j, that is, the probability term now used is $p_{r|s}(x_k|x_i, y_j)$. Additionally, transmission errors in resource locations can also be addressed; this case is analogous to the noisy vector quantization problem addressed in [26].

Remark: In the framework presented, the velocity fields ϕ are assumed to be known exactly. In the case of noisy dynamics, where ϕ is given by perturbations $n(t)$ about a nominal function $\bar{\phi}$, that is where $\phi = \bar{\phi}(x, y, t) + n(t)$, and where the measurements of site and resource locations are noisy, control designs based on estimated values of (x, y) give 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. Another closely related problem is real time estimation of ϕ. Since the sites dynamics are completely realized by *velocity* fields, these fields, when not known a priori, can be estimated and $u(t)$ designed based on the estimated field. The inherent robustness in the algorithm due to the properties of the Gibbs distribution again yield satisfactory performance.

B. Extending the Class of Coverage Problems

1) Constraints on Resources: The resources in the framework presented are assumed to be identical. However, depending on the problem, the resources can be dissimilar, where different constraints apply to different resources, for instance, vehicles can be of different sizes with different coverage capacity. The resources can be made non-identical by introducing a weight λ_j to each resource location y_j. This interpretation yields a modified free energy function given by

$$F = -\frac{1}{\beta}\sum_i p_i \log \sum_j \lambda_j e^{-\beta d(x_i,y_j)}. \tag{21}$$

The constraints on resources are implemented by specifying constraints on λ_j. For instance setting $\lambda_j = W_j$, 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, vector quantization are respectively presented in [13], [26], [41]. The same formulation easily extends to the dynamic case, where static constraints can be incorporated by redefining the free energy in (21), and minimizing the appropriate Lagrangian $L = F + \sum_j \mu_j(\lambda_j - W_j)$ with respect to $\{y_j\}$ and $\{\lambda_j\}$.

2) Inertial Forces Into Vehicle Dynamics: In this paper, we have presented tracking of cluster centers when velocity fields (one state per direction for each vehicle) of sites are given. The same procedure is applicable even when higher order (i.e., multiple-state) differential equations are given. For instance, in the context of vehicle systems, the dynamics of these autonomous mobile agents is 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 $x_i = [\xi_i \eta_i]^T \in \mathbf{R}^2$ and M resource locations $y_j = [\rho_j \omega_j]^T \in \mathbf{R}^2$, with dynamics given by

$$\begin{aligned} \ddot{x}(t) &= \boldsymbol{\Upsilon}(x(t), \dot{x}(t), y(t), \dot{y}(t), t), \\ \ddot{y}(t) &= u(t) \end{aligned} \tag{22}$$

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 (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 of high values of θ, thus gives relatively more importance to velocities and hence yields cluster centers for instantaneous headings.

C. Numerical Issues in the Algorithm

1) Distributedness and Scalability: The framework presented in this paper aims at avoiding local minima. As a result, the computations are global in the sense that computation of each resource location requires values of all the site-locations. However, the contribution of distant site-locations becomes progressively lower 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" is zero. In this sense, the computation of resource locations change from being *truly global* to *truly local* as β is increased from zero to infinity. In [13], this feature is exploited to evolve a scalable algorithm that provides a close approximation to the DA algorithm. In the dynamic setting, clusters can interact even when β values are high, due to site dynamics. However 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.

2) Approximation of Gibbs Functions: One of the more costly steps in the algorithm is computing the exponentials in the Gibbs distribution. To address this issue, we can substitute the Gibbs distribution in (7) by a low complexity distribution that approximates it closely. As shown in [42], these low complexity distributions provide fast computation with performance nearly equivalent to the Gibbs distributions. One such distribution is the variable-width triangle distribution, given by

$$\hat{p}(y_j|x_i) = \frac{R_x - d(x_i, y_j)}{NR_x - \sum\limits_{k=0}^{N-1} d(x_i, y_k)} \tag{23}$$

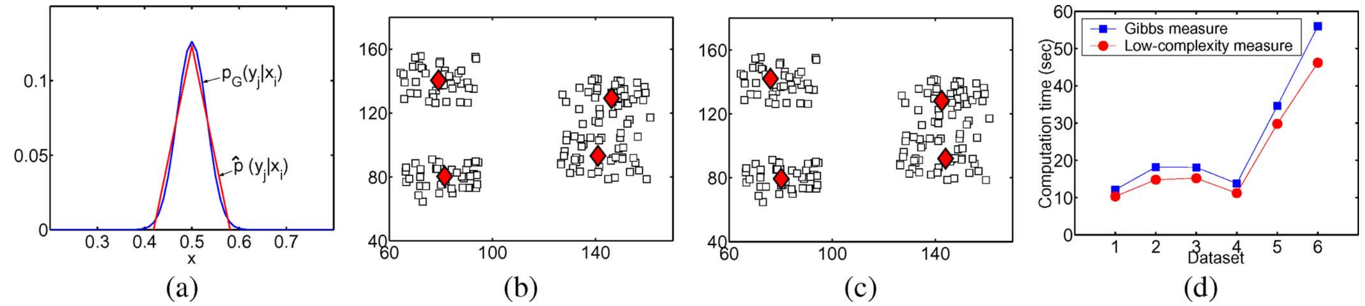
where R_x is determined for a given β by minimizing the L_1 distance between the approximate distribution $(\hat{p}(y_j|x_i))$ and the Gibbs distribution $(p_G(y_j|x_i))$. The computational cost incurred in computing one instance of the low complexity distribution using (23) requires $M + 7$ flops, where M is the number of resource locations in the span R_x. On the other hand, computing one instance of the Gibbs distribution in (7) requires $10M + 4$ flops. Fig. 11(a) shows a comparison between probability distributions generated using the exponential function $e^{-\beta z^2}$ and the low complexity triangle function $(R_x - z)/R_x$. The parameter R_x is computed by minimizing the Kullback-Leibler divergence [43] between the two distributions, which is also an upper bound on the L_1 distance between them, that is

$$D_{KL}(\hat{p}(y_j|x_i) \parallel p_G(y_j|x_i)) \geq \frac{1}{2\ln 2}\|\hat{p}(y_j|x_i) - p_G(y_j|x_i)\|_1^2$$

where equality is achieved when $\hat{p}(y_j|x_i) = p_G(y_j|x_i)$. This yields the value $R_x = \sqrt{9(\pi - 2)/\pi\beta}$, thereby prescribing a schedule for R_x in terms of the β schedule [42]. Simulation results and computation times using this approximate Gibbs distribution are presented in Fig. 11(b)–(d) As seen in Fig. 11(d), using such low-complexity distributions results in a decrease of 8%-18% in the computation time, while the clustering resolution deteriorating only by 3%-6%.

VI. Conclusion

In this paper, we have 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



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

Fig. 11. (a) Comparison between the Gibbs distribution ($p_G(y_j|x_i)$) and the low-complexity triangle distribution ($\hat{p}(y_j|x_i)$) (Comparison of the results obtained using (b) Gibbs distribution, (c) Low-complexity triangle distribution. Both (b) and (c) show the instantaneous positions of the sites and resources at the end of the time horizon. The final locations of the resources (shown by diamonds) do not vary considerably for the two cases. (d) Computation times for six data sets comparing the two cases.

naturally allow for quantification of inter and intra-cluster dynamics, and greatly facilitate the simultaneous design process for the coverage and tracking objectives.

Appendix

Maximum Entropy Principle: The MEP deals with ascribing a probability mass function for a vector-valued random variable x such that it guarantees that the expected values for a given set of m functions of the random variable $f_1(x), f_2(x), \ldots f_m(x)$ are equal to the m given constants $F_1, F_2, \ldots, F_m$, i.e., $\langle f_k(x)\rangle = F_k,\ 1 \leq k \leq m$. The MEP postulates that the probabilities should be chosen such that they maximize the Shannon entropy [33] and also satisfy the m expected value constraints.

Statement: The probabilities given by the Gibbs distribution $p_i = \exp\left\{-\sum_{j=1}^{m}\lambda_j f_j(x_i)\right\}/\left(\sum_{i=1}^{n}\exp\left\{-\sum_{j=1}^{m}\lambda_j f_j(x_i)\right\}\right)$, maximize the Shannon entropy, where constants λ_j are such that they satisfy $F_k = \langle f_k(x)\rangle = \sum_{i=1}^{n} p_i f_k(x_i),\ 1 \leq i \leq n$ and $1 \leq k \leq m$. For the proof and other details, see [31].

Implementation of the DA Algorith:

- Step 1: Initialize $\beta = 0$ and set limits on the maximum value of β (i.e., $\beta_{\max}$), and the maximum number of clusters M. Set the initial number of resources to 1, i.e., initialize $K = 1$ and cooling rate parameter γ to be constant greater than 1.
- Step 2: Determine resource locations $\mathbf{y} = \{y_j\}$ using (9).
- Step 3: Use the value of y_j to update the association weights $p(y_j|x_i)$ using (7) for $1 \leq j \leq K, 1 \leq i \leq N$.
- Step 4: Iterate between Steps 2 and 3 till convergence.
- Step 5: If $\beta > \beta_{\max}$, perform a last iteration at $\beta = \beta_{\max}$ and STOP. If not, then increase β, i.e., $\beta_{k+1} = \gamma\beta_k$.
- Step 6: If the number of resources (K) is less than M, check the phase transition condition for each cluster. If the condition is satisfied for a cluster y_j, split it by adding a resource location and redistributing the weights between them. Set K = K+1. After implementing this for each cluster, obtain the expanded set of resource locations $y_{\text{new}} = q_{\text{split}}(y)$ as in (24) described below. Set $y = y_{\text{new}}$.
- Step 7: Go to Step 2.

Procedure for Splitting Resource Locations: The implementation of the splitting process in Step 6 is carried out by replacing each resource location y_j by two new resource locations (say $y_{j,1}$ and $y_{j,2}$) by giving a local perturbation to each y_j, that is $y_{j,i} = y_{j,i} + \epsilon_i$, $\|\epsilon_i\| < \epsilon$, and ϵ is a small bound determined by the numerical precision of the implementation software. Both these resources are assigned a weight $p(y_j)/2$. These perturbed locations remain close to each other as long as the phase transition condition (in Theorem 1, part 2) is not reached, since y is a stable minima under these conditions. At the value of parameter β when the phase transition condition is satisfied for cluster j, i.e., $\beta^{-1} = 2\lambda_{\max}(C_{x|y_j})$, where $C_{x|y_j} = \sum_i p(x_i|y_j)(x_i - y_j)(x_i - y_j)^T$, for some j, the new resources $y_{j,i}$ slide to the new minima of F under the iterations given by (10). For the locations y_k for which the phase transition condition is not reached (at the current value of β), the perturbed locations $y_{k,i}$ are collapsed back to y_k. Thus a new set of distinct resource locations y_{new} is formed which comprises of split locations $y_{j,i}$ and the original "unsplit" locations. We represent this map ($y \rightarrow y_{\text{new}}$) by q_{split}

$$y_{\text{new}} = q_{\text{split}}(y). \tag{24}$$

Proof for Theorem 1:

[3. Sensitivity-to-Temperature Property] Proof: We obtain $dy_j/d\beta$ by differentiating (9) with respect to β. Pre-multiplying $dy_j/d\beta$ by $p(y_j)(dy_j/d\beta)^T$ and summing over the index j gives

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

$$= \underbrace{\sum_{ijk} p(y_j, x_i)(x_i - y_j)^T\frac{dy_j}{d\beta}\{p(y_k|x_i)d(x_i, y_k) - d(x_i, y_j)\}}_{T_2}$$

$$\underbrace{-\,2\beta\sum_i p_i\left(\sum_j p(y_j|x_i)(y_j - x_i)^T\frac{dy_j}{d\beta}\right)^2}_{T_3}.$$

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

$$\min_j\{p(y_j)\}\Delta\sum_j\frac{dy_j}{d\beta}^T\frac{dy_j}{d\beta} = \min_j\{p(y_j)\}\Delta\left\|\frac{dy}{d\beta}\right\|^2. \tag{25}$$

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

$$\sum_{ijk} p(y_j, x_i) p(y_k|x_i) \left[d(x_i, y_k) - d(x_i, y_j) \right] (y_j - x_i)^T \frac{dy_j}{d\beta}. \tag{26}$$

To obtain an upper bound on T_2, we first obtain

$$\begin{aligned} p(y_j|x_i) &= \frac{e^{-\beta(d(x_i,y_j)-d(x_i,y_k))}}{1+\sum\limits_{m\neq k} e^{-\beta(d(x_i,y_m)-d(x_i,y_k))}} \\ &\leq e^{-\beta(d(x_i,y_j)-d(x_i,y_k))}. \end{aligned} \tag{27}$$

Since $\|x - y_j\| \leq 2w_1 \triangleq 2\,\mathrm{diameter}(\Omega)$ and $|e^{-\beta(d(x_i,y_m)-d(x_i,y_k))}(d(x_i, y_m) - d(x_i, y_k))| \leq e^{-1}/\beta$ (since $\theta e^{-\gamma\theta} < e^{-1}/\gamma$ for $\gamma > 0$), we can get a bound on T_2 as

$$T_2 \leq \frac{e^{-1}}{\beta}(2w_1) \sum_j \left\| \frac{dy_j}{d\beta} \right\| \leq \left(\frac{e^{-1}}{\beta}(2w_1)\sqrt{M} \right) \left\| \frac{dy}{d\beta} \right\|. \tag{28}$$

From (25) and (28), we get that

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

which implies that $\|dy/d\beta\| \leq c(\beta)/\Delta$, where $c(\beta) = \left((e^{-1}/\nu\beta)(2w_1)\sqrt{M}\right)$ 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. ■

Less Conservative Bound for Sensitivity W.r.t Temperature: For a data set where the smallest distance between two distinct resource locations is μ and no sites are in the annulus described by $\{x|\bar{\rho}\mu < \|x - y_j\| \leq (1 - \bar{\rho})\mu\}$, the bound on $\|dy_j/d\beta\|$ is proportional to $e^{-\beta\mu^2|1-2\bar{\rho}|}$.

Proof: Since $p(y_k|x_i)$ is of the form $e^{-\beta d(x_i,y_k)}/\sum_j e^{-\beta d(x_i,y_j)}$

$$\begin{aligned} p(y_k|x_i) &= \frac{1}{1+\sum\limits_{j\neq k} e^{-\beta(d(x_i,y_j)-d(x_i,y_k))}} \\ &\leq \frac{1}{1+e^{-\beta d(x_i,y_j)} e^{\beta d(x_i,y_k)}}. \end{aligned} \tag{29}$$

Since there are no sites x in the annulus around each y_j described by $\{x|\bar{\rho}\mu < \|x - y_j\| \leq (1 - \bar{\rho})\mu\}$, this implies that $d(x_i, y_j) \leq \bar{\rho}^2\mu^2$ and $d(x_i, y_k) \leq (1 - \bar{\rho})^2\mu^2$. Thus

$$p(y_k|x_i) \leq \frac{1}{1+e^{-\beta\bar{\rho}^2\mu^2} e^{\beta(1-\bar{\rho})^2\mu^2}} \leq e^{-\beta\mu^2(1-2\bar{\rho})}. \tag{30}$$

Using this bound in (26) and the bound obtained in (25), we infer that the bound on $|T_2|$ is of the form $\|T_2\| \leq \hat{c}(\beta)\,\|dy/d\beta\| \Rightarrow \|dy/d\beta\| \leq \hat{c}(\beta)/\Delta$, where $\hat{c}(\beta) = e^{-\beta\mu^2(1-2\bar{\rho})}$.

Some Useful Lemmas:

Lemma E.1: For a nonnegative function $g : \mathbb{R} \to \mathbb{R}$ of bounded variation, if $\int_0^\infty g(\tau)d\tau < \infty$, then $\lim_{t\to\infty} g(t) = 0$.

Proof: Suppose $\lim_{t\to\infty} g(t) \neq 0$. This implies $\exists\, \epsilon > 0$ and a sequence $\{t_n\} \subset \mathbb{R}$ such that $g(t_n) > \epsilon$ and $t_{n+1} - t_n > 1 \forall n \in \mathbb{N}$. Since $h(t) = \int_0^t g(\tau)d\tau$ is an increasing function that is bounded above by $a \triangleq \int_0^\infty g(\tau)d\tau$, therefore $h(t)$ converges to a as $t \to \infty$. Hence, for every $\epsilon' > 0$, there exists a $T_{\epsilon'}$ in $\mathbb{R}$ such that $a - h(T_{\epsilon'}) < \epsilon'$, that is, $\int_{T'_\epsilon}^\infty g(\tau)d\tau < \epsilon'$. In particular, $\exists$ a sequence $\{T_n\} \subset \{t_n\}$ such that $\int_{T_n}^\infty g(\tau)d\tau < (0.5)^n\epsilon$. Since $T_{n+1} - T_n > 1$ and $\int_{T_n}^{T_{n+1}} g(\tau)d\tau \leq \int_{T_n}^\infty g(\tau)d\tau < (0.5)^n\epsilon$, there is at least one point $S_n \in (T_n, T_{n+1})$ such that $g(S_n) \leq (0.5)^n\epsilon$. The variation V_m of $g(t)$ over the sequence of intervals $\{(T_n, S_n)\}_{n=1}^m$ is

$$\begin{aligned} V_m = \sum_{k=1}^m |g(S_k) - g(T_k)| &= \sum_{k=1}^m g(T_k) - g(S_k) \\ &\geq m\epsilon - \sum_{k=0}^\infty (0.5)^k\epsilon = (m-1)\epsilon. \end{aligned}$$

Therefore, the variation given by $\sup_m V_m = \infty$, which implies g is not of bounded variation. This contradiction implies that $\lim_{t\to\infty} g(t) = 0$. ■

Lemma E.2: Let x in $\left(\mathbb{R}^2\right)^N$ and y in $\left(\mathbb{R}^2\right)^M$ be such that $\|y_{c[i]} - x_i\|^2/\|y_j - x_i\|^2 \leq \delta < 1, \forall\, 1 \leq j \leq M, j \neq c[i]$, where $y_{c[i]}$ is the unique closest resource location to x_i, i.e., $\|y_{c[i]} - x_i\| = \min_{1\leq j\leq M} \|y_k - x_i\|$ for $1 \leq i \leq N$. Then $|\partial p(y_j|x_i)/\partial y_k| < M\sqrt{2\beta(\delta^{-1} - 1)e}$ for all $1 \leq i \leq N$ and $1 \leq j,\ k \leq M$.

Proof: Since $p(y_j|x_i) = e^{-\beta\|y_j-x_i\|^2}/(\sum_l e^{-\beta\|y_l-x_i\|^2})$, we have

$$\begin{aligned} \frac{\partial p(y_j|x_i)}{\partial y_k} &= 2\beta p(y_j|x_i)[p(y_k|x_i)(y_k - x_i)^T - (y_j - x_i)^T] \\ &= \begin{cases} 2\beta p(y_j|x_i)(1-p(y_j|x_i))(y_j-x_i)^T & \text{if } k=j \\ 2\beta p(y_j|x_i)p(y_k|x_i)(y_k - x_i)^T & \text{if } k\neq j \end{cases}. \end{aligned}$$

Therefore $|\partial p(y_j|x_i)/\partial y_k| \leq 2\beta \min\{p(y_j|x_i)(1 - p(y_j|x_i))\|(y_j - x_i)\|,\ (1 - p(y_k|x_i))p(y_k|x_i)\|(y_k - x_i)\|\}$. Note that if $j = c[i]$, then

$$\begin{aligned} 1 - p(y_j|x_i) &= \frac{\sum\limits_{m\neq j} e^{-\beta(\|y_m-x_i\|^2-\|y_j-x_i\|^2)}}{1+\sum\limits_{m\neq j} e^{-\beta(\|y_m-x_i\|^2-\|y_j-x_i\|^2)}} \\ &\leq M e^{-\beta(\delta^{-1}-1)\|y_j-x_i\|^2} \end{aligned}$$

and if $j \neq c[i]$, then

$$\begin{aligned} p(y_j|x_i) &= \frac{e^{-\beta(\|y_j-x_i\|^2-\|y_{c[i]}-x_i\|^2)}}{1+\sum\limits_{l\neq c[i]} e^{-\beta(\|y_l-x_i\|^2-\|y_{c[i]}-x_i\|^2)}} \\ &< M e^{-\beta(\delta^{-1}-1)\|y_j-x_i\|^2}. \end{aligned}$$

Thus $p(y_j|x_i)(1 - p(y_j|x_i))\|(y_j - x_i)\| < M\sqrt{2\beta(\delta^{-1} - 1)e}$, since $e^{-cx^2}x$ attains its maximum at $x = 1/\sqrt{2c}$. Similarly $p(y_k|x_i)(1 - p(y_k|x_i))\|(y_k - x_i)\| < M\sqrt{2\beta(\delta^{-1} - 1)e}$. ■

Lemma E.3: There exists a constant $\bar{k}_2 < \infty$ such that $\|\check{P}_x\check{P}_{y|x}\check{P}_{x|y}^T - \check{P}_x\check{P}_{y|x}^c\check{P}_{x|y}^{cT}\| < 3k_2\|\bar{y}\|(1 + k_2\|\bar{y}\|^2 + k_2^2\|\bar{y}\|^2)$ for the assumptions in Lemma E.2.

Proof: Let $g_j(y) : \mathbb{R}^M \to \mathbb{R}$ be given by $g_j(y) = e^{-\beta\|a-y_j\|^2}\sum_k e^{-\beta\|a-y_k\|^2}$ for some $a \in \Omega$. From multivariate mean-value theorem, $|g_j(y_1) - g_j(y_2)| \leq \|(\partial g_j/\partial y)(y^\star)\|\|y^1 - y^2\|$ for any y^1, y^2 in Ω for some $y^\star$ in the line segment connecting y^1 to y^2. Since $\|\partial g_j/\partial y\|$ is bounded (E.2), we have $|g_j(y) - g_j(z^c)| \leq k_1\|y - z^c\|$, for some k_1 in $\mathbb{R}$ for $z^c = \check{P}_{x|y}^T x$ and $a = x_i$. So, each element of $P_{y|x} - P_{y|x}^c$ is bounded by $k_1\|\bar{y}\|$, implying that $\|P_{y|x} - P_{y|x}^c\|$ is bounded by $k_2\|\bar{y}\|$. Also since $P_y - P_y^c =$

$[1\cdots 1](P_{y|x}-P^c_{y|x})P_x$, we have $\|P_y-P^c_y\|\leq k_2\|\bar{y}\|$ and $\|P_y^{-1}-P_y^{-1c}\|=\|(P_yP^c_y)^{-1}(P^c_y-P_y)\|\leq (k_2/\nu^2)\|\bar{y}\|$. Thus

$$\begin{aligned}\|\check{P}_x\check{P}_{y|x}\check{P}^T_{x|y}-\check{P}_x\check{P}^c_{y|x}\check{P}^{cT}_{x|y}\| &= \|\check{P}_x(\check{P}_{y|x}\check{P}_y^{-1}\check{P}^T_{y|x}-\check{P}^c_{y|x}\check{P}_y^{-1c}\check{P}^{cT}_{y|x})\check{P}_x\|\\ &\leq 3\bar{k}_2\|\bar{y}\|(1+\bar{k}_2\|\bar{y}\|^2+\bar{k}_2^2\|\bar{y}\|^2)\end{aligned}$$

where $\bar{k}_2=k_2/\nu^2$.

Proof for Theorem 2:

[1. Positivity] Proof: $F+(1/\beta)\log M = -(1/\beta)\sum_i p_i\log(\sum_k e^{-\beta d(x_i,y_k)}/M)\geq 0$, since $|\sum_k e^{-\beta d(x_i,y_k)}|\leq M$ for all x,y in Ω. ■

[2. Structured Derivative] Proof: From (8), we have

$$F=-\frac{1}{\beta}\sum_i p_i\log\sum_j e^{-\beta((\xi_i-\rho_j)^2+(\eta_i-\omega_j)^2)}. \tag{31}$$

On differentiating w.r.t the components ξ,ρ,η and ω, we obtain

$$\left(\frac{\partial F}{\partial\zeta}\right)^T = 2\zeta^T\begin{pmatrix}\check{P}_x & -\check{P}_{xy}\\ -\check{P}^T_{xy} & \check{P}_y\end{pmatrix}. \tag{32}$$

Therefore from (1) and (32), $\dot{F}=(\partial F/\partial\zeta)\dot{\zeta}=2\zeta^T\Gamma f(t,\zeta)$. $\Gamma=\Gamma(\zeta)$ is a nonlinear, state-dependent matrix, and possesses algebraic structure which makes analysis straightforward.

[Γ is a Symmetric Positive Semidefinite Matrix for All ζ. It 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 $\rho(W)=1$] Proof: From the definition of Γ (32) it is clear that it is symmetric. Note that every element of P_x, P_y and P_{xy} is nonnegative which implies that every off-diagonal term in Γ is nonpositive; i.e.

$$\begin{aligned}\Gamma &\in Z_{2N+2M}\\ &\triangleq \{Q=[q_{ij}]\in(\mathbb{R}^2)^{(N+M)\times(N+M)}\ \text{s.t. } q_{ij}\leq 0 \text{ for } i\neq j\}.\end{aligned}$$

This implies that $\exists\ \bar{W}=[\bar{w}_{ij}]\in(\mathbb{R}^2)^{(N+M)\times(N+M)}$ and $\alpha>0$ in $\mathbb{R}$ such that $\Gamma=\alpha I-\bar{W}$ where $\lambda_{\max}(\bar{W})\leq\alpha$ and $\bar{w}_{ij}\geq 0\ \forall\, i,j$ (see Lemma 2.5.2.1 in [44]). Choose $W=(1/\alpha)\bar{W}$. This implies $\Gamma=\alpha(I-W)$, $\lambda_{\max}(W)\leq 1$ and W is element-wise nonnegative. Also W is symmetric. Furthermore

$$\begin{aligned}\Gamma e_{2N+2M} &= \begin{pmatrix} I_2\otimes P_x & -I_2\otimes P_{xy}\\ -I_2\otimes P^T_{xy} & I_2\otimes P_2\end{pmatrix}\begin{pmatrix} e_2\otimes e_N\\ e_2\otimes e_M\end{pmatrix}\\ &= \begin{pmatrix} e_2\otimes(p_x-p_x)\\ e_2\otimes(p_r-p_r)\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}\end{aligned}$$

which implies $We_{2N+2M}=e_{2N+2M}$ and since Γ is symmetric, W is symmetric. Thus $e^T_{2N+2M}W=e^T_{2N+2M}$; this implies W is a symmetric doubly-stochastic with $|\lambda_{\max}(W)|=1$, which in turn implies that Γ is positive semidefinite. ■

[3. Lack of Dynamic Control Authority at Cluster Centers] Proof: Equation (32) can be simplified to

$$\frac{dF}{dt}=\bar{x}^T\bar{\phi}+\bar{y}^T\check{P}_y\bar{u} \tag{33}$$

where $\bar{x}=x-\check{P}_{y|x}\check{P}^T_{x|y}x$, $\bar{\phi}=\check{P}_x\phi$, $\bar{y}=y-\check{P}^T_{x|y}x$, and $\bar{u}=u-\check{P}_{x|y}\phi$. Thus, dF/dt becomes independent of control u, that is, $\partial\dot{F}/\partial u=0$ at those time instants t_c when $\bar{y}(t_c)=0$. Note that $\bar{y}(t_c)=0\Rightarrow y(t_c)=\check{P}^T_{x|y}x$

$$\Rightarrow y_j(t_c)=\sum_{i=1}^N p(x_i(t_c)|y_j(t_c))x_i(t_c),\ \text{ for } 1\leq j\leq M \tag{34}$$

which is identical to the centroid condition that we had previously (the control authority can also be lost if the orthogonality condition $\bar{y}^T\bar{u}=0$ occurs even when $\bar{y}\neq 0$; however the proposed design is such that $\bar{y}^T\bar{u}$ is not zero unless $\bar{y}=0$).

Control for Exponential Convergence of $\bar{y}(t)$ to 0: We choose $u=\bar{u}+\check{P}^T_{x|y}\phi$ with $\bar{u}$ in $\bar{\mathcal{U}}(\bar{x}^T\bar{\phi})$ with $\theta(\bar{\zeta})=\sqrt{(\bar{x}^T\bar{\phi})^2+(\bar{y}^T\bar{y})^2}+2M_1(\|x\|+\|\bar{y}^2\|)$, where M_1 is a bound assumed on $\|\dot{\check{P}}_{y|x}\|_\infty\triangleq\max_{ij}|\check{P}_{y|x}[i,j]|$. Since $P_y=[1\ \cdots\ 1]P_xP_{y|x}$ and $P_xP_{y|x}=P_{x|y}P_y$, we have $\|\dot{\check{P}}_y\|_\infty\leq\|\dot{\check{P}}_{y|x}\|_\infty\leq M_1$ and $\|\check{P}_y\dot{\check{P}}_{x|y}\|_\infty\leq\|\dot{\check{P}}^T_{y|x}\check{P}_x-\dot{\check{P}}_y\check{P}_{x|y}\|_\infty\leq 2M_1$. With this control, we first show that $\bar{y}$ converges exponentially to zero with time, that is, the resource locations converge exponentially to cluster centers. Note that if we define $V_1(\bar{y})=(1/2)\bar{y}^T(I_2\otimes P_y)\bar{y}$, then

$$\begin{aligned}\dot{V}_1 = -K_0\bar{y}^T\check{P}_y\bar{y} &- \left(\sqrt{(\bar{x}^T\bar{\phi})^2+(\bar{y}^T\check{P}_y\bar{y})^2}-\bar{x}^T\bar{\phi}\right)\\ &-\left(2M_1\|x\|-\check{P}_y\dot{\check{P}}^T_{x|y}x\right)-\left(2M_1\bar{y}\check{P}_y\bar{y}-\bar{y}^T\dot{\check{P}}_y\bar{y}\right)\end{aligned}$$

which implies $\dot{V}_1\leq -K_0V_1$. Therefore $V_1(t)\leq e^{-K_0t}V_1(0)$ which implies $\|\bar{y}\|\leq\nu^{-1}e^{-0.5K_0t}\|\bar{y}(0)\|$, where $\nu>0$ is the assumed lower bound on $\min_j\{p(y_j)\}$. Thus time Δt required for $\bar{y}(t)$ to decrease to $\delta\|\bar{y}(0)\|$ is given by $|2\log(\nu\delta)|/K_0$.

To prove that this choice of u is bounded, we know $\|\bar{u}\|\leq(K_0+(2\|\bar{x}\|\|\bar{\phi}\|+\bar{y}^T\bar{y}+2M_1\|\bar{y}\|^2+M_1\|x\|)/\bar{y}^T\bar{y})\|\bar{y}\|$. Therefore at time t_0, if the sites and resources are at x_0 and y_0, then at time $t=t_0+\Delta t$, $\|x(t)\|\leq\|x_0\|+c_2|\log(\delta)|/\alpha$ where c_2 is an upper bound on $\|\phi\|$ and $\|\bar{y}(t)\|\leq\delta\|y_0\|$. Hence

$$\|\bar{u}\|\leq(K_0+2M_1+1)\delta\|y_0\|+\frac{2c_2+M_1}{\delta\|y_0\|}\left(\|x_0\|+c_2\left|\frac{\log(\nu\delta)}{K_0}\right|\right).$$

References

[1] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Feb. 2004.

[2] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proc. IEEE Conf. Decision Control*, 2004, pp. 3357–3363.

[3] C. Omar, M. Vaidya, T. W. Bretl, and T. P. Coleman, "Using feedback information theory for closed-loop neural control in brain-machine interfaces," in *Proc. 17th Annu. Computat. Neurosci. Meeting (CNS)*, Portland, OR, 2008, Invited Session on 'Methods of Information Theory in Computational Neuroscience', [CD ROM].

[4] P. Nunez and R. Srinivasan, *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford, U.K.: Oxford Univ. Press, 2006.

[5] B. Blankertz, G. Dornhege, M. Krauledat, K. Müller, and G. Curio, "The non-invasive Berlin Brain-Computer Interface: Fast acquisition of effective performance in untrained subjects," *Neuroimage*, vol. 37, no. 2, pp. 539–550, 2007.

[6] J.-B. Sheu, "A fuzzy clustering-based approach to automatic freeway incident detection and characterization," *Fuzzy Sets Syst.*, vol. 128, no. 3, pp. 377–388, 2002.

[7] R. R. Baker, *The Evolutionary Ecology of Animal Migration*. New York: Holmes & Meier, 1978.

[8] D. Chudova, S. Gaffney, E. Mjolsness, and P. Smyth, "Translation-invariant mixture models for curve clustering," in *Proc. 9th Int. Conf. Knowledge Discover Data Mining KDD'03)*, New York, NY, 2003, pp. 79–88.

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

[10] Z. Drezner, *Facility Location: A Survey of Applications and Methods*. New York: Springer Verlag, 1995.

[11] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, Dec. 1999.

[12] C. Therrien, *Decision, Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*, 1st ed. New York: Wiley, 1989, vol. 14.

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

[14] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewoods Cliffs, NJ: Prentice-Hall, 1998.

[15] J. Harigan, *Clustering Algorithms*. New York: Wiley, 1975.

[16] N. Elia and S. Mitter, "Stabilization of linear systems with limited information," *IEEE Trans. Autom. Control*, vol. 46, no. 9, pp. 1384–1400, Sep. 2001.

[17] S. Mitter, "Control with limited information: The role of systems theory and information theory," *IEEE Inform. Theory Soc. Newsl.*, vol. 50, no. 4, pp. 1–23, Dec. 2000.

[18] S. Salapaka, "Combinatorial optimization approach to coarse control quantization," in *Proc. 45th IEEE Conf. Decision Control*, Dec. 2006, pp. 5234–5239.

[19] P. Sharma, S. Salapaka, and C. Beck, "Entropy based algorithm for combinatorial optimization problems with mobile sites and resources," in *Proc. Amer. Control Conf.*, 2008, pp. 1255–1260.

[20] R. Gray and E. Karnin, "Multiple local minima in vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 256–361, 1982.

[21] Y. Y. J. Li and J. Han, "Clustering moving objects," in *Proc. 10th ACM SIGKDD, Int. Conf. Knowledge Discovery Data Mining (KDD)*, 2004, [CD ROM].

[22] S. Har-Peled, "Clustering motion," *Discrete Computat. Geom.*, vol. 31, no. 4, pp. 545–565, 2003.

[23] C. S. Jensen, D. Lin, and B. C. Ooi, "Continuous clustering of moving objects," *IEEE Trans. Knowledge Data Eng.*, vol. 19, no. 9, pp. 1161–1174, Sep. 2007.

[24] P. Kalnis, N. Mamoulis, and S. Bakiras, "On discovering moving clusters in spatio-temporal data," in *Proc. Int. Symp. Spatial Temporal Databases*, 2005, pp. 364–381.

[25] Q. Zhang and X. Lin, "Clustering moving objects for spatio-temporal selectivity estimation," in *Proc. 15th Australasian Database Conf.*, 2004, pp. 123–130.

[26] K. Rose, "Deterministic annealing for clustering, compression, classification, regression and related optimization problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210–39, Nov. 1998.

[27] K. Rose, "Constrained clustering as an optimization method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 8, pp. 785–794, Aug. 1993.

[28] K. Rose, "Deterministic Annealing, Clustering, and Optimization," Ph.D. dissertation, California Insti. Technol., Pasadena, CA, 1991.

[29] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[30] P. Sharma, S. Salapaka, and C. Beck, "A maximum entropy based scalable algorithm for resource allocation problems," in *Proc. Amer. Control Conf.*, 2007, pp. 516–521.

[31] E. T. Jaynes, "Information theory and statistical mechanics," *Phys. Rev.*, no. 4, pp. 620–630, 1957.

[32] E. T. Jaynes, *Probability Theory – The Logic of Science*. London, U.K.: Cambridge Univ. Press, 2003.

[33] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: Univ. Illinois Press, 1949.

[34] L. D. Landau and E. M. Lifshitz, *Statistical Physics*, 3rd ed. Oxford, U.K.: Oxford Univ. Press, vol. 3, pt. 1.

[35] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[36] R. Sepulchre, M. Jankovic, and P. Kokotovic, *Constructive Nonlinear Control*. New York: Springer-Verlag, 1997 [Online]. Available: http://www.montefiore.ulg.ac.be/services/stochastic/pubs/1997/SJK97a

[37] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM J. Control Optim.*, vol. 21, pp. 462–471, 1983.

[38] E. D. Sontag, "A 'Universal' construction of Artstein's theorem on nonlinear stabilization," *Syst. Control Lett.*, vol. 13, no. 2, pp. 117–123, 1989.

[39] S. Salapaka, "On combinatorial optimization problems with mobile sites and resources," in *Proc. 44th IEEE Conf. Decision Control, Eur. Control Conf. (CDC-ECC '05)* , Dec. 2005, pp. 6978–6983.

[40] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. New York: Wiley, 2004.

[41] S. Salapaka and A. Khalak, "Locational optimization problems with constraints on resources," in *Proc. 41st Allerton Conf.*, Oct. 2003, pp. 1240–1249.

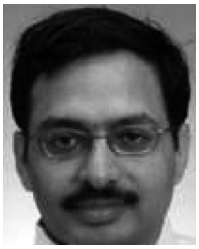
[42] K. Demirciler and A. Ortega, "Reduced-complexity deterministic annealing for vector quantizer design," *EURASIP J. Appl. Signal Processing*, vol. 12, pp. 1807–1820, 2005.

[43] S. Kullback, "The Kullback-Leibler distance," *Amer. Stat.*, vol. 41, pp. 340–341, 1987.

[44] R. A. Horn and C. R. Johnson, *Matrix Analysis*. London, U.K.: Cambridge Univ. Press, 1985.

Puneet Sharma received the B.Tech degree in mechanical engineering from the Indian Institute of Technology (IIT), Bombay, in 2001, and the M.S. degree in general engineering and the Ph.D. degree in systems and entrepreneurial engineering from the University of Illinois at Urbana-Champaign, in 2003 and 2008, respectively.

He is a Project Manager in the Image Analytics & Informatics Group, Siemens Corporate Research, Princeton, NJ. His research interests are in the area of computational modeling, simulation and optimization of large-scale systems.



Srinivasa M. Salapaka (M'03) was born in Andhra Pradesh, India, in 1973. He received the B.Tech. degree in mechanical engineering from the Indian Institute of Technology, Chennai, in 1995 and the M.S. and Ph.D. degrees in mechanical engineering from the University of California, Santa Barbara, in 1997 and 2002, respectively.

From 2002 to 2004, he was a Postdoctoral Associate in the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge. Since 2004, he has been a Faculty Member in Mechanical Science and Engineering, University of Illinois, Urbana-Champaign. His areas of current research interest include controls for nanotechnology, combinatorial resource allocation, and numerical analysis.

Dr. Salapaka received the National Science Foundation CAREER Award in 2005.

Carolyn L. Beck (SM'07) received the B.S. degree in electrical and computer engineering, from California State Polytechnic University, East San Gabriel Valley, in 1984, the M.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1985, and the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, in 1996.

From 1985 through 1989, she was a Research and Development Engineer for Hewlett-Packard, Santa Clara, CA. She is currently an Associate Professor at the University of Illinois at Urbana-Champaign, in the Department of Industrial and Enterprise Systems Engineering. Her primary research interests lie in the development of model reduction and control analysis theory, multiobjective optimization techniques, communications networks analysis, and bioengineering applications.

Dr. Beck received the ONR Young Investigator Award in 2001, and the NSF CAREER Award in 1998.