2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Entropy Based Algorithm for Combinatorial Optimization Problems with Mobile Sites and Resources

Puneet Sharma, Srinivasa Salapaka and Carolyn Beck

Abstract— **In this paper, we propose an algorithm for solving combinatorial resource allocation problems with mobile sites and resources. We consider the instantaneous coverage problem, which is formulated in a Maximum Entropy Principle (MEP) framework, and identify a metric for optimal coverage between mobile sites and resources. This metric, referred to as the *Free Energy* has been motivated by principles of statistical physics. The determination of resource locations and velocities is cast as a control problem to ensure that the time derivative of Free Energy function is always non-increasing. This guarantees coverage through the time horizon. The issues of scalability and localization are also addressed by taking into account inter-resource interaction levels.**

I. INTRODUCTION

In recent years, there has been a considerable amount of research on problems that address deployment of static or mobile resources that "cover" a set of sites in a given region. Most of these problems are intimately related to a class of combinatorial resource allocation problems that have been studied extensively in various formulations such as the minimum distortion problem in data compression [6], facility location assignments [2], optimal quadrature rules and discretization of partial differential equations [3], pattern recognition [18], drug discovery [15], neural networks [9], and clustering analysis [8]. In contrast, these problems are relatively recent in the control literature, having arisen in coarse quantization [4], [12], coverage control, mobile sensing networks, and motion coordination algorithms [1]. These areas, either directly or indirectly, bring together concepts from information theory and control theory. Although these problems focus on seemingly unrelated goals, they have a number of fundamental common attributes.

The aforementioned combinatorial optimization problems share the fundamental goal of aiming to determine an optimal partition of the underlying domain in which they are defined (e.g., a library of compounds for drug discovery, an unknown area of interest for coverage control), and an optimal assignment of values, or elements, from a finite resource set to each *cell* in the partition. Computationally, these problems are typically complex and time intensive if not intractable. For example, in the drug discovery problem, selecting 25 representative compounds from an array of 1000 compounds results in 4.76×10^{49} possibilities. This rules out any exhaustive search method over all partitions. The complexity of such problems is further complicated by their inherent non-convex nature, which makes it important to design algorithms that avoid local minima [7]. Since the static resource allocation problem is NP-hard, we cannot expect to find optimal solutions. As a result, the proposed algorithms are either based on heuristics, or solve a slightly modified version of the original problem.

The complexity of the resource allocation problems we consider is further compounded by the addition of dynamics to the sites. These mobile sites could be moving threat locations in a battlefield scenario, forest fires, unmanned vehicles, or swarms, depending on the context of the underlying problem. The task at hand is to design a velocity field for resources such that coverage is maintained throughout the time horizon. This class of problems has been studied as deployment and tracking problems [1], [5], where emphasis has been given to distributed implementations (i.e. under limited information). The emphasis on distributed algorithms is well justified since the underlying computational costs incurred for centralized schemes are impractical. Additionally, the centralized schemes are not practically viable for implementation in many application areas, due to the limited range of sensors. However, the distributed algorithms are prone to converge to one of the many local minima that typically riddle the coverage functions. As a result, the performance of these algorithms (i.e. coverage cost) is very sensitive to initial placement of the resources.

In this paper, we consider a coverage problem with mobile sites and resources. Our emphasis is on developing algorithms that are designed to avoid local minima and are insensitive to initial placement of the resources. An entropy term that quantifies the randomness in the choice of the initial resource allocations and corresponding partitions is introduced in the problem formulation in order to monitor the sensitivity of the solution to the initial allocation. This term and an associated averaging operation is used to smoothen the coverage function, and thus ensures that the algorithm successfully avoids the local minima, albeit at a higher computational expense than distributed algorithms. The relative importance of the coverage function with respect to the entropy term is controlled using a parameter, *temperature*, so named to emphasize the analogy to the annealing problem in statistical physics [13]. Consequently, an algorithm is developed which seeks to minimize an adjusted coverage function (termed *free energy*) that includes the coverage cost as well as the entropy term designed to avoid local minima. The probability distribution on the space of partitions for computing the entropy is determined using the maximum entropy principle (MEP) [10].

The authors are with the Coordinated Science Lab, University of Illinois at Urbana Champaign. Emails: psharma2, salapaka, beck3 @uiuc.edu. This work is partially supported by NSF ITR Collaborative Research grant ECS-0426831 and NSF ECS 0449310 CAR grant.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

The *free energy* is analogous to a control Lyapunov function where the velocity field on resources is sought to make the time derivative of the free energy non-positive. Computational complexity of the resulting algorithm is then addressed by exploiting the structure of this time derivative. This algorithm becomes more *local* (i.e. the resource locations becomes less sensitive to *far off* sites) as the temperature value is reduced. This feature is exploited to make this algorithm scalable and computationally efficient. Computational expense is included as a cost term in the MEP formulation, resulting in an algorithm that seeks minimization of a mix of coverage, computational cost and sensitivity to initial placement. Results are presented to demonstrate the ability of the algorithm to successfully avoid local minima. Results are also presented to show the progressively distributed implementation of the algorithm, and its comparison to the centralized scheme. As expected, such an implementation is computationally efficient, and can be used to address the issue of scalability associated with large datasets.

This paper is organized as follows. We present a mathematical formulation of the dynamic resource allocation problem in Section II. The solution for the instantaneous coverage problem is sought under the MEP framework. We then present the solution for the dynamic coverage problem in a control systems framework. The implementation of the algorithm is discussed in Section III. Simulation results are presented in Section IV. Finally we conclude the paper by revisiting the important results and identifying future goals.

II. DYNAMIC PROBLEM

Consider a problem where the task is to track the motion of a group of sites/objects in a given area. The motion of these sites is such that they may move in different clusters, change associations, split and rejoin clusters. Alternatively, this problem can also be posed as a coverage problem, where the aim is to successively identify representative objects in the area such that they "cover" the moving sites at all times. The number of these representative objects is far less than those of the moving sites. This class of problems is closely related to the problem of video segmentation often encountered in medical imaging, military tracking and reconnaissance, surveillance and other applications.

Let us consider an illustrative example where we have video data for movement of people in a given area, and our task is to identify each cluster and track it over time. This scenario is depicted in Fig. 1. It is assumed that the background does not change. Since video data is a time indexed series of still images, one obvious solution is to cluster the image data in each frame. Any of the standard clustering algorithms (for static data) can be used for this purpose. This is a very time intensive process and is usually not viable. In order to address this issue, we pose this as a coverage problem, where the optimization task is to identify representative locations (denoted by crosses) such that they *track* each cluster over time. A mathematical formulation for such a class of coverage problems is provided below. In the rest of the paper, we have used the term "mobile sites" to refer to the people movement and "mobile resources" to refer to the representative locations which identify each cluster.

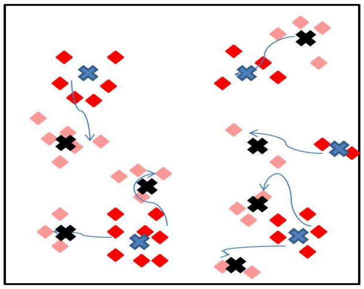

Fig. 1. Tracking movement of people in a given area. Red squares: initial position of people (sites), pink squares: final position. Blue crosses: initial location of resources, black crosses: final position of the resources. Splitting of some clusters has resulted in an increase in the number of resources.

A. Problem Formulation

Consider a domain consisting of mobile sites $x_i(t)$ with known velocities. Our goal is to determine the locations and velocities of mobile resources $y_j(t)$, such that they "cover" the mobile sites at any instant of time. For notational convenience, we will sometimes use x in place of $x(t)$, and time dependence will inherently be assumed, if not mentioned otherwise. We have a domain with N sites $x_i = [\xi_i \;\; \eta_i]^T \in \mathbf{R}^2$ and M resource locations $y_j = [\rho_j \;\; \omega_j]^T \in \mathbf{R}^2$, whose dynamics are given by

$$\dot{x} = \phi(x, r), \;\; x(0) = x_0 \qquad (1)$$

$$\dot{y} = u, \;\; y(0) = y_0 \qquad (2)$$

Our task is to determine the dynamics for the M mobile resources ($\dot{y} = u(t)$), and their initial locations (y_0), such that adequate coverage is maintained. This dynamic resource allocation problem (at time t) can be stated as:

Given a distribution $p(x)$ of the elements $x(t)$ in a space Ω, find the set of M resource locations $y_j(t)$ that solves the following minimization problem (at time t):

$$\min_{y_j,\; 1\leq j\leq M} \sum_{\Omega} p(x) \left\{ \min_{1\leq j\leq M} d(x, y_j) \right\} dx. \qquad (3)$$

Here $d(x, y_j)$ represents an appropriate *distance* metric between the resource location y_j and the element x. The most natural choice of the distance metric is the Euclidean distance. Modified versions of this metric can also be used in specific scenarios. One such metric is $d(x_i, y_j) = \|x_i - y_j\|^2 + \|\phi_i - u_j\|^2$, where velocity as well as location terms, are included. Minimizing (3) is akin to finding a velocity field for resources such that coverage condition is satisfied at time t. Alternatively, this problem can also be formulated as finding an *optimal* partition of the descriptor space Ω at time t into M cells R_j and assigning to each cell R_j a resource location y_j which minimizes $\sum_j \sum_{R_j} d(x, y_j) p(x) dx$. Solving the above optimization problem at a fixed time (t) is equivalent to solving a static resource location problem. Deterministic annealing (DA) algorithm [13] has been successfully implemented in a host of areas to tackle such problems. The DA algorithm can be viewed as a modification of Lloyd's algorithm [11], in which the initial step consists of randomly choosing resource locations and then successively iterating between the steps: (1) forming Voronoi partitions, and (2) moving the resource locations to the respective centroids of the cells till the sequence of resource locations converge. Note that the Lloyd's algorithm solution depends substantially on the initial allocation, as in the successive iterations, the locations are influenced only by 'near' sites in the domain and are virtually independent of 'far' sites. As a result, the solution 'typically' gets stuck to local minima. As in the DA algorithm, we can diminish this local influence of domain elements by allowing each $x \in \Omega$ to be associated with every resource location y_j through a weight $p(y_j|x)$. Introducing these weights eliminates the hard partitions of Lloyd's Algorithm. Under this formulation, we have a distortion term

$$D = \sum_i p(x_i) \sum_j d(x_i, y_j) p(y_j|x_i), \text{s.t.} \ \sum_j p(y_j|x_i) = 1$$

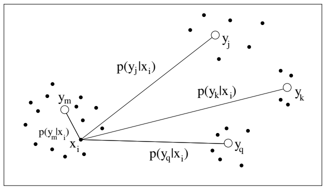

Fig. 2. Dimnishing local influence $p(y_m|x_i) >> p(y_j|x_i)$

Note that the instantaneous weighting term (at time t) $p(y_j|x_i)$ can alternatively be viewed as a probability of association between the mobile site x_i and the mobile resource y_j. This enables us to address the problem of determining these weighting parameters under MEP.

The MEP deals with ascribing a probability mass function for a vector valued random variable such that given constraints are satisfied. In the MEP framework, these constraints are such that the expected values of a given set of functions (of the random variable) are equal to known constants. The MEP states that the probabilities should be such that they maximize the Shannon entropy [14] and at the same time satisfy the expected value constraints.

Consider the space of partitions $\mathcal{Q}$ of Ω. For any instance $q = \{\mathcal{Y}, \mathcal{P}\} \in \mathcal{Q}$, with the partition $\mathcal{P} = \{\Omega_j\}$ and the resource locations $\mathcal{Y} = \{y_j\}$, the coverage cost in (3) is

$$D(q) = \sum_i \sum_j \chi_{ij} d(x_i, y_j), \tag{4}$$

where χ_{ij} is the indicator function, i.e. $\chi_{ij} = 1, \text{if } x_i \in \Omega_j$ and 0 otherwise. Now consider a probability distribution $P(q)$ on the space of partitions $\mathcal{Q}$. Under this distribution, the average coverage cost is given by $\sum P(q)D(q)$, where the summation is over all possible elements in the set $\mathcal{Q}$ (i.e. all possible partitions). Since we have no prior information on the distribution over $\mathcal{Q}$, we use MEP to determine this probability distribution. Such a distribution thus maximizes the entropy under the constraint that it achieves the average cost, and is given by a Gibbs distribution over partitions

$$P(q) = \frac{e^{-\beta D(q)}}{\sum_{q' \in \mathcal{Q}} e^{-\beta D(q')}},$$

where β can be determined from the value of the average cost [10]. The most probable set of resource locations can be determined by maximizing the marginal probability $P(\mathcal{Y}) = \sum_{\mathcal{P}} P(\mathcal{Y}, \mathcal{P})$. In order to compute it, we note from (4) that

$$\sum_{\mathcal{P}} e^{-\beta D(\mathcal{Y}, \mathcal{P})} = \prod_x \sum_k e^{-\beta d(x, y_k)} := Z, \tag{5}$$

where Z is the *partition function* (as it has parallels in statistical physics [13]). Next, we define the *free energy*,

$$F := -\frac{1}{\beta} \log Z = -\frac{1}{\beta} \sum_i \log \sum_k e^{-\beta d(x_i, y_k)}. \tag{6}$$

Now, the marginal probability $P(\mathcal{Y})$ can be rewritten as

$$P(\mathcal{Y}) = \frac{Z}{\sum_{\mathcal{Y}'} Z} = \frac{e^{-\beta F}}{\sum_{\mathcal{Y}'} e^{-\beta F}}. \tag{7}$$

Thus the most probable resource location set $\mathcal{Y}$ is the one that minimizes the Free Energy We now consider the entropy associated with a specific q, given by

$$H = -\sum_{\Omega} p(x) \sum_j p(y_j|x) \log p(y_j|x) dx, \tag{8}$$

which measures the randomness of the distribution of the associated weights $p(y_j|x)$. This entropy term is the largest when the distribution of weights over each resource location is identical, i.e. ($p(y_j|x) = 1/M$) for each x, i.e., when all x have the same influence over every resource location.

B. Solution

Using the MEP, we have demonstrated that minimizing the instantaneous Free Energy can be used as a criteria for solving the coverage problem of mobile sites and resources. Thus, maximizing the instantaneous coverage can be viewed as solving the following optimization problem

$$\min_{y_j} \min_{p(y_j|x)} F := D - T_k H \tag{9}$$

at the kth iteration where $T_k := \frac{1}{\beta_k}$ is the *temperature* parameter which tends to zero as k tends to infinity. Clearly for large values of T_k, we mainly attempt to maximize the entropy. As T_k is lowered we trade entropy for the reduction in distortion, and as T_k approaches zero, we minimize D directly to obtain a *hard* solution. Minimizing F with respect to $p(y_j|x)$ is straightforward and gives a *Gibbs* distribution

$$p(y_j|x) \ = \ \frac{e^{-d(x, y_j)/T}}{\sum_i e^{-d(x, y_i)/T}}. \tag{10}$$

The corresponding minimum of F (denoted by $\hat{F}$) is obtained by substituting for $p(y_j|x)$ from (10). Next we minimize $\hat{F}$ with respect to the resource locations $\{y_j\}$, by setting the corresponding gradients equal to zero i.e., ($\frac{\partial \hat{F}}{\partial y_j} = 0$), giving implicit equations for the locations,

$$y_j \ = \ \sum_{\Omega} p(x|y_j) x dx \quad 1 \leq j \leq M, \tag{11}$$

where $p(x|y_j)$ denotes the posterior probability calculated using Bayes's rule. The above equations clearly convey the *centroid* aspect of the solution.

Thus the instantaneous coverage problem can be addressed by minimizing the Free Energy with respect to resource locations (y_j) starting at high values of temperature and tracking its minimum while lowering the temperature value steadily. Solving this instantaneous problem at each point in time should give us the solution for the coverage problem over a finite horizon. However, in most of the applications, this frame-by-frame approach is not viable as it is extremely time-consuming. This is due to the fact that it requires us to solve multiple MEP problems at each instant of time.

The MEP based algorithm for solving the static problem has iterations dictated by the temperature values as per a *cooling law*. An exponential cooling rate (i.e. decreasing temperature exponentially) is often used for implementing the algorithm [13]. Now with the added dynamics of sites and resources, choosing a cooling law becomes an additional design issue. For example, we can design the rate of cooling to be much faster than the given dynamics of the sites, to have an algorithm which is similar to the frame-by-frame approach. One of the key features of the MEP based algorithm is its *phase transition* property. Even in the static case, there are critical values of temperature at which the resource locations split and between these critical values, a change in temperature does not have an appreciable effect on the resource locations. The fact that the resource locations do not continuously depend on temperature, but exhibit a phase transition property enables us to design an efficient algorithm that includes dynamics of the sites and resources. Under these added dynamics, we can explore similar phase transition behavior at critical temperatures and design cooling laws that are much faster than the frame-by-frame approach. Phase transition effectively takes into account the dynamics due to the given velocity field and the inherent dynamics induced by the change in temperature value, and successively identifies and tracks the natural clusters in the site domain.

In order to circumvent the problems faced by frame-by-frame approach, different optimization criterion (involving Free Energy) can be considered. In an optimal control setting, the coverage goal is defined as: $\min_u \int_0^\infty F(x,u,\phi,u,t)dt$. Another criterion is to determine u such that $\frac{dF(x,u,\phi,u,t)}{dt} < 0$, which provides continuously improved coverage of the clusters over time. We used the latter criterion as the former is effectively intractable.

We have formulated the problem in a control systems framework, where the task is to determine the control (u) such that Free Energy is non-increasing. Consider the sites-resources system defined by (1)-(2) again. Free Energy for this sites-resources combination is given by

$$F = -\frac{1}{\beta}\sum_i p(x_i)\log\sum_j e^{-\beta d(x_i,y_j)}$$

Differentiating, we get the partial derivatives

$$F_x{}^T = 2\left[(I_2\otimes P_1)x - (I_2\otimes P_{12})y\right] \qquad (12)$$

$$F_y{}^T = 2\left[(I_2\otimes P_2)y - (I_2\otimes P_{12}^T)x\right] \qquad (13)$$

where $P_1 = \mathrm{diag}(p(x_i)), P_2 = \mathrm{diag}(p(y_j)), P_{12} = [p(x_i,y_j)]$.

The time derivative of Free Energy is given by

$$\dot{F} = F_x\dot{x} + F_y\dot{y} = \zeta^T\Gamma\Psi, \text{ where}$$

$$\zeta = \begin{bmatrix} x \\ y \end{bmatrix}, \Psi = \begin{bmatrix} \phi \\ \nu \end{bmatrix}, \Gamma = \begin{bmatrix} I_2\otimes P_1 & -I_2\otimes P_{12} \\ -I_2\otimes P_{12}^T & I_2\otimes P_2 \end{bmatrix}$$

Introducing new transformed variables gives us: $\bar{y} = y-(I_2\otimes Q_2{}^T)x, \bar{u} = (I_2\otimes P_2)(u-(I_2\otimes Q_2^T)\phi), \bar{\phi} = (I_2\otimes P_1)\phi$ where $Q_1 = [p(y_j|x_i)] \in \mathbf{R}^{N\times M}$ and $Q_2 = [p(x_i|y_j)] \in \mathbf{R}^{M\times N}$ Thus we have the relation $P_1Q_1 = Q_2P_2 = P_{12}$. In the transformed coordinates, $\frac{dF}{dt}$ is given by

$$\frac{dF}{dt} = \begin{bmatrix} x \\ \bar{y} \end{bmatrix}^T \begin{bmatrix} I_2\otimes(I-Q_2Q_1^TP_1) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{\phi} \\ \bar{u} \end{bmatrix}$$

$= x^TR\bar{\phi}+\bar{y}^T\bar{u} = x^TS\phi+\bar{y}^T\bar{u}$, where $R = (I-Q_2Q_1^TP_1)$

Any $\bar{u}$ that will make $\dot{F} < 0$ will provide us with consistent coverage at all times. For the case when $\bar{y}^T\bar{u} \neq 0$, we use Sontag's formula [17] to prescribe one such velocity field for the mobile resources,

$$u = (I_2\otimes Q_2^T)\phi-(I_2\otimes P_2)^{-1}\left(\frac{S\phi+\sqrt{(S\phi)^TS\phi+\bar{y}\bar{y}^T}}{\bar{y}\bar{y}^T}\right)\bar{y}^T \qquad (14)$$

Whenever the term $\bar{y}^T\bar{u} = 0$, we lose "controllability", and thus no control u exits that guarantees $\dot{F} < 0$. At this point, we change the temperature value and recompute the association probabilities and resource locations. With these new values, we verify whether the phase transition condition is satisfied; this is discussed further in the next paragraph.

C. Phase Transitions

For the implementation of the algorithm, we start at a very high value of temperature (T) and reduce it as the algorithm proceeds. To begin, we have just one resource location at the centroid of the dataset, i.e. $y = (I_2\otimes Q_2{}^T)x$. As the temperature is reduced, the system undergoes a series of phase transitions [13], where as before *phase transition* refers to the splitting of resource locations and the identification of natural clusters in the data. Successive phase transitions identify finer clusters.

At the instant when a split occurs, the Hessian of minimized F loses positive definiteness. To analyze this temperature value, we place coincident resources at a single location y_k and perturb it by an amount $\epsilon\Psi_k$. The necessary condition for optimality is $\frac{\partial}{\partial\epsilon}F_\epsilon|_{\epsilon=0} = 0$. We also need to ensure that the second partial derivative is positive ($\frac{\partial^2F_\epsilon}{\partial\epsilon^2} > 0$). This happens when twice the maximum eigenvalue of the posterior covariance matrix, defined by $C_{x|y_j} = \sum_i p(x_i|y_j)(x_i-y_j)(x_i-y_j)^T$ becomes greater than the temperature value. The temperature at which the split occurs is called the critical temperature (T_c), and is given by $T_c = 2\lambda_{max}[C_{x|y_j}]$.

D. Scalability

One of the major problems with combinatorial optimization algorithms is that of scalability, i.e. the number of computations scales up combinatorially with an increase in the amount of data. This property is a deterrent to using this algorithm for large (complex) problems. One of the features of the MEP based algorithm is that it becomes more local (that is the computation of a resource location becomes less sensitive to far off terms) as the temperature variable is successively reduced. We have exploited this feature to incorporate the distributed aspect in the problem formulation, thereby improving the computational efficiency of the algorithm. It is clear that as temperature is decreased, the association probability of a mobile site $x(t)$ with distant resources decreases exponentially to zero. Thus a very small error will be incurred if we do not consider these negligible interactions. This tradeoff between error in resource locations and computation time has been characterized.

In [16], we present a scalable resource allocation problem based on the MEP. We use a similar framework to address the distributed computation aspect herein. As is discussed in [16], the association probabilities can be expressed as a modified Gibbs-type distribution,

$$p(y_j|x_i) = \frac{e^{-\beta_1 d_{ij} - \beta_2 d_{ij} M_{ij} - \beta_3 N_j^2}}{Z(\beta_1, \beta_2, \beta_3; \sigma_j)} \tag{15}$$

where the partition function Z is defined by

$$Z_i(\beta_1, \beta_2, \beta_3; \sigma_j) = \sum_j e^{-\beta_1 d_{ij} - \beta_2 d_{ij} M_{ij} - \beta_3 N_j^2} \tag{16}$$

Next, the resource locations are computed as follows:

$$y_j = \frac{\sum_i p_{ij}\left\{W_{ij}x_i - \frac{2\beta_3 N_j}{\sigma_j}\left[\sum_k e^{\frac{-d_{kj}}{\sigma_j}} x_k\right]\right\}}{\sum_i p_{ij}\left\{W_{ij} - \frac{2\beta_3}{\sigma_j} N_j^2\right\}} \tag{17}$$

The phase transition phenomenon is further compounded by the fact that we have three different temperature variables for incorporating additional constraints. We have explicitly computed the condition for splitting [16], taking into account the change of all three temperatures. Simulation results for scalable algorithms are included in Section IV

III. IMPLEMENTATION

We have summarized below the steps that the algorithm traces in order to solve the coverage problem.

- 0: Initialize the algorithm at $T = \infty$ (i.e. $\beta = 0$).
- 1: Determine the resource locations (11) together with the association probabilities (10).
- 2: Simulate the site movement under (1) and determine the resource velocities (14) to maintain coverage.
- 3: If $\bar{y}^T \bar{u} = 0$, decrease the temperature, go to Step 1.
- 4: If the phase transition condition is satisfied, split the resources, re-distribute their weights, go to Step 1.
- 5: Stop if the stopping criterion (e.g. number of resources, time horizon) is met, otherwise go to Step 1.

IV. SIMULATION RESULTS

In this section, we have provided simulation results for both scalable and non-scalable version of the algorithm.

A. Basic Algorithm

For the purpose of simulation, we chose a scenario with 10 mobile sites. Random velocities were chosen for each of the mobile sites, with the constraint that natural clusters be preserved. To begin, the dataset had four natural clusters (Fig. 3). The blue colored rhombuses show the initial location of the sites. The algorithm begins by placing one resource at the centroid of the site positions (at $t = 0$) (shown by red cross). As the sites move (direction shown by arrows), the algorithm progressively updates the association probabilities and resource locations, and determines the control value (i.e. the resource velocities). We have shown the movement of sites by magenta colored lines. Temperature is steadily decreased until phase transition occurs, whereby a single resource splits into two distinct resources (with lower weights). The 2 blue crosses denote the resource locations after the first split. Similarly, the 3 green crosses and the 4 black crosses denote the location of the mobile resources after further splits. We have not shown the resource locations computed at every time step, so as not to obscure the main stages of progression of the algorithm. The resource locations maintain coverage at all times and track the natural clusters. The velocity field is designed such that coverage metric is always non-increasing.

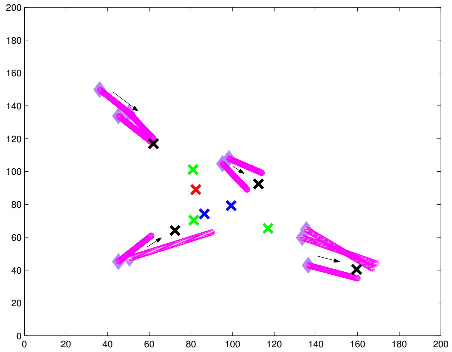

Fig. 3. Snapshots showing locations of mobile sites and resources

As expected, the MEP based algorithm is much faster than the frame-by-frame approach. The MEP based algorithm took 424 seconds to solve the above problem over the complete time horizon (on an Intel 1.5 GHz Centrino processor), while the frame-by-frame approach had solved only 25% of the time horizon in the same amount of computation time.

B. Scalable Algorithm

In order to demonstrate the scalable version of the algorithm, we chose a scenario with 16 mobile resources. Random velocities were chosen for each of the mobile sites,with the constraint that some of the clusters did not interact with one another. At $t = 0$, the dataset had four natural clusters, as seen in Fig. 4. The scalable algorithm begins by placing one resource at the centroid of the site positions (shown by the red cross). As the sites move (as shown by the arrows), the algorithm progressively updates the association probabilities (15) and resource locations (17), and determines the control value (i.e. the resource velocities). After the first split, two distinct resource locations (shown by blue crosses) are identified and the parameter σ_j prescribes two separated regions (roughly shown by the green dotted lines). Further computations are truncated in these regions, and thus only local influence of the mobile sites is accounted for. Computations for Cluster 1 and Cluster 2 do not take into account the interactions from Cluster 3 and Cluster 4 (and vice-versa). As temperature is decreased further, resource locations split, denoted successively by black and green crosses. At this instance, the parameter σ identifies three invariant regions to be used for truncated computation (shown by black dashed lines). The resource locations maintain coverage at all times and track the natural clusters formed by the mobile sites.

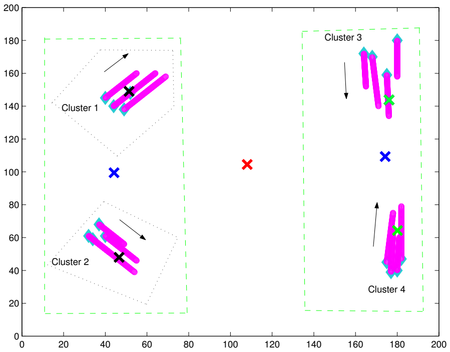

Fig. 4. Snapshots showing locations of mobile sites and resources

The scalable algorithm is computationally more efficient than the basic version. As shown in Table 1, it takes just one-fourth of the time taken by the basic algorithm to solve the coverage problem on the same dataset. For the purpose of this comparison, the coverage cost was determined at $t = t_{final}$.

Algorithm	Coverage Cost	Computation Time (sec)
Basic Algorithm	181.3	784
Scalable Algorithm	194.1	208

TABLE I

COMPARISON OF THE COMPUTATION TIMES

V. ANALYSIS AND DISCUSSION

We have successfully used the MEP framework to define a coverage metric for the mobile sites-resources problem. One of the main features of the MEP is that it provides a solution that is based only on the available information (i.e. least biased) and not on any other factors. This 'fairness' criterion is exemplified in the Gibbs distribution for the association probabilities. Problems in statistical physics often encounter such distributions.

The complexity of the coverage problem can be gauged from the fact that even the static resource allocation problem is a NP-hard problem and thus we cannot expect to achieve optimal solutions that minimize the coverage cost. Hence the algorithms are either based on heuristics or solve a modified cost function. Our algorithm is closely aligned with principles of statistical physics and exploits those inherent properties to successfully avoid locally optimal solutions. We also exploit the fact that the computations become successively "local", thereby making the algorithm amenable for solving problems on large datasets. In order to implement the localized version of the algorithm, we add some computational complexity (by introducing additional constraints), which diminishes as the algorithm proceeds. This addition is minuscule when compared to the computational efficiency extracted by the distributed implementation of the algorithm.

As was discussed earlier, the distance function and the coverage metric can be suitably modified to account for additional constraints or address specific scenarios. During some simulations, we observed that when the multiple natural clusters tended to move towards one another, their respective resources also began to converge towards each other. This natural coalescence of resources indicates that a 'reverse phase transition' phenomenon is naturally occurring in the algorithm. We are presently investigating this further and trying to devise rules for fusing multiple resources in such a scenario. Additionally, we are also working on designing estimators for determining the velocity of the mobile sites.

REFERENCES

[1] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage Control for Mobile Sensing Networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

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

[3] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review*, 41(4):637–676, December 1999.

[4] N. Elia and S. Mitter. Stabilization of Linear Systems with Limited Information. *IEEE Transactions on Automatic Control*, 46(9):1384–1400, September 2001.

[5] E. Frazzoli and F. Bullo. Decentralized algorithms for vehicle routing in a stochastic time-varying environment. *IEEE Conference on Decision and Control*, pages 3357– 3363, 2004.

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

[7] R. Gray and E.D. Karnin. Multiple local minima in vector quantizers. *IEEE Transactions on Information Theory*, IT-28:256–361, 1982.

[8] J.A. Harigan. *Clustering Algorithms*. Wiley, New York, 1975.

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

[10] E. T. Jaynes. *Probability Theory - The Logic of Science*. Cambridge University Press, 2003.

[11] S.P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[12] S.K. Mitter. Control with limited information. Plenary lecture at International Symposium on Information Theory, Sorento, Italy, 2000.

[13] K. Rose. Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems. *Proceedings of the IEEE*, 86(11):2210–39, November 1998.

[14] C. E. Shannon and W. Weaver. The mathematical theory of communication. *Univ. of Illinois Press, Urbana IL.*

[15] P. Sharma, S. Salapaka, and C. Beck. A deterministic annealing approach to combinatorial library design for drug discovery. *Proc. of American Control Conference*, pages 979–984, 2005.

[16] P. Sharma, S. Salapaka, and C. Beck. A maximum entropy based scalable algorithm for resource allocation problems. *Proc. of American Control Conference*, pages 516–521, 2007.

[17] E. D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *System and Control Letters*, 13(2):117–123, 1989.

[18] C.W. Therrien. *Decision, Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*, volume 14. Wiley, New York, 1st edition, 1989.