Locational Optimization Problems with Constraints on Resources

S. Salapaka[1] and A. Khalak[2]

Abstract

This paper discusses a clustering approach to locational optimization in the context of UAV mission planning. In particular, the Deterministic Annealing (DA) algorithm from the data compression literature is adapted to address such problems, which bears a strong analogy to the statistical physics formulation of the annealing process (i.e. material transformation under a decreasing temperature schedule). The mission planning domain motivates several extensions to DA to handle the case of heterogeneous UAVs, multiple resource types, and fungible and non-fungible resource types. These extensions introduce constraints on the basic optimization problem. Algorithmically, these are addressed by modifications to the free energy of the DA algorithm. An analysis of the algorithm shows that the iterations at a given temperature are of the form of a Decent Method, which motivates scaling principles which tend to accelerate convergence.

Introduction

Recently there has been a growing interest in the seemingly unrelated areas of coarse quantization [1, 2], coverage control, mobile sensing networks, and motion coordination algorithms [3]. These problems each with different and unrelated goals, in fact have some fundamental common attributes. All these areas, either directly or not, bring together the concepts from information theory and control theory. For example, [1] studies the aspect of control with minimum information while [3] studies the coverage control problem for mobile sensing networks with a dynamically changing environment with a distributed communication and computation architecture. Another striking similarity in these areas is that after disregarding the details, they aim to solve the same optimization problem - in fact, they try to obtain; (1) an optimal partition of the underlying domain, and (2) an optimal assignment of values from a finite set to each cell of the partition. The differences in these problems come from having different conditions of optimality and constraints. For example the coarse quantization problem consists of obtaining a partition of the state space and allocating a control value to each cell in such a way to obtain the *coarsest* such partition while maintaining the stability of the underlying system. Similarly the coverage control problem consists of obtaining an optimal placement and tuning of sensors via *optimal* partitioning of the underlying space.

Even though these formulations are relatively recent in the control theory, these optimal partitioning-assignment (also referred as *locational optimization*) problems come up in various forms and are studied in different areas such as minimum distortion problem in data compression [4], facility location problems [5], optimal quadrature rules and discretization of partial differential equations [6], pattern recognition [7], neural networks [8], and clustering analysis [9]. These problems are non convex and computationally complex. It has been well documented (e.g. [10]) that most of them suffer from poor local minima that riddle the cost surface. A variety of heuristic approaches have been proposed to tackle this difficulty, and they range from repeated optimization with different initialization, and heuristics to good initialization, to heuristic rules for cluster splits and merges. A known technique for non convex optimization that capitalizes on the analogy of annealing process in physical chemistry is stochastic relaxation or simulated annealing [11]. It was shown that this iterative method would achieve global minimum, but with an annealing rate so slow that this algorithm is not realistic in many applications. In this paper, we present the Deterministic Annealing algorithm developed in the data compression literature [12]. This algorithm offers two important features: (1) ability to avoid many poor local optima and (2) has a relatively faster convergence rate. It formulates an effective energy function parameterized by a (pseudo) temperature variable and this function is deterministically optimized at successively reduced temperatures.

In this paper, we introduce a set of problems that come under the class of locational optimization problems. They arise in the prepositioning problem of Unmanned Air Vehicles (UAVs) and resources for anticipated response. More precisely, we seek to determine the best response locations of UAVs and the appropriate allocation

[1]svasu@mit.edu, Massachusetts Institute of Technology, MA 02139.
[2]akhalak@alphatech.com, Alphatech Inc., Burlington, MA 01803

of resources to them when the probability distribution of the activities is given. The allocation of resources are further subjected to capacity constraints on UAVs and resources. These problems are non convex and computationally complex. In this paper, we describe the modifications that we have made to adapt the DA algorithm to these problems in order to accommodate the special capacity constraints.

1 Problem Statements

The problems discussed in this paper are motivated from prepositioning of UAVs equipped with various resources in anticipation of a variety of future activities. These problems have the objective of obtaining locations $\{r_j\}$, $1 \leq j \leq M$ to position M UAVs in a domain $\mathcal{D}$ to minimize the expected squared distance of the locations of anticipated future activities to their nearest UAV. The anticipated activities follow the probability distribution, $p(x)$, where $x \in \mathcal{D}$.This results in the following optimization problem:

P1. No constraints:

$$\min_{r_j,\ 1\leq j\leq M} \int_{\mathcal{D}} p(x) \left\{ \min_{1\leq j\leq M} \|x - r_j\|^2 \right\} dx.$$

This formulation is the one that the DA algorithm directly solves. We impose constraints on UAVs and resources to model the different scenarios that come up in the prepositioning problem which result in the following formulations:

P2. Capacity Constraints: In this scenario, the number of resources that jth UAV $(1 \leq j \leq M)$ can carry is specified. We denote the capacity of jth UAV by W_j. If we denote the number of resources in the jth UAV by λ_j, we have the following constrained minimization problem:

$$\min_{r_j,\ 1\leq j\leq M} \int_{\mathcal{D}} p(x) \left\{ \min_{1\leq j\leq M} \|x - r_j\|^2 \right\} dx.$$

s.t.

$$\lambda_j = W_j \quad 1 \leq j \leq M.$$

Note that the constraints here seem to be independent of the cost function. This is not so as λ_j can be interpreted as the number of resources at the location r_j. A larger value of λ_j implies that the activities near r_j require more resources. This parameter is more naturally incorporated in the cost function in the modified algorithm that we present in Section 3.

P3. Interchangeable Resource Types: In this scenario, there are p types of resources where the effectiveness of nth type resource is denoted by μ_n. Also the number of each type of resources is given - the total number of nth type is denoted by ν_n. If we represent the number of resources of type n in the jth UAV by λ_{jn}, then we have the following optimization problem to be solved:

$$\min_{r_j,\ 1\leq j\leq M} \int_{\mathcal{D}} p(x) \left\{ \min_{1\leq j\leq M} \|x - r_j\|^2 \right\} dx.$$

s.t.

$$\sum_j \lambda_{jn} = \nu_n \quad 1 \leq n \leq p.$$

P4. Multiple Resource Types: These constraints relate to the scenario in which there are p types of resources. Further, the resource requirements for a given activity can not be exchanged for different resource types. The probability of the activity corresponding to location x^n needing resources of type n is given by $p_n(x^n)$.

The capacity of jth UAV with respect to nth type resource is given by W_{jn}. The resulting optimization problem

$$D = \sum_n \int_{\mathcal{D}} p_n(x^n) \sum_{j=1}^{M} \|x^n - r_j\|^2 p(r_j|x^n) dx^n.$$

st

$$\lambda_{jn} = W_{jn} \quad 1 \leq j \leq M, \ 1 \leq n \leq p,$$

where λ_{jn} denotes the number of nth type resources in the jth UAV.

We look at the cost function in P1-P3 in another way - we view it as the problem of associating each element $x \in \mathcal{D}$ with a resource r_j such that the cost function is minimized or equivalently to partition the domain $\mathcal{D}$ into M cells R_j and allocate a representative resource to each cell such that the expected distance of the point $x \in R_j \subset \mathcal{D}$ from the corresponding resource location r_j is minimized, i.e.

$$\min_{r_j, R_j \in \mathcal{R}} \sum_j \int_{R_j} p(x) \|x - r_j\|^2 dx,$$

where $\mathcal{R}$ denotes the partition $\{R_j\}$. So we have shown that these problems fall in the category of locational optimization problems.

The locational optimization problems are non convex and computationally complex. For example in P1, the optimal allocation of 20 resources in a domain of 30 points and a given probability mass function would require search over 30 million partitions! This renders searches over all partitions practically impossible. Accordingly, there have been many algorithms that use some heuristics to come up with reasonable solutions to these problems. In this paper, we describe an iterative algorithm described in [13] which has been observed to give 'good' solutions in simulations even though it does not guarantee the achievement of global minimum.

2 Deterministic Annealing Algorithm

This algorithm can be viewed as a modification of another algorithm called Lloyd's algorithm [14, 4]. The Lloyd's algorithm is an iterative method which ensures that at each iteration, the partition of domain and the resource locations satisfy the following two necessary (but not necessarily sufficient) properties that the solution has:

1. Nearest Neighbor condition (Voronoi partitions): The partition of the domain is such that each element in the domain is associated to the nearest resource location.

2. Centroid condition: The resource locations are such that the location r_j is in centroid of the jth cell R_j.

In this 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 till the sequence of resource locations converge. It should be noted that the solution depends substantially on the initial allocation of resource locations as in the successive iterations the locations are influenced only by 'near' points of the domain and are virtually independent of 'far' points. As a result the solutions from this algorithm 'typically' get stuck to local minima.

The DA algorithm [13, 15] does away with this local influence of domain elements by allowing each element $x \in \mathcal{D}$ to be associated with every resource location r_j through a weighting parameter $p(r_j|x)$. Thus this algorithm does away with the hard partitions of the Lloyd's Algorithm. The DA formulation includes a modified distortion term

$$D = \int_{\mathcal{D}} p(x) \sum_j \|x - r_j\|^2 p(r_j|x) dx,$$

which is similar to the cost function in P1. It also includes an entropy term

$$H = -\int_{\mathcal{D}} p(x) \sum_j p(r_j|x) \log p(r_j|x) dx,$$

which measures the randomness of distribution of the associated weights. This entropy is the highest when the distribution of weights over each resource is the same ($p(r_j|x) = 1/M$) for each x, i.e., when all x have the same influence over every resource location. This algorithm solves the following optimization problem

$$\min_{r_j} \min_{p(r_j|x)} \underbrace{D - T_k H}_{:=F}$$

at the kth iteration where T_k is a parameter called *temperature* which tends to zero as k tends to infinity. The cost function F is called *Free Energy* as this formulation has a close parallel in statistical physics [16]. 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 (non random) solution. Minimizing the Free Energy term F with respect to the association probabilities $p(r_j|x)$ is straightforward and gives the *Gibbs* distribution

$$p(r_j|x) = \frac{e^{-\|x-r_j\|^2/T}}{Z}, \quad \text{where} \quad Z := \sum_i e^{-\|x-r_i\|^2/T} \tag{1}$$

is called the *partition function*. The corresponding minimum of F is obtained by substituting for $p(r_j|x)$ using Equation 1, $\hat{F} = -T \int_{\mathcal{D}} p(x) \log Z$. To minimize $\hat{F}$ with respect to the resource locations $\{r_j\}$, we set the corresponding gradients to zero ($\frac{\partial \hat{F}}{\partial r_j} = 0$) yields the corresponding implicit equations for the resource locations

$$r_j = \int_{\mathcal{D}} p(x|r_j) x dx \quad 1 \leq j \leq M \quad \text{where} \quad p(x|r_j) = \frac{p(x)p(r_j|x)}{\int_{\mathcal{D}} p(x')p(r_j|x')dx'}. \tag{2}$$

Note that $p(x|r_j)$ denotes the posterior probability calculated using Bayes's rule and the above equations clearly conveys the 'centroid' aspect of the solution.

The DA algorithm consists of minimizing $\hat{F}$ with respect to $\{r_j\}$ starting at high values of T_k and tracking its minimum while lowering the values of T_k. The steps at each k are

1. fix $\{r_j\}$ and use Equation 1 to compute the new weights $\{p(r_j|x)\}$.
2. fix $\{p(r_j|x)\}$ and Equation 2 to compute the resource locations $\{r_j\}$.

3 Adaptation of the DA algorithm to Capacity Constraint problems

It should be noted that the resources (UAVs and resources) in the formulation of the DA algorithm are not distinguishable. However, the capacity constraints in P2-P4 distinguish one resource from another, i.e. UAVs are of different capacities (can hold different amounts of resources) or resources themselves are of different types or sizes. We now present the modifications that we have made to the DA algorithm to accommodate each constraint:

P2. Capacity Constraints: We reformulate the algorithm by associating each resource location r_j to a multiplicity of resources λ_j. Thus λ_j can be interpreted as the number of resources deployed at the same location r_j or alternatively, it can be thought of as the number of resources required by the UAV at location r_j. Thus the constraint in P1 is easily captured by $\lambda_j = W_j$. This formulation differs from the DA algorithm in the fact that now we are considering multiple units of resources at each location while the previously we were associating each location only with one resource. Taking this multiplicity into account, the partition function in Equation 1 can be rewritten as $Z := \sum_i \lambda_i e^{-\|x-r_i\|^2/T}$. Thus the original DA algorithm can be modified to include the weighting of resource locations just by reinterpreting the partition function and thereby modifying it. The corresponding Gibbs distribution and the Free Energy is given by

$$p(r_j|x) \quad = \quad \frac{\lambda_j e^{-\|x-r_j\|^2/T}}{Z} \quad \text{and} \quad \hat{F} = -T\int_{\mathcal{D}} p(x)\log\sum_i \lambda_i e^{-\|x-r_i\|^2/T}. \tag{3}$$

Therefore at each iteration k we need to solve the Lagrangian: $F' = \hat{F} + \sum_{j=1}^{M} q_j(\lambda_j - W_j)$, where q_j $1 \leq j \leq M$ represent the Lagrange multipliers corresponding to the constraints $\lambda_j = W_j$. The optimal set of resource locations $\{r_j\}$ satisfy $\frac{\partial F'}{\partial r_j} = 0$ and since the constraint is independent of these locations, we obtain the same implicit equation for them as before:

$$r_j \quad = \quad \int_{\mathcal{D}} p(x|r_j)x dx \quad 1 \leq j \leq M \tag{4}$$

except that $p(r_j|x)$ is given by the modified Gibbs distribution in Equation 3. Plugging the constraint in this solution (equivalently setting $\frac{\partial F'}{\partial q_j} = 0$) we obtain the same equation for the resource location as in Equation 4 where $p(x|r_j)$ is given by Bayes's rule in Equation 2 where $p(r_j|x) = \frac{\lambda_j e^{-\|x-r_j\|^2/T_k}}{\sum_i \lambda_i e^{-\|x-r_i\|^2/T_k}}$.

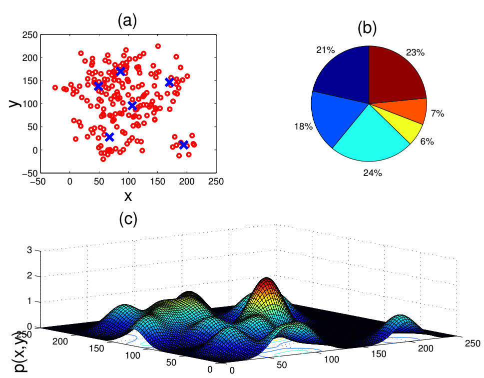

Figure 1: (a) The threat locations (circles) and the UAV locations (crosses). (b) The capacities of different UAVs (c) The probability density function $p(x, y)$

Figure 1 shows a scenario of 200 possible threat (activity) locations (represented by circles) in a two dimensional rectangular domain. The probability distribution $p(x, y)$ of a location (x, y) being a threat is shown in (c). There are 6 UAVs to 'cover' the domain. Their relative capacities or strengths are shown in pie chart (b). We ran the algorithm and the locations that it picked for UAVs are shown by the crosses in (a). We see that the allocation is intuitively appealing with all big threat locations adequately covered. We tried this algorithm on many test cases and found them to achieve their global minima.

P3. Interchangeable Capacity Constraints: In this problem the constraints add two new parameters to the formulation. The parameter μ_n $1 \leq n \leq p$ denotes the effectiveness of nth type resource and ν_n is total number of resources of nth type. If we denote the number of the nth type resources carried by UAV at location r_j by λ_{jn}, the constraint equation becomes $\sum_j \lambda_{jn} = \nu_n$ $1 \leq n \leq p$. Again, these constraints are independent of the resource locations r_j and the weight $p(r_j|x)$. Therefore as in P2, the new formulation can be interpreted as a change in the partition function: $Z := \sum_i \sum_n \lambda_{in}\mu_n e^{-\|x-r_i\|^2/T}$, where we are assuming

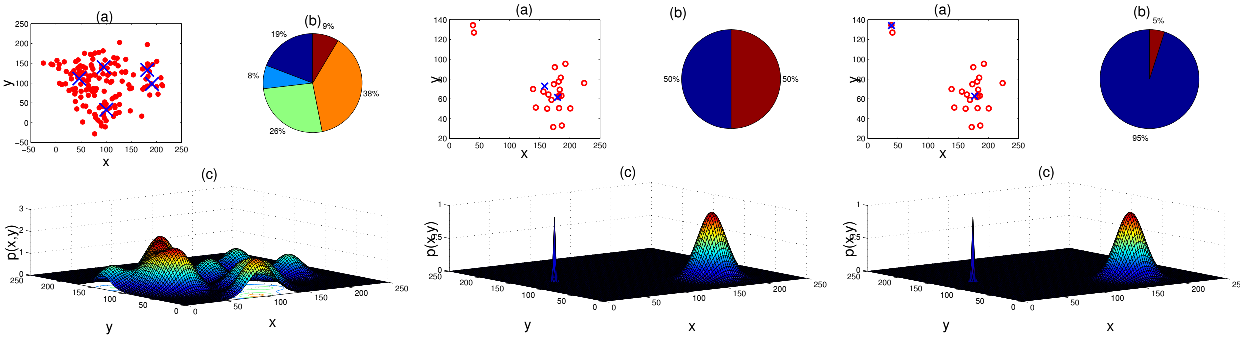

Figure 2: Left, Center and Right: (a) Threat locations (circles) and UAV locations (crosses). (b) The relative capacities of UAV as decided by the algorithm. (c) the probability density function of the threats.

that there are $\lambda_{jn}\mu_n$ units of n type resources at location r_j. This results in the following equations for $p(r_j|x)$ and the free energy:

$$p(r_j|x) \quad = \quad \frac{\lambda_{jn}\mu_n e^{-\|x-r_j\|^2/T}}{Z} \quad \text{and} \quad \hat{F} \quad = \quad -T\int_{\mathcal{D}} p(x)\log\sum_{in}\lambda_{in}\mu_n e^{-\|x-r_i\|^2/T}. \tag{5}$$

Adding the constraints to this equation, we derive the new lagrangian given by $F' = \hat{F} + \sum_n q_n(\sum_j \lambda_{jn} - \nu_n)$, where q_n $1 \leq n \leq p$ are Lagrange multipliers. Since the constraint is independent of resource locations, we obtain the same implicit equation for them (by setting $\frac{\partial F'}{\partial q_j} = 0$) as before given by Equations 4 except that $p(r_j|x)$ is given by the modified Gibbs distribution in Equation 5. It should be remarked here that we relax the constraint of λ_{jn} to be integers and allow them to take all positive real values. To determine the optimal values of λ_{jn} which corresponds to the problem of allocating resources to the UAVs, we set $\frac{\partial F'}{\partial \lambda_{jn}} = 0$

$$\Leftrightarrow \quad \frac{1}{T_k}\lambda_{jn}q_n = \int_{\mathcal{D}} p(r_j|x^n)p(x^n)dx^n \;\; \Leftrightarrow \frac{1}{T_k}\sum_n \nu_n q_n = \int_{\mathcal{D}} p(x)dx =: N(\text{ summing over } j \text{ and } n) \tag{6}$$

$$\Leftrightarrow \quad \frac{1}{T_k}\nu^T q = N, \quad \text{where} \;\; \nu = (\nu_1 \cdots \nu_p)^T \text{ and } q = (q_1 \cdots q_p)^T.$$

These are a linear set of equations on q and one of the solutions is given by $\frac{1}{T_k}q = \frac{N\nu}{\nu^T\nu} \Leftrightarrow \frac{1}{T_k}q_n = \frac{N\nu_n}{\nu^T\nu}$, which when substituted back into Equation 6, we obtain

$$\lambda_{jn} = \frac{\nu^T\nu}{N\nu_n}\int_{\mathcal{D}} p(r_j|x)p(x)dx, \tag{7}$$

where $N := \int_{\mathcal{D}} p(x)dx = 1$ denotes the total mass of the domain (normalized to 1 in our formulation). In this formulation, the algorithm consists of finding the resource locations $\{r_j\}$ and the resource allocations λ_{jn} at each temperature T_k by solving the Equations 4 and 7 respectively.

Figure 2(Left) (a) shows a scenario of 150 threat locations (circles) to which 5 UAVs have to be allocated. The probability distribution of threat locations is given in (c). There is only one kind of resource ($p = 1$). The relative amounts of resources allocated to each UAV as decided by the algorithm is shown in (b). The locations of the UAVs (crosses) as picked by the algorithm is shown in (a). The figure shows that there is an allocation which again seems intuitively reasonable. We tried this algorithm on many test cases and actually found them to achieve their global minima.

To bring out the difference between original and modified DA algorithm we did the following simulation (see Figure 2(Center and Right). We created two clusters of data sets: one large cluster with 20 threat locations and a small cluster with just 2 threats. We ran both the algorithms to find the optimal locations for two UAVs. The plot on the left corresponds to the original DA algorithm and the plot to the right corresponds to the modified DA algorithm. In both these plots, the threat clusters can be easily seen by their locations (circles) in (a) and the corresponding probability density functions in (c). We are considering here one type of resource scenario to have a fair comparison of the two algorithms. Plots (b) show that the original DA algorithm allots equal resources to each UAV and therefore places both of them in the vicinity of the big cluster (seen by locations of crosses in (a) while the modified algorithm distinguishes between the UAVs and allots a large part (95%) of resources to one UAV and allots it to the approximately the centroid of the larger cluster and gives a tiny percentage (5%) of resources to the other UAV and places it in the centroid of the smaller cluster (as seen by crosses in (a)).

P4. Multi Capacity Constraints: This problem involves further constraints on the resources. There are p types of resources and correspondingly there are p types of activities such that i type activity can only be serviced by an ith type resource. Furthermore, capacity constraints on each type of resource for each UAV is specified (denoted by W_{jn}). This problem differs from the above two problems as here there are p types of activities whose locations we denote by x^n and the corresponding probability fields by $p_n(x^n)$. The modified distortion and the entropy terms in this case are given by

$$D = \sum_n \int_{\mathcal{D}} p_n(x^n) \sum_{j=1}^{M} \|x^n - r_j\|^2 p(r_j|x^n) dx^n \text{ and } H = -\sum_n \int_{\mathcal{D}} p_n(x^n) \sum_{j=1}^{M} p(r_j|x^n) \log p(r_j|x^n) dx^n.$$

We proceed along the same lines as the DA algorithm by minimizing the Free Energy given by $F = D - T_k H$. The weights obtained by setting $\frac{\partial F}{\partial p(r_j|x^n)} = 0$ gives the following Gibbs distribution: $p(r_j|x^n) = \frac{e^{-\|x^n - r_j\|^2/T_k}}{Z_n}$,where the partition functions Z_n are given by $Z_n = \sum_i e^{-\|x^n - r_i\|^2/T_k}$and the resulting Free Energy is given by $\hat{F} = -\frac{1}{T_k} \sum_n \int_{\mathcal{D}} \log Z_n p_n(x^n) dx^n$.

Now if we denote the number of nth type resources in UAV at location r_j by λ_{jn}, we can modify the partition functions as

$$Z_n = \sum_i \lambda_{in} e^{-\|x^n - r_i\|^2/T_k} \tag{8}$$

in the same way as we did in the problems P2 and P3 and obtain the corresponding Gibbs distribution as $p(r_j|x^n) = \frac{\lambda_{jn} e^{-\|x^n - r_j\|^2/T_k}}{\sum_i \lambda_{in} e^{-\|x^n - r_i\|^2/T_k}}$. Adding the constraints to this equation, we derive the new Lagrangian given by

$$F' \;=\; -\frac{1}{T} \sum_n \int_{\mathcal{D}} \log \left(\sum_j \lambda_{jn} e^{-d(x^n, r_j)/T} \right) p_n(x^n) dx^n + \sum_j \sum_n q_{jn} (\lambda_{jn} - W_{jn}),$$

where q_{jn} $1 \le n \le p,\ 1 \le j \le M$ are Lagrange multipliers. The optimal resource locations are obtained by setting $\frac{\partial F'}{\partial r_j} = 0$) which gives the following set of implicit equations:

$$r_j = \frac{\sum_n p_n(x^n) p(r_j|x^n) x^n dx^n}{\sum_n p_n(x^n) p(r_j|x^n) dx^n}. \tag{9}$$

Plugging the constraint in this solution (equivalently setting $\frac{\partial F'}{\partial q_j} = 0$) we obtain the same equation for the resource locations as in Equation 9 with λ_{jn} substituted by W_{jn}.

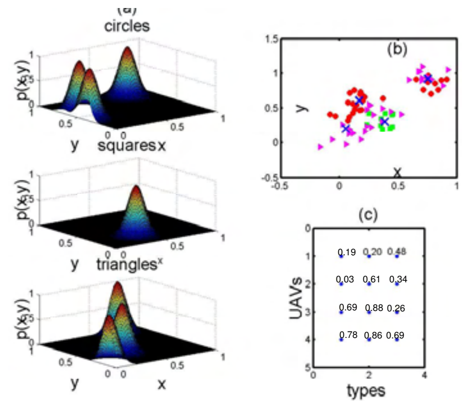

Figure 3: (a) probability density functions of different types of threats. (b) threat locations and allocated UAV locations. (c) The capacities of each UAV to different types.

Figure 3 represents a simulation of the scenario in which the distribution of three types of activity locations is shown in circles, squares and triangles (shown in (b)). The corresponding probability density functions are shown in (a). In this simulation, the matrix of $\{W_{jn}\}$ is represented in (c). Here we are allocating 4 UAVs to three types of threats (shown in (b)) with the probability density functions as shown in (a). The allocation is represented by the crosses in (b). We conducted many test simulations and found again that this algorithm gave global optima for these test cases.

4 Scaling laws and convergence rates

In the DA algorithm, at each iteration or equivalently for each value of the temperature T_k, it is needed to solve the following set of implicit equations

$$r_j = \int_{\mathcal{D}} p(x) \frac{e^{-\|x-r_j\|^2}}{\sum_i e^{-\|x-r_i\|^2}} x dx, \ 1 \leq j \leq M.$$

This is solved by using the following iteration scheme:

$$r_j(n+1) = \int_{\mathcal{D}} p(r_j(n)|x)p(x)xdx =: g_j(r(n)),$$

where $1 \leq j \leq M$, $n = 0, 1, 2, \ldots$ and $r_j(0)$ is assigned the solution of the implicit equations at the previous temperature value, i.e. at T_{k-1}. By denoting $r = (r_1 \ r_2 \ \ldots \ r_M)^T$ the above iteration scheme can be simply written as $r(n+1) = g(r(n))$. Here, we show that this iteration scheme is in fact equivalent to a *Descent Method* [17]. We consider the Free Energy $\hat{F} = -T_k \int_{\mathcal{D}} p(x) log \sum_i e^{-\|x-r_i\|^2} dx$ and therefore

$$\frac{\partial \hat{F}}{\partial r_j} = T_k \int_{\mathcal{D}} p(x)p(r_j|x)(r_j - x)dx = T_k p(r_j)\left(r_j - g_j(r)\right))$$

which implies

$$\nabla \hat{F} = T_k \hat{P}(r - g(r)) \Rightarrow r(n+1) = r(n) - \frac{1}{T_k}\hat{P}^{-1}\nabla \hat{F},$$

where $\hat{P} = diag(p(r_1(n)), \ldots, p(r_M(n)))$. Therefore the iteration is in the form $r(n+1) = r(n) + \alpha_k d_k$ where the descent direction $d_n = -\hat{P}^{-1}\nabla\hat{F}$ satisfies $d_n^T \nabla\hat{F} \leq 0$ with the equality being true only when $\nabla\hat{F} = 0$. This shows that this is a Descent Method. The advantage of this observation is that there is a vast literature on convergence rates of descent methods which can be used to analyze our iteration scheme.

Now we make another observation: if we scale the r and x variables by a scaling factor σ, i.e. we obtain a new scaled domain $\hat{\mathcal{D}}$ whose elements are given by $\hat{x} = x/\sigma$ and the corresponding resource locations by $\hat{r}_j = r_j/\sigma$, the nature of the optimization problem does not change (which is intuitively clear). In fact, we can easily verify that solving $\min \hat{F}(T_k, \mathcal{D})$ is equivalent to solving $\min \hat{F}(T_k/\sigma^2, \hat{\mathcal{D}})$. The implicit equation for the scaled domain is given by

$$r(n+1) = r(n) - \frac{\sigma^2}{T_k}\hat{P}^{-1}\nabla\hat{F},$$

where σ can be appropriately designed to obtain faster convergence rates. Design of such parameters forms an important part of the Descent Methods literature. Another interesting observation is that this scaling law relates the *spatial* scaling factorσ to the temperature variable T_k. This relation also justifies our intuition that to resolve smaller clusters one would require lower temperatures to start with. We propose to use scaling factors as functions of eigenvalues of covariance matrix of posteriori distribution $C_{x|r} = \int_{\mathcal{D}} p(x|r)(x-r)(x-r)^T dx$. Our preliminary results have shown an improvement in the convergence rates. This work is in progress and we are in the process of obtaining theoretical justification for our results.

5 Conclusions

In this paper, we have presented the modifications to deterministic annealing problem to solve some capacity constraint resource allocation problems originating from the task of prepositioning UAVs in a domain. These modifications were done by appropriately interpreting the Free Energy term in the DA algorithm. These algorithms were found to obtain global minima in many test simulations in which other methods as Lloyd's algorithm failed. These algorithms converge relatively faster (for example in comparison to simulated annealing) and they were implemented for real time applications for the optimal prepositioning and resource assignment of UAVs by Alphatech Inc, Burlington. We have also shown that the underlying structure of the problem considered in this paper and those considered in [1, 2, 3] are the same. We propose that the DA can be adapted to these formulations. We also identified spatial scaling laws scaling which have a close relation to temperature cooling (annealing laws) in the DA algorithm. We are investigating this relation and our preliminary results of exploiting it to obtain better convergence shows a great promise.

Acknowledgements

This research was supported by the Defense Advanced Research Projects, Information Systems Office (DARPA/IXO) and the United States Air Force Research Laboratory, Air Vehicles Directorate, Control Theory Optimization Branch (AFRL/VACA) under contract number F33615-01-C-3149.

References

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

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

[3] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage Control for Mobile Sensing Networks: Variations on a Theme. *Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002*, July 2002.

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

[5] Z. Drezner. *Facility Location: A survey of Applications and Methods.* Springer Series in Operations Research. Springer Verlag, New York, 1995.

[6] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, December 1999.

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

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

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

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

[11] S. Kirkpatrick, C.D. Gelatt, and M.P. Vechhi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[12] K. Rose. *Deterministic annealing, clustering, and optimization.* PhD thesis, California Institute of Technology, Pasadena, 1991.

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

[15] K. Rose. Constrained Clustering as an Optimization Method. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:785–794, August 1993.

[16] K. Rose. Statistical mechanics and phase transitions in clustering. *Physics Review Letters*, 65(8):945–948, 1990.

[17] A. Quarteroni, Sacco R., and F. Saleri. *Numerical Mathematics*, volume 37 of *Texts in Applied Mathematics.* Springer, 2000.