TECHNICAL RESEARCH REPORT

UAV Placement for Enhanced Connectivity in wireless Ad-hoc Networks

by Majid Raissi-Dehkordi, Karthikeyan Chandrashekar, John S. Baras

CSHCN TR 2004-18
(ISR TR 2004-34)

UAV Placement for Enhanced Connectivity in Wireless Ad-hoc Networks

Majid Raissi-Dehkordi
Institute for Systems Research
University of Maryland
College Park, Maryland 20742
Email: majid@isr.umd.edu

Karthikeyan Chandrashekar
Institute for Systems Research
University of Maryland
College Park, Maryland 20742
Email: karthikc@isr.umd.edu

John S. Baras
Institute for Systems Research
University of Maryland
College Park, Maryland 20742
Email: baras@isr.umd.edu

***Abstract*— In this paper we address the problem of providing full connectivity in large (wide area) ad hoc networks by placing advantaged nodes like UAVs (as relay nodes) in appropriate places. We provide a formulation where we can treat the connectivity problem as a clustering problem with a summation-form distortion function. We then adapt the Deterministic Annealing clustering algorithm to our formulation and using that we find the minimum number of UAVs required to provide connectivity and their locations. Furthermore, we describe enhancements that can be used to extend the basic connectivity problem to support notions of reliable connectivity that can lead to improved network performance. We establish the validity of our algorithm and compare its performance with optimal (exhaustive search) as well as non-opitmal (hard clustering) algorithms. We show that our algorithm is near-optimal both for the basic connectivity problem as well as extended notions of connectivity.**

Keywords: Mathematical Programming, Optimization, Combinatorics

I. Introduction

Recent emphasis in ad hoc network research has been on the study of connectivity, coverage and capacity issues for large ad hoc networks. These networks could be spatially large (wide area deployed) or large in terms of the number of nodes deployed (node density). Connectivity is a key issue especially in wide area critical ad hoc networks (battlefield networks). It is desirable in such networks that any pair of nodes should be able to communicate, albeit multihop using intermediate nodes as relays. More precisely, if the network is represented by a graph, with an edge between two nodes showing direct connectivity between them, complete connectivity is equal to having a connected graph. Connectivity amongst the nodes depends on a number of factors. The transmission range of the nodes determines the distance based connectivity. The nature of the terrain determines the propagation loss and therefore connectivity, thus two nodes which are within communication range may still not be able to communicate with each other due to the terrain induced path loss. Node movement in such networks is reflective of the tasks assigned to these nodes and therefore will result in mobility related connectivity changes. Widely dispersed networks may also result in the formation of independent groups where the nodes in each group may be connected but two groups may be disconnected. It is then reasonable to expect, and is in fact the case, that the network will not be fully connected at all times. Also, it might be important to ensure that certain high priority nodes in the network always remain connected.

One way to eliminate such disconnections is to deploy advantaged nodes with increased range (implies increased ground coverage) and improved capabilities so as to provide connectivity, reliability and improve QOS. Aerial platforms such as Unmanned Aerial Vehicles (UAVs) are ideal advantaged nodes for this purpose and the critical nature of the network justifies the use of such platforms. A group of UAVs can hover over the network and perform as relays to provide connectivity for the network. The role of UAVs is in fact not limited only to connectivity establishment, these devices have also been proposed for providing capacity, scalability and coverage [1]–[5] in different networks. In this paper, we address the problem of providing basic connectivity and other notions of connectivity that result in improved reliability and QOS for several disconnected ground sub-groups by placing a number of UAVs at the appropriate positions.

It is obvious that the minimum requirement for full connectivity of the network is for each sub-group to have at least one node communicating with an UAV (assuming the inter-UAV communication range is arbitrarily large). Since the UAVs are scarce and

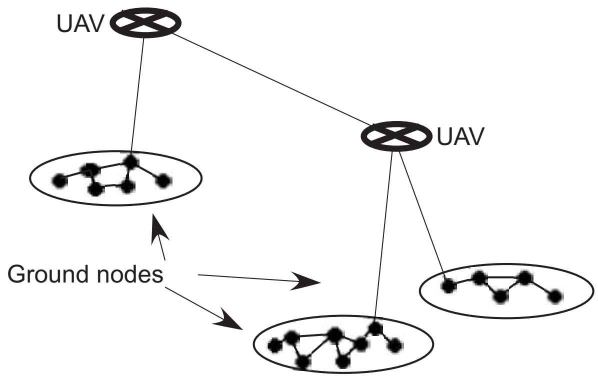

Fig. 1. Network with three partitions and two connecting UAVs.

expensive resources, the goal is to find the minimum number of required UAVs and their locations to have a fully connected network (Figure 1).

It is not difficult to see the similarities between the above problem and the Facility Location problems [6], [7] in the field of operations research. In particular, with the assumption of constant UAV altitude and communication range, our problem is similar to the continuous *p-center* problem [6], [8] and is in fact an extension of it. It is therefore easy to show that our placement problem is NP-hard using the NP-hardness results of the *p-center* problem [9], [10].

In order to avoid the computational complexity and difficulties of the $minimax$ problems, in this work we follow a clustering approach for finding near-optimal solutions to our problem. We will show that with careful definitions, the problem can be treated as a clustering problem with a summation-form distortion function and clustering algorithms can be used to address the problem. However, since the particular form of our formulation results in a non-convex optimization problem, we look for the clustering algorithms that avoid the local minima and result into near-optimal solutions. We use the Deterministic Annealing [11] algorithm for solving the above problem and also use a modification of this method to address a constrained version of the problem. Our results show that this approach performs very close to optimal and provides a flexible framework for addressing various forms of the UAV placement problem.

A. Related Work

To our knowledge, the problem of UAV placement in cooperation with multi-hop connectivity of the ground nodes has not been addressed before and we believe our formulation and results can motivate future work on this subject. There are, however, a number of similar works that have addressed other aspects of UAV placement when enough number of UAVs are placed in the network to provide direct coverage for all nodes as a backbone regardless of the node-to-node communication capability of the network. In [3] and [12], a novel architecture with ground nodes, UAVs and also unmanned ground vehicles is introduced and the performance of the overall network is evaluated by simulation studies. However, the UAV placement is with respect to the full coverage of the nodes which resembles the classic *p-center* problem and some other variations of it. [5] mainly deals with the implications of the UAVs on the routing protocol and its performance. The main source of related work on similar problems is the literature about the Facility Location problem and specifically the *p-center* problem. It is shown in [10] and [9] that the *p-center* problem is NP-Hard. Exact solutions are provided by [8], [13]–[15] all address the Euclidean distance p-center problem but involve rather inefficient schemes that do not scale to larger problems. There are also several heuristic methods for the *p-center* problem that are based on the assumption that the single facility location problem can be efficiently solved. All of those methods quickly find locally optimal solutions though. As mentioned before, our problem is an extension of the *p-center problem* and most of the methods mentioned above are either not efficient or not applicable to our problem. However, we used the placement idea based on the solution to the single facility location problem and proposed a low-complexity algorithm in [16]. Other constrained forms of the *p-center* problem have been studied in a number of papers (e.g. [17], [18]).

B. Our contribution

In this work we present methods which address the specific placement problems defined in the paper in such a way that the algorithm is scalable and also capable of avoiding local minima to some degree. We will show that our proposed method is flexible enough to address a number of extensions to the initial problem and provide solutions to them as well. We believe our results are also of value for the original *p-center* problem and operations research community since our general formulation contains that problem as a special case and we have not found any such approach to that problem in the literature.

This paper is organized as follows. Section II explains our assumptions and the exact formulation of our problem as a clustering problem. Section III presents a brief review of the Deterministic Annealing algorithm and explains how it is adapted to our problem. The results of the algorithm are presented in section IV where we compare the results with those obtained from an exhaustive search algorithm, as a benchmark, and also when using another well-known clustering algorithm. In section V, we present a number of extensions to the original problem and discuss how they can be accommodated in the algorithm and present some of the related results. Finally, section VI is dedicated to the concluding remarks.

II. Formulation of the Problem

A. Assumptions

In this paper, we make the simplifying assumption that all of the ground nodes have the same altitude. This assumption is only for having a better graphic representation of the results and is not a requirement of the algorithm. We also assume that the graph of the network is fully defined. In other words, the nodes are capable of detecting their neighbors via some type of neighborhood discovery protocol and also the locations of the nodes are known to the algorithm. In battlefield scenarios where the nodes are soldiers or vehicles, their locations are measured with GPS devices and are transmitted to their associated centers as part of the periodic health signals. Although the connectivity of the ground nodes heavily depends on the terrain, the relatively higher altitude of the UAV provides line-of-sight links to all the ground nodes regardless of the terrain structure. In other words, as long as a node is inside the geometric footprint of a UAV, it can communicate with the UAV. We also do not enforce any restrictions on the UAV-UAV communication range. We call each separate subgraph of the network a *cluster*. A cluster is therefore a collection of nodes that can all communicate with each other via single- or multi-hop paths and cannot communicate with any other nodes in the network.

B. Formulation

Let us denote by N the total number of ground nodes, M the number of subgraphs(clusters) in the network and by C_i; $i = 1, \ldots, M$ each of those clusters. We assume that the UAVs fly at a constant altitude h. The maximum node-to-UAV communication range is R

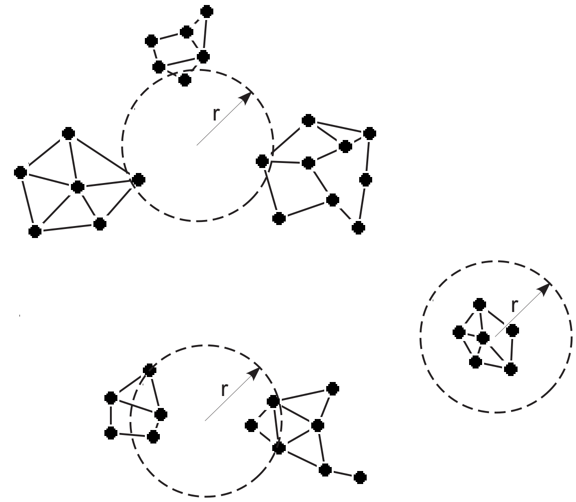

Fig. 2. Five clusters covered by three circles.

which, together with h, defines the maximum coverage radius $MaxRadius = \sqrt{R^2 - h^2}$ on the ground for each UAV which is greater than the maximum range for ground-ground communications. By definition, as long as at least one node from a cluster is within the communication range of a UAV, all nodes of that cluster can connect to the UAV via that node. The problem is therefore to find the minimum number of circles with radius $MaxRadius$ and their centers in such a way that at least one node from each cluster is within one circle. Figure 2 presents a graphical representation of our definitions.

If we denote by K the number of UAVs and by $y_1, \ldots, y_K$ their locations on the ground, the problem can be written as

$$Minimize\ K \tag{1}$$

$$subject\ to$$

$$\min_{y_1,\ldots,y_K} \max_{j\in\{1,\ldots,M\}} \min_{\substack{x\in C_j \\ i\in\{1,\ldots,K\}}} \|x - y_i\| \leq MaxRadius$$

where $\|x-y\|$ is the l^2-norm or the Euclidean distance between the points x and y on the ground. Finding the exact solution of this problem involves an exhaustive search on the different ways the nodes can be selected from each cluster and the ways clusters can be grouped together for coverage by single UAVs. It is not difficult to show that the problem is NP-hard, particularly, when we consider it as an extended version of the *p-center* problem [6], [7], [19]. The *p-center* problem in its optimization form tries to find the minimum number of

circles with a fixed radius and their locations to cover a given set of points. Our problem involves another degree of complexity for choosing the *best* nodes from each cluster. Thus, it seems, a natural approach would be to extend existing methods for solving the *p-center* problem and apply them to our problem. However, most of the existing works provide either complicated and inefficient methods (e.g. [8], [20], [21]), or heuristic methods (e.g. [14], [16], [22]–[24]) that are not necessarily extendable to other variants of the problem including our problem. Due to these difficulties, we propose a clustering approach for addressing our problem as follows:
Let's define the cluster-to-center distance function $l(C, y) : (\mathcal{N} \times \mathcal{R}^2) \times \mathcal{R}^2 \rightarrow \mathcal{R}$ as

$$l(C_j, y_i) = \min_{x \in C_j} \|x - y_i\|. \tag{2}$$

Problem 1 can then be written as

$$Minimize\ K \tag{3}$$
$$subject\ to$$
$$\min_{y_1,\ldots,y_K} \max_{j\in\{1,\ldots,M\}} \min_{i\in\{1,\ldots,K\}} l(C_j, y_i) \leq MaxRadius$$

From a clustering point of view, this is a clustering problem with a maximum distortion constraint which for every fixed K tries to find the optimal $y_1, \ldots, y_K$ locations to minimize the maximum value of distortion and increases K if it can't be smaller than $MaxRadius$. In other words, if each cluster is considered as a single entity, the goal is to find the best locations of the K centers(codewords) to minimize a worst-case distortion function. Using the approximation

$$\max(x_1, \ldots, x_n) \cong \sqrt[\alpha]{x_1^\alpha + \ldots + x_n^\alpha}\ for\ large\ \alpha,$$

and the fact that minimizing a fixed power α of the distortion function instead of the function itself will not change the results, we can approximate problem 3 with

$$Minimize\ K \tag{4}$$
$$subject\ to$$
$$\min_{y_1,\ldots,y_K} \textstyle\sum_{j=1}^{M} D(C_j, y_{u(j)}) < MaxRadius$$

where

$$D(C_j, y_i) = \min_{x \in C_j} \|x - y_i\|^\alpha \tag{5}$$

and $u(j) : \{1, \ldots, M\} \rightarrow \{1, \ldots, K\}$ is the function that assigns a center to every cluster C. It is well-known that the clustering problems of the above form are non-convex optimization problems except in special cases. In this work, we have used the Deterministic Annealing (DA) [11], [25] method to solve the clustering problem for near-optimal solutions. An advantage of the DA method, among others, is that it allows soft assignment of clusters to UAVs, which will be useful for addressing more general coverage problems, as we will see later, and also provides a better chance of converging to near-optimal solutions. In the following, we will first present a brief review of the DA method and then present how it has been used in the current work. To our knowledge, the use of the DA algorithm even for the traditional forms of the facility location problems is not studied before and our results introduce another approach for addressing those problems.

III. Deterministic Annealing and the UAV Placement Problem

A. Deterministic Annealing algorithm

Deterministic Annealing [11] is a stochastic method for clustering, compression, classification and similar problems where a large number of data points(nodes) need to be assigned to a small number of centers such that distortion function is minimized or bounded. The distortion function is usually an average over the distances of the nodes from their associated centers with the distance function defined based on the specific requirements of the problem at hand. Almost all distortion functions are non-convex and with multiple local minima [26] excluding the case with squared-Euclidean distance function. The clustering methods (e.g. [27], [28]) therefore lead to locally optimal results in the general case. The Deterministic Annealing(DA) approach tries to improve the clustering by turning a hard clustering problem into a soft clustering problem. The typical clustering problems are usually of the "hard" type where each data point has to be assigned to only one center. The distortion function for this type of problems can be written as

$$D = \sum_x p(x) d(x, y(x))$$

where x's are the data points, $y(x)$ is the center to which x is assigned to and, $d(.,.)$ is the appropriate distance function. DA defines the distortion function as

$$D = \sum_x p(x) \sum_y p(y|x) d(x, y)$$

where $p(y|x)$ is the association probability relating data point x to center y. Adding an entropy term H to the objective function and minimizing the new objective function

$$F = D - TH$$

will result into a joint optimization problem over y and $p(y|x)$ values and will naturally lead to the Gibbs distribution for the $p(y|x)$

$$p(y|x) = \frac{\exp\left(-\dfrac{d(x,y)}{T}\right)}{Z_x} \tag{6}$$

with

$$Z_x = \sum_y \exp\left(-\frac{d(x,y)}{T}\right)$$

being the normalization factor that plays the role of the partition function of statistical physics. The optimal y values are also given by

$$\sum_x p(x,y)\nabla_y d(x,y) = 0 \;\forall y. \tag{7}$$

or

$$\sum_x p(y|x)\nabla_y d(x,y) = 0 \;\forall y. \tag{8}$$

when all $p(x)$ are equal which is the case in our problem. The gradient is with respect to both elements of the y point and equation (8) is actually two equations. Parameter T in the above formulation controls the randomness level of the problem. For $T = 0$ the problem reduces to the hard clustering problem while for $T = \infty$ all centers will collapse to a single location and every x will equally belong to all centers. The DA algorithm then suggests that by starting from large values of T and updating the y and $p(y|x)$ while decreasing T gradually, a near-optimal solution can be achieved. It is shown in [11], that we can start with a single center and gradually add more centers during the cooling process as needed. Moreover, there exists a recipe for calculating the times when additional centers need to be introduced. This property is one of the advantages of this approach over the Simulated Annealing algorithm [29]–[31] that may come in mind due to the similarities between the two methods. However this calculation is not always straightforward and can become very complicated depending on the nature of the problem. Another advantage of the DA method, which is the source of the term ”deterministic”, is the deterministic nature of calculations of $p(y|x)$ and y values for fixed Ts as opposed to the ”random perturbation” approach of the Simulated Annealing. This will generally result into faster convergence of the algorithm and the execution time being more predictable.

B. UAV placement using Deterministic Annealing

The UAV Placement algorithm is shown in algorithm 1. We used the *mass constrained* [11] version of the DA which is computationally more efficient. We follow the steps shown in the algorithm to progressively update the center locations y_i and increase their number when necessary until every cluster is covered by at least one center. This criteria is checked by calculating the $K \times M$ Euclidean distance matrix L between every center and every cluster such that $L_{ij} = l(C_j, y_i)$.

A number of comments to clarify different stages of the above algorithm are in order. For simplicity we will refer to our algorithm as the UAV Placement algorithm from this point on.

1) The $allcovered$ flag is an indicator of the case where every cluster is inside the coverage circle of at least one UAV (lines 2 and 5 of the algorithm), this condition is verified by checking whether every column of matrix L has at least one element smaller than the $MaxRadius$ value.
2) Addition of the new centers in line 6 in practice is done by adding a small perturbation to the current location of one of the existing centers and dividing the probabilities associated with the current center equally between that and the new center. In other words, if center i is picked for perturbation

$$\begin{aligned} p(y_i) &= p(y_i)/2 \\ p(y_{K+1}) &= p(y_i) \\ K &= K+1. \end{aligned}$$

If the introduction of a new center is really needed at the current time step, the two circles move apart from each other. Otherwise, they will merge again after a few steps. Therefore, a few steps after each perturbation, the distance between the old and the new center is checked and if it is less than a threshold, the new center is removed and its probability is added back to the old center. Otherwise, the perturbation becomes permanent. The next perturbation attempt occurs after certain number of steps are passed since the decision about the previous perturbation was made. In [11] an exact condition for adding a new center is discussed. However, that method is not used here due to its computational complexity.
3) A large number of tests showed that the centers with farthest associated clusters are among the best choices for adding a new center based on perturbation.

Data : $MaxRadius$: coverage radius of each UAV,
N: number of nodes,
$X_i;\ i = 1, \ldots, N$: node locations,
$N \times N$ connectivity matrix of the nodes

Result : K: number of UAVs,
$y_i;\ i = 1, \ldots, K$: UAV locations

begin

- Calculate the $N \times N$ connectivity matrix
- Find the number of clusters M and their nodes
- $allcovered = 0,\ T = T_{init},\ K = 1$
- $p(y_1|C_i) = 1\ i = 1, \ldots, M$

1 - Calculate the location of y_1 from equation(s)

$$\sum_C p(y_1|C)\nabla_{y_1} D(C, y_1) = 0 \tag{9}$$

2 - Calculate the L matrix and the $allcovered$ flag

while $T > T_{min}$ *and* $allcovered = 0$ **do**

- Calculate for $i = 1, \ldots, K$

$$p(y_i) = \frac{1}{M}\sum_C p(y_i|C)$$

repeat

for $i = 1, \ldots, K$

3

$$p(y_i) = \frac{1}{M}\sum_{j=1}^{M} \frac{p(y_i)e^{-D(C_j,y_i)/T}}{Z_{C_j}}$$

$$where\ Z_{C_j} = \sum_{l=1}^{K} p(y_l)e^{-D(C_j,y_l)/T}$$

until *All* $p(y_i)$*s are stabilized*

- Update $p(y_i|C_j)$ values for all i and j

$$p(y_i|C_j) = \frac{p(y_i)e^{-D(C_j,y_i)/T}}{Z_{C_j}}$$

4 - Update y_i values from equation 9

5 - Update the L matrix and the $allcovered$ flag

if $allcovered = 0$ **then**

6 - Add a new center if needed

end

- $T = \beta T\ \ (\beta < 1)$

end

end

Algorithm 1: Finding the number and locations of the required UAVs

4) The recursion in line 3 of the algorithm has a unique solution and converges to the solution of that equation as the recursion goes on. A proof of this fact can be found in [32] where the Blahut's method for calculating the rate-distortion curve is explained.
5) Equation 9 is in fact the two following simultaneous equations in the x and y coordinates

$$y = \frac{\sum_C p(y|C)x(C,y)\left[D(x(C,y),y)\right]^{1-2/\alpha}}{\sum_C p(y|C)\left[D(x(C,y),y)\right]^{1-2/\alpha}} \tag{10}$$

$$\tag{11}$$

or

$$y = \frac{\sum_C p(y|C)x(C,y)\|x(C,y)-y\|^{\alpha-2}}{\sum_C p(y|C)\|x(C,y)-y\|^{\alpha-2}} \tag{12}$$

$$\tag{13}$$

where the x and y represent points on the plane as before. Also, notation $x(C, y)$ is used to represent the node in cluster C that is closest to center y. Solving the above equations in general involves finding the $x(C, y)$ node for every choice of y, and calculating the derivative based on that point. However, in situations where the the cluster diameters are small compared to the $MaxRadius$ value, taking the derivative with respect to the current $x \in C$ saves a lot of computation cost and does not have a considerable effect on the overall convergence speed of the algorithm. General convergence properties of this equation is currently under study.
6) Finally, the choice of the initial temperature depends on the range of values the distance function takes which itself depends on the form of the distance function and the width of the area. Our choice of the initial temperature was directly proportional to the average distance of the clusters from the center of mass of all clusters calculated via equations (10).

IV. EXPERIMENT SETUP AND RESULTS

We evaluate and compare the performance of the UAV Placement algorithm with the results of an optimal exhaustive search (Grid) algorithm as well as a well-known hard-clustering algorithm (K-means using the distance function (5)). These results provide us with an understanding of how close to optimal the algorithm performs and of its advantages over other clustering algorithms.

A. Validity of the distance approximation

Before evaluating the performance of the UAV Placement algorithm, we evaluate the validity of our distance approximation (4) and our choice of $\alpha = 10$ in the experiments. To do this, we design a simple scenario where 6 clusters are close to each other on one side of a hypothetical circle with radius $MaxRadius$ and another cluster is located on the other side of that circle (Figure 4). The nodes in the clusters are placed such that every cluster has just one node barely inside the circle. With this setting, only a circle aligned with our hypothetical circle can provide coverage to all clusters. We test our algorithm for two values of the exponent $\alpha = 2$ and $\alpha = 10$. With $\alpha = 2$, the distance function is the Euclidean distance and the update equations (10) reduce to finding the center of mass of the points. Figure 3 shows this case and we see that we need 2 UAVs to cover the network. On the other hand when $\alpha = 10$ our approximation for the max function is more precise and we need only one UAV as shown by Figure 4.

B. Performance evaluation and comparison

We generate a scenario with 170 nodes divided into 17 equally sized clusters. Furthermore we force the placement of nodes such that no two clusters are within communication range of each other, i.e., nodes can communicate (single/multi-hop) only to other nodes in the same cluster. The nodes are placed in an area of 1×1. The inter-node communication distance is 0.1 and the *MaxRadius* is set to 0.2. In the Grid algorithm we divide the area into grid points with a granularity of $.01$ i.e. the area is covered by a total of 10000 equally spaced points. We then perform an exhaustive search over the grid points to determine the minimum number of UAVs required to connect the network. Obviously, this procedure is not scalable and is only used in our relatively small scenarios to provide some benchmarks for our results. Figure 5 shows the output of the UAV Placement algorithm, we see that we need to place 5 UAVs to ensure that the network is fully connected. The algorithm converges in 76 iterations. Figure 6 shows the output of the Grid algorithm. The number of UAVs is again 5 indicating that the UAV placement algorithm performs well. Figure 7 shows the output of a hard-clustering algorithm that iteratively finds centers and assigns clusters to centers based on our distance function (5). Clearly the hard clustering algorithm will converge to a local minima and this is reflected in the output as now we need 6 UAVs to connect the network.

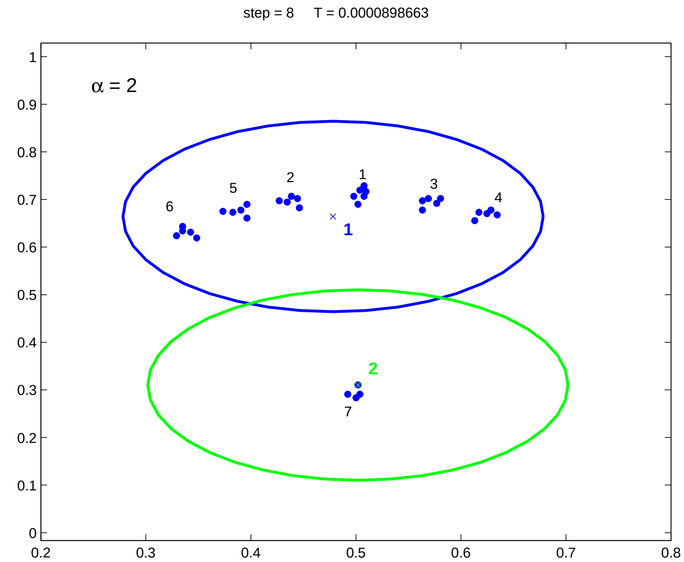

Fig. 3. Number of centers required when $\alpha = 2$

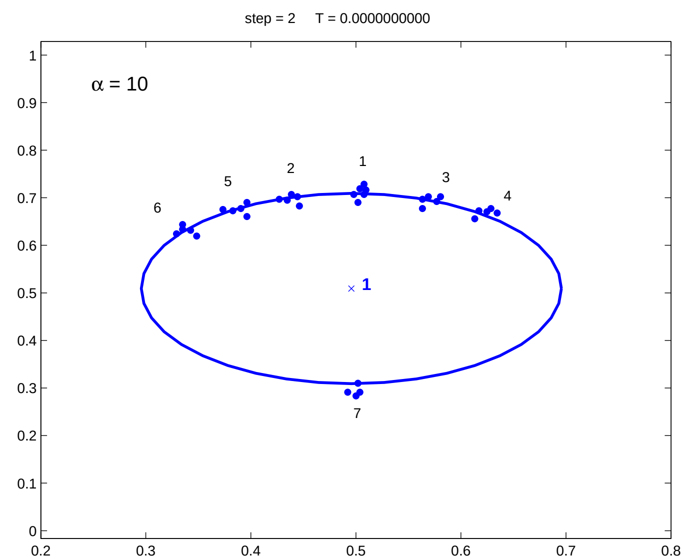

Fig. 4. Number of centers required when $\alpha = 10$

C. Dynamic scenarios

Ultimately, the goal would be to use the UAV Placement algorithm in a scenario where the ground nodes are moving. In this case we assume that the algorithm will be called periodically depending on ground node movement to determine the new number of UAVs and their locations to connect the network. It is then desirable that the algorithm converges as quickly as possible. In such situations we can take advantage of the fact that the new node locations will not be drastically different

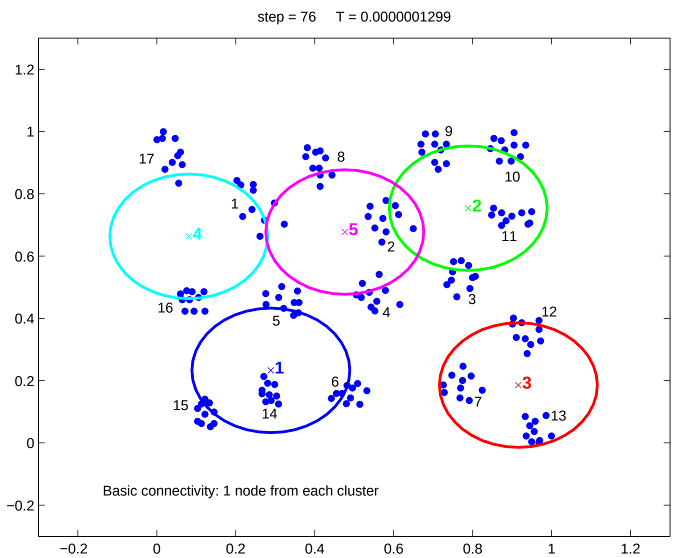

Fig. 5. UAV Placement Algorithm: Number and location of centers

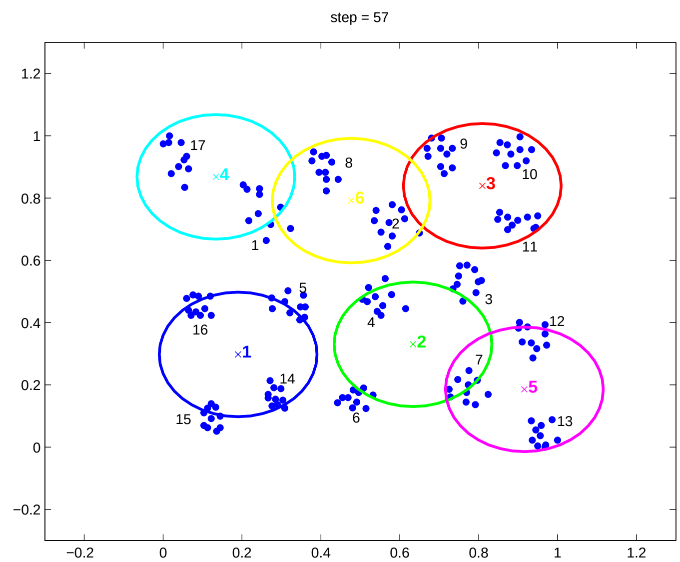

Fig. 7. Modified K-means: Number and location of centers

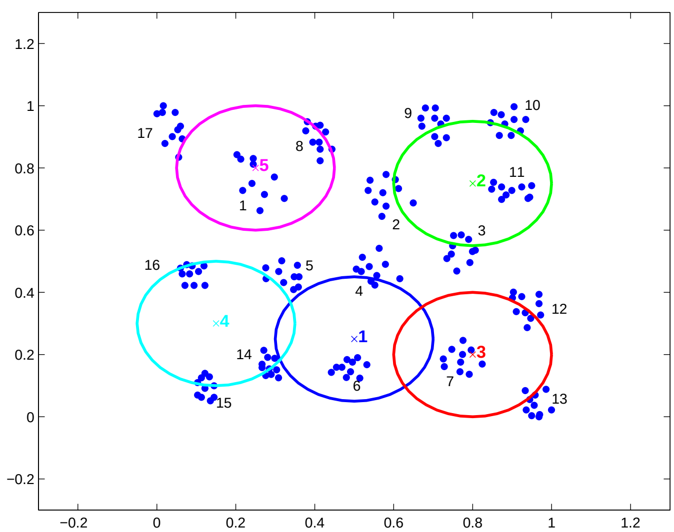

Fig. 6. Grid Algorithm: Number and location of centers

from their previous values. We can start the algorithm with a lower temperature T and the previous locations of UAVs as the new starting centers so that the convergence time is reduced. Figure 8 shows the new locations of the UAVs after displacing the nodes. We see that the number of iterations required is only 15. This represents a significant reduction in convergence time and implies that the algorithm when called periodically can be started at a lower temperature and then once in a while it can be run from a high temperature to ensure that the output is close to the global optimal.

A simplified analysis shows the complexity of the algorithm to be $O(TK(N+(B+3)M))$. In this calculation we have not included the complexity in determining the clusters and their members which is of complexity $O(N^3)$. This information can be typically obtained from the routing (eg. OLSR routing) and other protocol information in the network and be used by the algorithm. T is the number of iterations of the DA algorithm. B is the number of repetitions in Blahut's method for calculating $p(y_i)$. From our experiments we found that $E[B]$, the average number of repetitions to be around 5. B however clearly depends on K and M. Thus the algorithm performs at most $O(TBKN)$ operations. T reduces significantly when the algorithm is started from a lower temperature and with the previous locations of the UAVs (figure Figure 8).

V. Extensions to the UAV Placement Algorithm

Having shown that the algorithm finds a near-optimal solution to the basic connectivity problem, we can also extend the algorithm to solve more constrained definitions of connectivity to address key issues like reliability, redundancy and some notion of quality of service. In critical ad hoc networks where the ground topology is continuously changing we would like to have several of the nodes in a cluster to be able to reach the UAV in a single-hop rather than just one node being able to reach

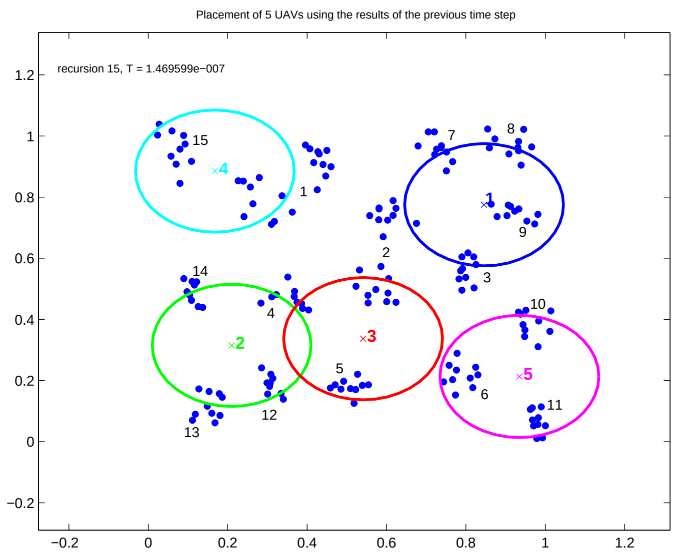

Fig. 8. Output when started from previous center locations and reduced temperature

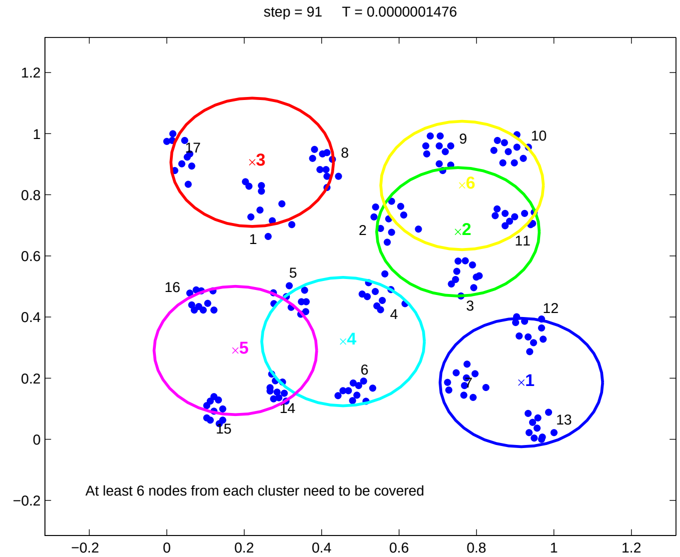

Fig. 9. Number and location of centers to connect 6 nodes in each cluster

the UAV (as required for the connectivity problem). Furthermore, it may be reasonable to provide multiple UAV support to clusters containing high priority nodes or nodes in need of more bandwidth. We discuss these two enhancements in detail in the ensuing sections.

A. *UAVs covering multiple nodes in each cluster*

As described above, covering multiple nodes in a cluster can have several benefits. First, the connectivity of the whole cluster to the UAV is more resilient to node mobility and ground conditions. Also, having multiple nodes in a cluster connected single-hop to the UAV can improve the overall capacity for the nodes in the cluster and also reduce congestion and routing overhead within the cluster that would result if there was only one node connected to an UAV.

We achieve the goal of covering multiple nodes in a cluster by modifying the distance function (5) and redefining it as:

$$D(C_j, y_i) = order_{x \in C_j}(d^{\alpha}(x, y_i), n) \qquad (14)$$

where the function $order(d, n)$ orders the nodes ($x \in C$) according to their distance from the center y_i and returns the distance of the nthclosest node $d^{\alpha}(min(n, |C|), y)$ where, n is the number of nodes required to be covered in each cluster and $|C|$ is the number of nodes in that cluster. $n = 1$ takes us back to the previous definition of distance. The algorithm using this modified distance function will find the minimum number of UAVs and their locations such that N nodes in each cluster are directly covered by the UAV. Figure 9 illustrates the results for the scenario where we require at least 6 nodes in each cluster to be directly connected to a UAV. We see that 6 UAVs are needed to provide network connectivity with the modified distance function. This enhancement emphasizes the flexibility of the algorithm in incorporating various network connectivity constraints.

B. *Multiple UAVs covering each cluster*

Another form of providing reliability and improving capacity for certain high traffic generating priority nodes would be to cover such clusters with multiple UAVs. This would also ensure some reliability through redundancy as the disappearance of one UAV does not result in the cluster being isolated.

In order to achieve this goal we make a fundamental change to the update procedure of the Deterministic Annealing algorithm. As described in Section III-A, the basic Deterministic Algorithm involves repetition of two steps. Finding the set of ys and updating the association probabilities $p(y|x)$ for the data points x. The association probability $p(y_i|x_j)$ indicates the influence of data point x_j in determining the center location y_i. The association probabilities start from an uniform

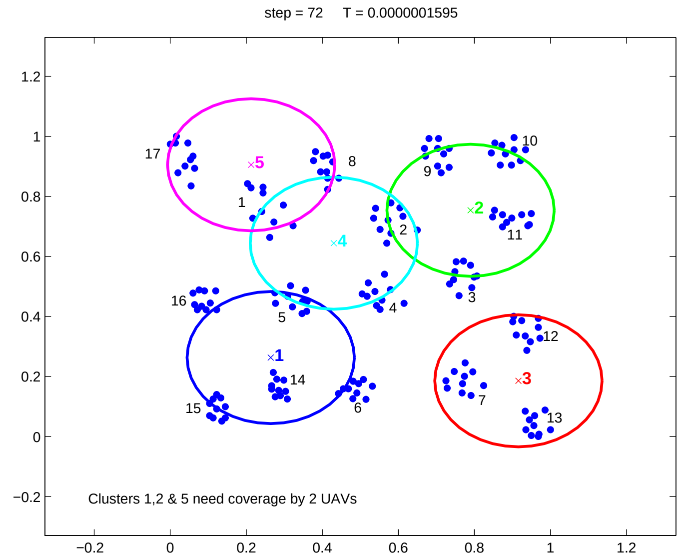

Fig. 10. Some clusters covered by 2 UAVs

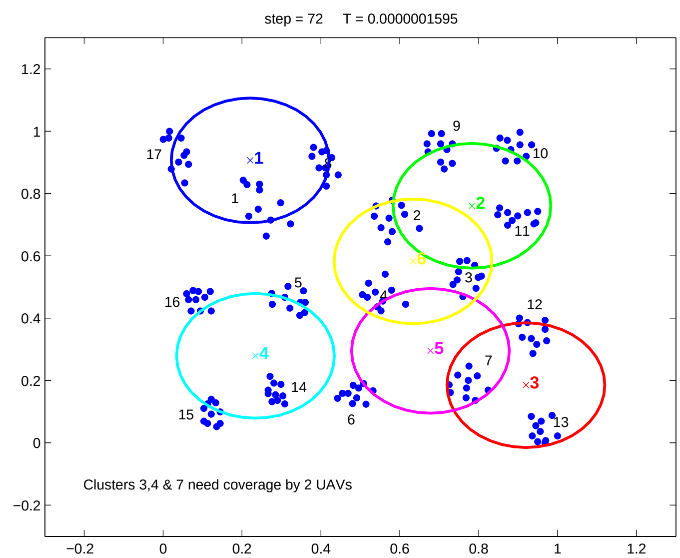

Fig. 11. Some clusters covered by 2 UAVs

distribution at high temperatures and converge to 0 or 1 value at low temperatures (hard clustering), where, each data point affects only one code word. We argue that for a cluster to be connected by L UAVs, at low temperatures the association probability of this cluster to the L centers should be large enough to determine (influence) the location of the L centers. This can be achieved by adjusting the association probabilities once they are calculated such that the first L probabilities for each cluster (ordered from max to min) are made equal. For example, let's assume that the calculated association probabilities for a cluster C ordered from max probability to min probability be $\mathbf{p}(y|C)$

$$\mathbf{p}(y|C) = [p_1 \; p_2 \; \dots \; p_K]$$

where k is the number of centers. The adjusted association probabilities for C would be

$$p(y|C) = [\frac{1}{L}\sum_{i=1}^{i=L} p_i \; \dots \; \frac{1}{L}\sum_{i=1}^{i=L} p_i \; p_{L+1}, \dots, p_K]$$

Intuitively, the system can now be thought of one where for each cluster, the temperature reduction affects only the $K-L$ centers that are not part of the adjustment process. For the L centers part of the adjustment process, reduction in temperature has no impact as they are equally influenced by the cluster through the adjusted association probabilities.
The modified algorithm using the adjustment to $p(y|x)$ will find the minimum number of UAVs and their locations such that each cluster is connected to L UAVs. Clearly, each cluster need not be covered by the same number of UAVs (L), depending on the requirement for additional connectivity, clusters can have different number of UAVs covering them. This follows directly from the above discussion as the association probabilities are adjusted per cluster.

Figure 10 shows the output of our enhanced algorithm when clusters 1,2 and 5 need to be covered by 2 UAVS. We see from Figure 5 that these clusters already have 2 UAVs covering them (though cluster 5 is covered by 1 UAV we can observe that it will covered by a minor perturbation of center 5) therefore the new algorithm should not require more UAVs. The output is consistent with this observation and the number of required UAVs is still 5. Figure 11 shows the output when clusters 3, 4 and 7 need to be covered by 2 UAVs. Clusters 3 and 4 in Figure 5 were covered by one UAV each and cluster 7 is already covered by 2 UAVs. The output shows that we need 6 UAVs to cover each of these clusters with 2 UAVs and the rest with one UAV. We now extend the test to consider the case where clusters 4 and 7 need to be covered by 2 UAVs, cluster 3 needs to be covered by 3 UAVs and the rest require basic connectivity. Figure 12 shows the output for this test. We see that we need 6 UAVs to provide connectivity based on the given constraints. The correctness of this result can be checked from visual observation from Figure 11 , where we see that cluster 3 is already covered by 3

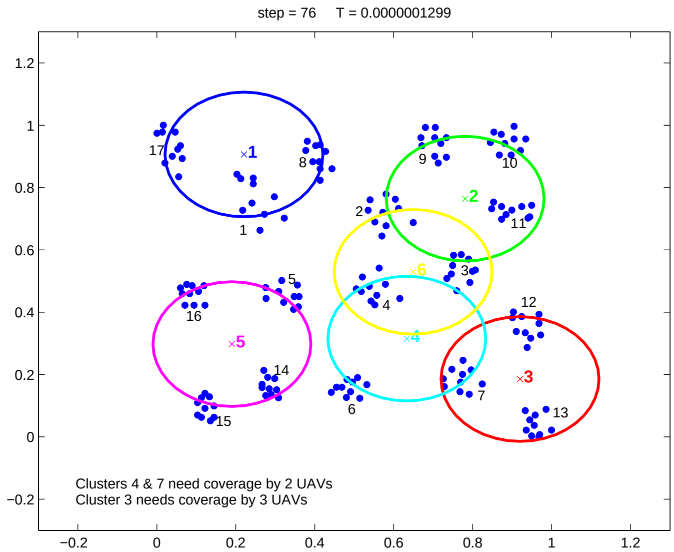

Fig. 12. Some clusters covered by 3 UAVs

UAVs. Unfortunately, the grid approach even for these small scale examples is not computationally feasible. This enhanced UAV Placement algorithm can thus find the minimum number of UAVs to provide constrained connectivity. This enhancement can be used to allocate resources (capacity, improved connectivity) to clusters based on their requirement and priority.

VI. Conclusion

Ensuring that the nodes in a wide area ad hoc network are always connected is very important especially in critical ad hoc networks (battlefield networks, rescue scenarios ...). We have shown that by formulating the connectivity problem as a clustering problem we can utilize existing well defined clustering solutions. The Deterministic Annealing algorithm as the clustering algorithm is a good choice as it can be adapted easily to the connectivity problem and its extensions, reaches the global optimum and has less execution time than other search algorithms.

Our approximation for the 'max' function holds for an exponent value of $\alpha = 10$. We establish the validity of the approximation through a simple simulation. The modified algorithm generates the optimal number of UAVs required to establish connectivity. This is established by comparison with the result of the exhaustive grid-search algorithm. We also show that a non-optimal algorithm (modified K-means) would in fact require more UAVs to establish connectivity due to its convergence to local minima. Successive runs of the algorithm starting from lower temperatures and the previous centers would converge faster. The algorithm can therefore be used in dynamic scenarios.

The algorithm and the distance function can be modified to support extended notions of connectivity like multiple nodes in each cluster being directly connected to the UAV and multiple UAVs covering each cluster. These enhanced connectivity constraints in fact mirror some of the current issues in ad hoc networks like capacity enhancement, resource allocation, reliable networks etc. Results show that the enhanced algorithm does in fact generate the minimum number of centers and their locations satisfying the imposed constraints.

The enhanced algorithm is capable of handling a wide range of connectivity constraints and the fact that the algorithm is suitable for dynamic scenarios make it a useful tool for network planning and resource allocation. Further extensions to the algorithm are currently being pursued, our current focus is on the capacitated-connectivity problem where the bandwidth constraints of the UAVs are also taken into account.

Acknowledgment

References

[1] J. L. Hillman et. al., "Communications network architectures for the army future combat system and objective force," 2002.

[2] B. Fletcher, "Autonomous vehicle programs and applications at spawar systems center," *Trnasactions of OCEANS '99 MTS/IEEE*, 1999, V.1, 196-200.

[3] I. Rubin and R. Zhang, "Performance behaviour of unmanned vehicle aided mobile backbone based wireless ad hoc networks," .

[4] I. Rubin, R. Zhang, and H. Ju, "Topological performance of mobile backbone based wireless ad hoc networks with unmanned vehicles," .

[5] D.L. Gu et. al., "Uav guided intelligent routing for ad-hoc wireless networks in single-area theatre," .

[6] *Facility Location; a survey of applications and methods*, Springer-Verlag, 1995.

[7] *Facility Location; Applications and Theory*, Springer-Verlag, 2002.

[8] R. Chen, "Solution of minisum and minimax location-allocation problems with euclidean distances," *Naval Research Logistics*, 1983.

[9] S. T. Masuyama et. al., "The computational complexity of the m-center problems on the plane," *Transactions of the IECE of Japan*, 2002, E64, 57-64.

[10] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on computing*, 1984, 13, 182-196.

[11] K. Rose, "Deterministic annealing for clustering, compression, classification, regression and related optimization problems," *IEEE*, 1998.

[12] S. Benjamin and I.Rubin, "Connected disc covering and applications to mobile gateway placement in ad hoc networks," in *Proceedings of ADHOC NetwOrks and Wireless (ADHOC-NOW)*.
[13] Z. Drezner, "The p-center problem - heuristic and optimal algorithms," *Journal of the Operational Research Society*, 1984.
[14] R. Vijay, "An algorithm for the p-center problem in the plane," *Transportation Science*, 1985.
[15] R. Chen and G. Y. Handler, "A relaxation method for the solution of the minimax location-allocation problem in euclidean space," *Naval Research Logistics Quarterly*, 1987.
[16] K. Chandrashekar, M. Raissi-Dekhordi, and J.S. Baras, "Providing full connectivity in large ad hoc network using aerial platforms," Tech. Rep., University of Maryland, College Park, 2004.
[17] S. Khuller and Y.J. Sussmann, "The capacitated k-center problem," *SIAM Journal of Discrete Mathematics*, 2000.
[18] S. Khuller, R. Pless, and Y.J. Sussmann, "Fault tolerant k-center problems," *Proceedings of the 3rd Italian conference on algorithms and complexity, CIAC 97*, 1997.
[19] *Logistics of Facility Location and Allocation*, Marcel Dekker, 2001.
[20] C.D.T. Watson-Gandy, "The multi-facility min-max webber problem," *European Journal of Operational Research*, 1984.
[21] Z. Drezner, "On the rectangular p-center problem," *Naval Research Logistics*, 1987.
[22] M.E. Dyer and A.M. Frieze, "A simple heuristic for the p-center problem," *Operation Research Letters*, 1985.
[23] H.A. Eiselt and G. Charlesworth, "A note on p-center problems in the plane," *Transportation Science*, 1986.
[24] S. Guha and S. Khuller, "Greedy strikes back: Improved facility location algorithms," *SIAM Journal of Discrete Mathematics*, 2000.
[25] K. Rose, "Vector quantization by deterministic annealing," *IEEE Transactions on Information Theory*, 1992.
[26] R. M. Gray and E. D. Karnin, "Multiple local minima in vector quantizers," *IEEE transactions on Information Theory*, 1982.
[27] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communication*, 1980.
[28] *Vector Quantization and Signal Compression*, Kluwer, 1991.
[29] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machin Intelligence*, 1984.
[30] S. Kirkpatrick et. al., "Optimization by simulated annealing," *Science*, 1983.
[31] B. Hajek, "A tutorial survey of theory and applications of simulated annealing," *Proc. 24th IEEE Conf. on Decision and Control*, 1985.
[32] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE transactions on Information Theory*, 1972.