Maximum Entropy Principle-Based Algorithm for Simultaneous Resource Location and Multihop Routing in Multiagent Networks

Nachiket V. Kale and Srinivasa M. Salapaka, *Member*, *IEEE*

Abstract—This paper presents a framework that develops algorithms for solving combined locational and multihop routing optimization problems. The objective is to determine resource node locations in a multiagent network and to specify the multihop routes from each agent to a common destination through a network of resource nodes that minimize total communication cost. These problems are computationally complex (NP-hard) where the cost functions are riddled with multiple minima. Algorithms based on Maximum Entropy Principle, which guarantee local minima and are heuristically designed to seek the global minimum are presented. These algorithms accommodate practical constraints on resource nodes as well as on the routing network architectures. Simulation results show that the multihop routes and resource locations allocated by these algorithms achieve lower costs (as low as 47 percent) than those algorithms where resource locational optimization is done without multihop routing or where the locational and routing optimization objectives are separated. The enabling feature of these algorithms is accommodating problems with resource constraints which is demonstrated through simulations.

Index Terms—Networks, optimization methods, resource management, routing.

1 Introduction

With rapid development in geopositioning information systems, cheap sensors, processing capability, and lower form factors, a variety of applications in fields such as wireless sensing networks, intelligent building systems, wireless communication, transportation, computer networking, facility allocation, and battlefield management have arisen that pose similar optimization problems. These problems simultaneously seek an optimal placement of resource nodes (locational optimization) that *covers* a set of sensor data elements as well as a *communication route* from each element to a destination center through the network of resource nodes. For instance, in intelligent building systems, locations of processing units (local control boxes) that are optimal in terms of covering a fixed network of sensors are needed. At the same time, communication routes from each sensor to the common (main control) center via the network of local control boxes that minimizes communication cost are needed. Similarly, in reconnaissance of an unknown territory, determining the locations of communication vans among a large number of agents as well as specifying a communication route from each agent to the control center poses the same locational and routing optimization problems. This paper proposes a framework to solve such problems that require combined locational and routing optimization.

The aim of locational optimization algorithms is to determine an optimal partition of a given area into *cells* and optimally assign a resource node location to each cell. The location of the resource node allocated to each cell of the partition is determined by minimizing a cost function defined by an appropriate *coverage* metric. Algorithms that address locational optimization problems (that do not consider routing) have been proposed in areas such as resource locational optimization [1], [2], [3], coverage control, mobile sensing networks [4], UAV strategic positioning [5], [6], data compression [7], drug discovery [3], and facility location assignment [8].

Multihop routing optimization aims at determining the optimal communication paths over a wired or wireless network. Given a set of nodes that form a network topology, a source element and a destination center, the message packet originates at the source, is communicated to the network through an entry node, passed on to other nodes in the network in maneuvers known as *hops* and finally communicated to the destination through an exit node. The aim of a routing algorithm is to generate a routing table that predetermines the route a message takes from a source to a destination through the network, for all possible combinations of sources and destinations. In many scenarios, multihop communication has benefits of cost reduction and energy management over single-hop communication. Multihop routing optimization algorithms are common in wireless sensor networks, packet switched networks, computer networking, satellite networking, and mobile networking.

Substantial research has been done in these two fields albeit without much overlap. Algorithms such as Lloyd's

- N.V. Kale is with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, 306 Talbot Laboratory, MC-236, 104 South Wright Street, Urbana, IL 61801. E-mail: nkale2@illinois.edu.
- S.M. Salapaka is with the Mechanical Science and Engineering Department, University of Illinois at Urbana-Champaign, 362c Mechanical Engineering Building, 1206 West Green Street, Urbana, IL 61801. E-mail: salapaka@illinois.edu.

Manuscript received 21 Sept. 2010; revised 21 Jan. 2011; accepted 4 Mar. 2011; published online 28 Apr. 2011.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-09-0438.
Digital Object Identifier no. 10.1109/TMC.2011.93.

1536-1233/12/$31.00 © 2012 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

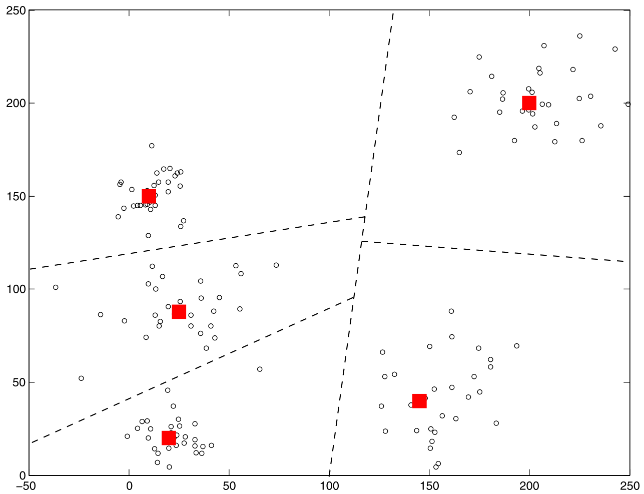


Fig. 1. An illustration of the resource allocation problem. Note the optimal partitioning of the area into cells and the subsequent allocations of a resource (red square) to each cell at an optimum location.

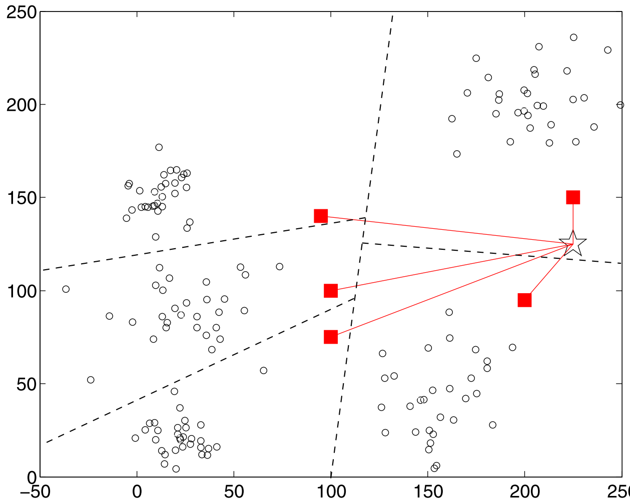


Fig. 2. An illustration of the resource allocation problem and subsequent routing. Note the optimal partitioning of the area into cells and the subsequent allocations of a resource (red square) to each cell at an optimum location to minimize the cost of communication between the elements and the destination center (star).

algorithm [9], [10], simulated annealing [11], and deterministic annealing (DA) [7] address locational optimization problems, but however, do not consider routing. Similarly, vast literature exists for routing-only problems, where various situation-specific and general routing algorithms have been developed for mobile communication networks and wireless sensor networks [12] applications. For instance these algorithms have been developed for ad hoc geographic routing [13], virtual coordinates routing [14], proactive, reactive or hybrid routing [15], distributed routing, and adaptive routing [16] where they address routing problems either over *fixed* resource nodes, and thus do not consider locational optimization aspects, or address networks with mobile nodes which consider a change in node location only as a disturbance rather than a design parameter. It needs to be emphasized here that this paper does not propose a new routing protocol. As shall be elaborated in the following sections, the algorithm takes into account *all* possible routes while finding the optimal resource locations and after the resources have been allocated, forms a routing table by performing a comprehensive search of all the routes under consideration.

Computational complexity forms the main challenge in both the locational optimization and the routing problems. The associated optimization problems are computationally complex [17] as the number of associations of sensor data elements to resource nodes depend combinatorially on their numbers. For example, allocating five resources over an area of 100×100 with 100 resource locations per unit area requires optimizing over 10^{18} combinations. In fact, even simple combinatorial resource-allocation problems such as k-means problems have been proved to be NP-hard when the number of resources k or the dimension of the underlying space form input parameters [18], [19], [20]. The computational complexity precludes divide-and-conquer search methods for global optimum, and therefore typically algorithms based on heuristics that solve approximations or relaxations of the underlying problem are sought. The difficulty in developing algorithms for these problems stems from their inherent nonconvex nature. The cost function is riddled with many local minima, and local algorithms such as Lloyd's algorithm, in the context of locational optimization, generally get trapped in the local minima and are extremely sensitive to the initialization step in their procedures. Further, in a multihop communication scenario, the number of ways a message can be routed in a network of nodes is also combinatorial, which along with the directionality of the message flow, results in numerous routes through various permutations of the nodes. Routing in a network being a challenging problem on its own, it adds substantially to the overall computational demands in the combined locational and routing optimization problem. Figs. 1 and 2 illustrate a resource allocation problem without and with routing, respectively.

The main contribution of this paper is a framework based on Maximum-Entropy-Principle (MEP) that *simultaneously* addresses the problems of locational and multihop routing. The resulting algorithms have close parallels to the Deterministic Annealing algorithm developed for locational optimization problems in data compression area [7] and thereby inherit a lot of its features, especially its versatility in terms of accommodating constraints on resource nodes. This framework is flexible in accommodating constraints on the network topology as well as size and functionality of resources. This flexibility allows for incorporating various problem-specific constraints that arise in different application areas while still addressing the locational and routing aspects common to all of them. The algorithms are heuristically designed to avoid local minima and are insensitive to the initialization step.

The central concept of the proposed algorithm is based on developing a homotopy from an appropriate convex function to the nonconvex cost function; where at every step of the homotopy the local minima of the cost function serves as the initialization for the subsequent step, analogous to the DA algorithm in [7]. Since minimization of the initial convex function yields the global minimum, this procedure is independent of initialization. The heuristic

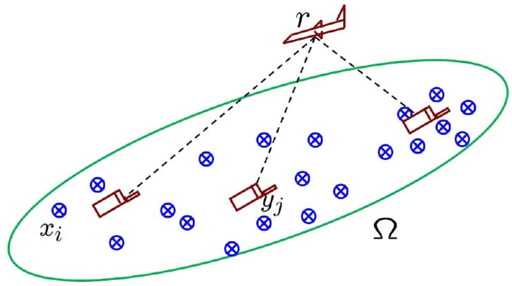


Fig. 3. A reconnaissance mission scenario: In a reconnaissance mission of an unknown terrain Ω, $x_i, 1 \leq i \leq N$ represent locations of agents who can communicate with communication sets at location $y_j,\ 1 \leq j \leq M$ to a UAV at location r. The communication sets can relay data directly to the UAV or pass it on to another set. The associated combinatorial optimization problem is to find the positions y_j of the communication sets and the routes for data communication from each individual to the UAV such that total communication cost is minimized.

is that the global minimum is tracked as the initial convex function deforms into the desired nonconvex cost function via the homotopy. The choice of the initial convex function is a Shannon entropy function that is motivated by an application of MEP. This MEP heuristic has a close parallel to the law of minimum free energy in statistical thermodynamics that applies to similar combinatorial problems in nature.

The paper is organized as follows: In Section 2, we state the combined locational and multihop routing problem and formulate it in a MEP framework. In Section 3, we propose and discuss a solution to a basic problem in this framework. Section 4 elaborates the flexibility of the framework by incorporating various constraints and additions structures to the problem scenarios and finding the corresponding solutions. Section 5 details the simulation results. In Section 6, we analyze and discuss the finer aspects of the algorithm. We conclude the paper with conclusions and future work in Section 7.

2 Problem Formulation

As described in the previous section, simultaneous locational optimization and routing problems arise in diverse applications. These problems are characterized by overlaying a communication network of M *resource nodes* y_j on a network of large number $N \gg M$ of sensor data nodes (*elements*) x_i such that total communication cost of sending data from each element to a destination center is minimized. For instance, consider a reconnaissance scenario (Fig. 3) where the objective is to gather intelligence from a distribution of agents on an unknown terrain and relay it to a central system, an unmanned aerial vehicle (UAV) hovering over the area (the UAV location is unchanging). There exist on the terrain a limited number of mobile units with communication handsets that can communicate with the agents, with each other and with the UAV. The agents relay their intelligence through the network of the mobile units to the UAV in a multihop fashion where one mobile unit receives the packet from an agent, and the packet is relayed from one unit to another until the last hop which occurs when a unit passes the message to the UAV. Note that, in this model, the agents do not directly communicate with the UAV which reflects practical constraints such as lack of sufficient communication range. Another area where a similar problem arises is in building sensor networks. In these networks, there exist a finite number of sensors (for example, fire detectors, temperature sensors, motion detectors) distributed over a building. These sensors communicate to a centralized computer through a network of communication nodes. The sensors cannot directly communicate with the computer. A sensor transmits its information to a communication node which relays it on to other nodes in a multihop fashion until it reaches the computer in the last hop. Note that the agents, the communication vans, and the UAV in the reconnaissance example, and the sensors, the communication nodes, and the centralized computer in building sensor networks, respectively, represent the sensor data elements, the resource nodes, and the destination center in the context of the combined locational and routing optimization problems.

The objective of the combined locational and routing optimization algorithm is twofold:

1. To partition the elements into cells, i.e., *clusters* and assign a resource node to each cluster and determine its location.
2. To determine the route, a communication packet must take from the element to the destination center using multiple hops through the network of resource nodes.

More specifically, we consider N elements whose locations (or spatial coordinates) are given by $x_i \in \Omega \subset \mathbb{R}^d$, where d is the dimension of the locational coordinates (for ease of illustration, we will consider $d = 2$ in this paper), and $M \ll N$ resource nodes $y_j \subset \mathbb{R}^d$. Each element can communicate with any resource node, and the resource nodes can communicate with each other. There is a destination center at location $r \in \mathbb{R}^d$ which can receive communication signals from the resource nodes. The objective is to transmit data from the elements to the destination center through the network of resource nodes $\{y_j\}$, where the signal can take a multihop route over the network. The associated locational optimization problem is to find the locations of the resource nodes y_k which yields the lowest total communication cost.

Let

$$z_l := [y'_{l_1}, y'_{l_2}, \ldots, y'_{l_q}]', \quad 1 \leq q \leq M, \tag{1}$$

denote a sequence of resource nodes (a vector in $\mathbb{R}^{2q}$) we refer to as a *route*. Here, $[(\cdot)]'$ represents the transpose operation. When x_i uses route z_l to communicate with the destination center at r, the communication packet follows the following hop sequence:

$$x_i \rightarrow y_{l_1} \rightarrow y_{l_2} \rightarrow \cdots \rightarrow y_{l_q} \rightarrow r.$$

Note that we include routes of all lengths with the number of hops less than or equal to M. We denote the cost of communication between x_i and r using the route z_l as $d(x_i, z_l)$. The corresponding combinatorial optimization problem can be written as

$$D_0 = \min_{\{y_j\}} \sum_{i=1}^{N} \Big[p(x_i) \min_{z_l}(d(x_i, z_l)) \Big], \tag{2}$$

where $p(x_i)$ represents relative importance of the data possessed by the ith element with $\sum_i p(x_i) = 1$. The algorithm assumes a generic communication cost $d(x_i, z_l)$, though we illustrate it through a simple model given by

$$d(x_i, z_l) = \|x_i - y_{l_1}\|^2 + \sum_{i=1}^{q-1} \theta_i \|y_{l_i} - y_{l_{i+1}}\|^2 + \theta_M \|y_{l_q} - r\|^2, \quad (3)$$

where route z_l is defined by (1), constants $\theta_i > 0$ denote relative weights and $\|(\cdot)\|$ represents the euclidean distance. The first term in this choice of distance function represents communication cost from the element to the first resource node in a route, the second term represents the cost of hops in between the resource nodes, and the last term represents the cost from the last resource node to the destination center. The euclidean distance metric approximately captures the inverse-square law of electromagnetic radiation where power density at a point varies as the inverse of the square of its distance from the source. A minimum power density is required for detection of the wireless signal, which leads to the weighted euclidean square distance as a good approximation of the wireless communication cost. We also assume that the communication range of the resource nodes is larger than the dimensions of the physical domain under consideration, enabling them to communicate with the elements, other resource nodes, and the destination center irrespective of their locations and all channels are free of interference. For ease of illustration, in this paper we assume $\theta_i = 1$ for all $1 \leq i \leq M$ unless stated otherwise. Note that if we represent the set of resource nodes by $Y = [y_1' \ y_2' \ \cdots \ y_M']'$, then the route-vector z_l in (1) is given by $z_l = H_l Y$ where $H_l \in R^{2q \times 2M}$ whose ith row is given by $e_{l_i} \otimes [1\ 1]$. Here, $e_{l_i} \in \mathbb{R}^M$ represents l_i^{th} row of identity matrix, $\otimes$ represents the Kronecker product, and q is the *route-length*, that is, the number of resource nodes in z_l. In this notation, the distance function can be rewritten as

$$d(x_i, z_l) = \left\| \begin{pmatrix} x_i \\ z_l \end{pmatrix} - \begin{pmatrix} z_l \\ r \end{pmatrix} \right\|^2 = \|Q_i + \Upsilon_l Y\|^2,$$

where

$$\Upsilon_l = \left[\begin{pmatrix} 0 \\ H_l \end{pmatrix} - \begin{pmatrix} H_l \\ 0 \end{pmatrix} \right] \in \mathbb{R}^{2(q+1) \times 2M}, \quad (4)$$

$$Q_i = (\, x_i \quad 0 \quad \ldots \quad 0 \quad -r\,)'. \quad (5)$$

3 Problem Solution

A straightforward approach would be to first carry out locational optimization by partitioning the set of elements by assigning a resource node to each element and then perform communication routing optimization on the network. The former can be carried out using any of the iterative algorithms available for locational optimization. For instance the deterministic annealing algorithm is heuristically designed to seek the global minimum in resource locational optimization problems [7]. Consider that once the locations of the resource nodes are determined using this algorithm, brute-force routing optimization is performed on the network which creates a routing table that dictates the least cost path originating at a resource node and terminating at the destination center. In the following sections, we shall consider this "two-step algorithm" as a comparison benchmark for the framework developed in this paper. Qualitatively (as corroborated by simulations), this two-step algorithm has a reduced flexibility and effectively a lower probability of finding the global optimum than an algorithm which moves over the combined locational-routing solution space.

The main feature of the algorithm we propose is that it is insensitive to the initial choice of routes and simultaneously addresses the locational as well as routing issues. This is achieved by allowing each element $x_i \in \Omega$ to be associated with every route z_l through a weighting parameter $p(z_l|x_i)$ (without loss of generality, these are such that $\sum_j p(z_l|x_i) = 1$ for every $1 \leq i \leq N$). This algorithm seeks minimization of a modified cost function given by

$$D = \sum_{i=1}^{N} p(x_i) \sum_{l=1}^{M} p(z_l|x_i) d(x_i, z_l). \quad (6)$$

We note here that the instantaneous weighting term $p(z_l|x_i)$ can alternatively be viewed as a probability of association between the element x_i and the route z_l. The choice of the weights $p(z_l|x_i)$ is crucial in determining the trade-off between the sensitivity to the initial choice of routes and the deviation of the modified cost (6) from the original cost function (2). For instance, a uniform distribution function $p(z_l|x_i) = 1/M$ makes the minimization of (6) with respect to z_l independent of the initial choice of z_l; however, the corresponding distortion function D is considerably different from the cost function in (2). While (6) denotes the average *weighted* distance of an element from a resource node, (2) is a measure of the average distance of an element from its nearest resource node. With the specific choice of a distribution that assigns $p(z_l|x_i) = 1$ when $d(x_i, z_l) = \min_k d(x_i, z_k)$ but zero otherwise, the distortion D in (6) becomes identical to the cost function in (2). Thus, minimizing D in the formulation of (6), optimization of hard partitions is replaced by optimizing over weights. The Maximum Entropy Principle [21], [22] provides a systematic way to determine a distribution that achieves a specific feasible value of the cost function, and thereby achieves a prespecified trade-off in the above context. More specifically, we seek a probability distribution $p(z_l|x_i)$ that maximizes the Shannon Entropy [23]

$$H = -\sum_{i=1}^{N} p(x_i) \sum_{l=1}^{M} p(z_l|x_i) \log(p(z_l|x_i)), \quad (7)$$

such that

$$D = \sum_{i=1}^{N} p(x_i) \sum_{l=1}^{M} p(z_l|x_i) d(x_i, z_l) = D_0.$$

The entropy term quantifies the level of randomness in the distribution of association probabilities, and thus its maximization results in a distribution that is maximally noncommittal toward any route while guaranteeing that the distortion attains the value D_0. Therefore in this framework, we first determine the weighting functions by maximizing the unconstrained Lagrangian

$$L = H(z|x) - \beta(D(x,z) - D_0),$$

with respect to $\{p(z_l|x_i)\}$, where $H(z|x)$ and $D(x,z)$ are given by (7) and (6), respectively, D_0 is the value of the cost function which the algorithm aims to achieve, and β is the Lagrange multiplier. This is equivalent to the following minimization problem:

$$\min_{\{p(z_l|x_i)\}} \underbrace{D - TH}_{\triangleq F}, \tag{8}$$

where the Lagrange multiplier $T = \frac{1}{\beta}$ and the term F are called *temperature* and *Free energy*, respectively, due to a close analogy to quantities in statistical thermodynamics (where Free energy is the difference between enthalpy (D) and temperature times entropy (TH)) [24].

The MEP theorem (see the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2011.93, and [21], [22]) gives the explicit solution for the weights given by the Gibbs distribution

$$p(z_l|x_i) = \frac{e^{-\beta d(x_i,z_l)}}{\sum_k e^{-\beta d(x_i,z_k)}}. \tag{9}$$

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

$$F = -\frac{1}{\beta}\sum_i p(x_i)\log\sum_l e^{-\beta d(x_i,z_l)}. \tag{10}$$

The resource locations Y are obtained by minimizing (local) F by setting $\frac{\partial F}{\partial Y} = 0$ which yields

$$\sum_i p(x_i)\sum_l p(z_l|x_i)\left[\Upsilon_l'\begin{pmatrix} x_i \\ 0 \\ -r \end{pmatrix} + \Upsilon_l'\Upsilon_l Y\right] = 0,$$

which implies

$$Y = \frac{1}{\sum_l p(z_l)\Upsilon_l'\Upsilon_l}\sum_{i,l} p(x_i,z_l)(\Upsilon_l' e_1 x_i - \Upsilon_l' e_{M+1} r), \tag{11}$$

where $p(x_i,z_l) = p(x_i)p(z_l|x_i)$.

This approach specifies the homotopy from the convex function $-H$ to the cost function D, which is parameterized by the temperature variable T. Clearly for small values of β (large values of T) in (8), we mainly attempt to maximize the entropy. Thus, a choice of weights corresponding to a high value (near infinity) of T leads to algorithms that are insensitive to the initial choice of routes, since their subsequent associations are affected almost equally by all the elements. As β is increased (T is lowered), we trade entropy for the reduction in the distortion, and as β approaches infinity (T approaches zero), we minimize D directly to obtain a *hard* (nonrandom) solution to the original problem (with the cost function as in (2)), where $p(z_l|x_i)$ taking either the value 0 or 1. Accordingly, in this algorithm, an *annealing* process is incorporated where the minimization problem (8) is repeatedly solved at different values $\beta = \beta_k$ where β_k is increased with k as

$$\beta_{k+1} = \gamma\beta_k, \quad \gamma > 1.$$

At each step we determine $p(z_l|x_i)$ using (9) and then use (11) to solve for the new resource node locations $\{y_j\}$.

Thus, the proposed algorithm consists of the following steps:

- Step 1: Start at a high value of T (low β) and initialize all nodes $\{y_j\}$ at the weighted centroid of the elements $\{x_i\}$, i.e., $y_j = \sum_i p(x_i)x_i$.
- Step 2: Compute the probability distributions $p(z_l|x_i)$ using (9).
- Step 3: Compute new node locations $\{y_j\}$ using (11).
- Step 4: Increase β; Stop if it reaches a user-defined maximum value, else goto Step 2.

An analysis of this algorithm reveals a *phase transition* structure (similar to the DA algorithm in [7]), which is explained as follows: Since the algorithm begins with a very high value of the temperature parameter T (ideally with $T = \infty$) the Free Energy term F in (8) is completely dominated by the Entropy term H, and its minimization results in a *uniform* distribution for the weights given by $p(z_l|x_i) = 1/M$. Thus, since the initial weights are all identical, the initial resource node locations $\{y_j\}$ computed from (8)-(10) are coincident and thereby the corresponding z_l are identical. As the annealing process continues (by decreasing the temperature parameter) and crosses a *critical temperature* value, the resource node location splits into multiple *distinct* locations. That is, *phase transition* from one to multiple resource node locations occurs. The annealing process is marked by successive such critical temperatures and phase transitions, at each of which, the number of distinct resource locations increases. There are two reasons why these critical temperatures are of interest. The first reason is motivated by the fact that the resource node locations do not vary continuously as a function of the annealing parameter albeit their variations are dominated by phase transitions. Therefore, any annealing law that crosses the critical temperatures successively is desired. The second reason is that since the number of distinct resource locations increases at every phase transition, it provides a mechanism in the algorithm to check if the number of distinct resource locations has reached the prespecified maximum number of resources. In this section, we shall analytically calculate the critical temperature values and also give an intuitive analysis of the underlying process.

The splitting of resources can easily be explained by tracking the conditions for attaining the minimum of the Free Energy F as a function of the annealing parameter β. At $\beta = 0$, the Free Energy F is convex, and therefore the minimum obtained by $\frac{\partial F}{\partial Y} = 0$ is the global minimum and the Hessian $\frac{\partial^2 F}{\partial Y^2}$ is positive definite. As β is increased, there exists a specific value of β_c (the critical temperature $T_c = 1/\beta_c$) at which the Hessian is no longer positive definite which in turn implies that $\frac{\partial F}{\partial Y} = 0$ does not necessarily imply a local minimum. In the case, it is a local maximum (or saddle point), an iteration of the algorithm drives Y to the new local minimum and Y tracks this local minimum as the annealing parameter β is increased till the next critical temperature is reached. In the following, we determine the conditions for critical temperature (where the Hessian loses positive definiteness) by employing variational calculus. Let $\{Y + \epsilon\psi\}$ denote a solution (Y is at a local minimum of F) that has been perturbed by a perturbation ψ and ϵ is a scalar that scales the magnitude of the perturbation. The condition necessary for optimality is

$$\frac{\partial}{\partial \epsilon}\hat{F}(Y+\epsilon\psi)|_{\epsilon=0} = 0, \tag{12}$$

for all choices of finite perturbation ψ.

We also require that

$$\frac{\partial^2}{\partial \epsilon^2}\hat{F}(Y+\epsilon\psi)|_{\epsilon=0} > 0, \tag{13}$$

for all choices of finite perturbation ψ. When the Hessian in (13) is equal to zero, bifurcation occurs and the algorithm increases the cardinality of the solution set.

Using (11) and differentiating the Free Energy term with respect to ϵ twice and with further algebra we get

$$\begin{aligned}&\hat{F}(Y+\epsilon\psi)\\ &\quad = -\frac{1}{\beta}\sum_i\left[p(x_i)\log\sum_l\left[e^{-\beta d(x_i,z_l(Y+\epsilon\psi))}\right]\right],\end{aligned} \tag{14}$$

$$\begin{aligned}&\Rightarrow \frac{\partial}{\partial\epsilon}(\hat{F}(Y+\epsilon\psi))\\ &= 2\sum_{i,l}p(x_i)[p(z_l^{\epsilon}|x_i)[Q_i'+(Y'+\epsilon\psi')\Upsilon_l']\Upsilon_l\psi],\end{aligned} \tag{15}$$

$$\begin{aligned}&\Rightarrow \lim_{\epsilon\to 0}\frac{\partial^2}{\partial\epsilon^2}(\hat{F}(Y+\epsilon\psi))\\ &= 4\beta\sum_i p(x_i)\left[\sum_q p(z_q|x_i)\rceil'_{iq}\Upsilon_q\psi\right]^2\\ &\quad + 2\sum_{i,l}[p(x_i)p(z_l|x_i)\psi'\Upsilon_l'(I-2\beta\rceil_{il}\rceil'_{il})\Upsilon_l\psi],\end{aligned} \tag{16}$$

where $\rceil_{il} = [Q_i + \Upsilon_l Y]$. (This derivation is analogous to the DA algorithm [7]). Note that the first term of the Hessian in (16) is nonnegative. However, we can show that whenever the second term is nonpositive, there exists a direction ψ such that the first term is zero (see the Appendix, available in the online supplemental material). This implies that the positivity of the right hand side of (16) for all directions ψ is determined by the positive definiteness of $(I-2\beta\rceil_{il}\rceil'_{il})$, which gives us the condition for critical temperature as follows:

$$\det[I-2\beta\rceil_{il}\rceil'_{il}] = 0 \Rightarrow T_c = \frac{1}{\beta_c} = 2\lambda_{\max}^{\rceil_l}, \tag{17}$$

where $\lambda_{\max}^{\rceil_l}$ is the largest eigenvalue of $\rceil_{il}\rceil'_{il}$. (Note that this critical temperature value is calculated for each distinct route l, and different coincident route vectors might split at different temperatures.)

4 Combined Locational and Routing Optimization of Multihop Network with Multiple Capabilities and Constraints

An important feature of the proposed algorithm is that it is flexible and can easily accommodate constraints on network topology as well as constraints on resources. We demonstrate this flexibility through the following scenarios.

4.1 Structured (Partially Connected) Networks

In the baseline algorithm described above, we have assumed a fully connected network topology (that is all resource nodes can communicate with each other). However, in certain applications, the underlying networks are partially connected. For instance, consider the case where all communication nodes (say of different makes) cannot talk to each other. The above baseline algorithm can be easily adapted to this case by embedding the partially connected graph into a fully connected graph and introducing a binary *connectivity* parameter $\lambda_l \in \{0,1\}$ for each route in the fully connected graph. Thus, when the connectivity parameter $\lambda_l = 1$ (or 0), it implies that the lth route in the fully connected network is (or is not) a route in the partially connected network. For instance by assigning the value 0 to each edge of the fully connected graph that is absent in the partially connected graph, and assigning the value 1 to each link that is present, we can determine the corresponding λ_l parameter for the lth route by multiplying the constituent edge values (Fig. 4).

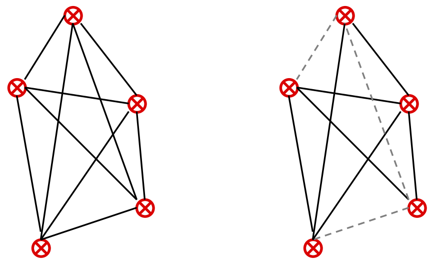


Fig. 4. The network on the left of the figure is an all-connected network topology of five resource nodes. The network on the right has certain links and routes disconnected (depicted by dotted lines) and is a partially connected network topology. The partially connected network can be thought of as the all-connected network with the connectivity parameter $\lambda_l = 0$ for all routes that contain the dotted links and $\lambda_l = 1$ for those routes that do not.

The connectivity is incorporated by scaling the terms corresponding to route z_l by λ_l in the weight $p(z_l|x_i)$ by which yields

$$p(z_l|x_i) = \frac{\lambda_l e^{-\beta d(x_i,z_l)}}{\sum_k \lambda_k e^{-\beta d(x_i,z_k)}}, \tag{18}$$

where $\lambda_l \in \{0,1\}$. Thus, we can eliminate infeasible routes by choosing the corresponding connectivity parameter to be zero. This could also be done by defining only feasible routes, but the above parametrization allows a more flexible implementation by allowing changes to the network topology to be reflected directly as change in the parameter values without requiring a change in the variable set $\{z_l\}$.

4.2 Resource Constrained Multilayered Structured Network

In certain applications, different routes are needed to be assigned different weights. This weight prioritization is accommodated by our algorithm through an additional term μ_l for each route l similar to the connectivity parameter introduced above. We consider the following characteristics that distinguish routes.

4.2.1 Number of Hops Priority

There is a distinct categorization of the routes based upon the number of hops a message takes when using that route. Often with each extra hop, there are associated penalties and overheads like processing energy cost of the extra resource node used and time delay while the message is being processed by the resource node. This motivates the need to make the framework flexible enough to accommodate prioritization of the routes according to the number of hops it contains. We accommodate this prioritization through a *hops-priority* parameter μ_{al} that premultiplies each term in the Gibbs distribution in the same manner as the connectivity parameter was incorporated in (18). Here, μ_{al} represents the relative weight of the route l. For instance, if the relative weight to be assigned for routes with n-hops is h_n, then the weight μ_{al} is given by $\mu_{al} = h_m$ if z_l is a route with m-hops.

4.2.2 Weakest Resource Node Constraint

Each route z_l uses one or more resource nodes y_j for communication. The weight $p(z_l) = \sum_i p(x_i)p(z_l|x_i)$ gives the relative usage of a route z_l, and $\sum_{l, y_j \in z_l} p(z_l)$ gives the relative usage of a resource node y_j. In certain applications, the relative weights w_j of the communication nodes are known a priori (for instance say that relative communication load that resource node y_j can handle is w_j). In order to constrain the use of a particular resource node y_j, we need to constrain all the routes that use that resource node. The constraint on each route is dictated by the weakest of all the resource nodes that are used by the route. We shall call this constraining parameter as the *weakest resource node* parameter $\mu_{bl} = \min_{y_j \in z_l}\{w_j\}$.

4.2.3 Weakest Link Constraint

Similarly, in wireless communication, the routes are communication channels that do not always offer complete connectivity, which constrain certain routes. We consider each link (or edge) as a pair of resource nodes $\zeta_h = \{y_f, y_g\} : f \neq g$, and ascribe weights v_h to each link of the network topology. The constraint on each route is dictated by the weakest link of the route. We call this constraining parameter as the *weakest link* parameter $\mu_{cl} = \min\{v_h\}$ such that $v_h = \{y_{l_k}\}, y_{l_{k+1}}$, where y_{l_k} and $y_{l_{k+1}} \in z_l$.

All three parameters defined above contribute to declaring a constraint on each route based on the priority in links of different hops, the weakest node, and the weakest link in the route. We combine the three parameters into a single parameter called the *route-weight* parameter $\mu_l = \mu_{al}\mu_{bl}\mu_{cl}$ which is used to prescale and then normalize the weight distribution $p(z_j|x_i)$ at each iteration. Accordingly the free energy gets modified as

$$F^* = -\frac{1}{\beta}\sum_i p(x_i) \log \sum_l \lambda_l \mu_l e^{-\beta d(x_i, z_l)},$$

where λ_l is chosen as 1 or 0 depending on whether the z_l route in the fully connected graph, respectively, does or does not exist. The resource locations y_j are found by setting $\frac{\partial F^*}{\partial Y} = 0$. The resource locations are given by the same expression as in (11) except that $p(z_l, x_i)$ is given by

$$p(z_l, x_i) = \frac{p(x_i)\lambda_l \mu_l e^{-\beta d(x_i, z_l)}}{\sum_k \lambda_k \mu_k e^{-\beta d(x_i, z_k)}}. \tag{19}$$

Remark. In the above development the *route-weights* μ_l were prespecified and known a priori through given values of μ_{al}, μ_{bl}, and μ_{cl}. In some applications, the weights μ_l are not known a priori but need to be determined. For instance, suppose in a network, we represent that amount of data carried by route l by μ_l and want to determine the location of resource nodes y_j under the constraint that $\sum_l \mu_l = 1$. That is, in addition to determining resource node locations y_j, we need to determine *route-weights* μ_l such that the total communication load is as given. This can be easily incorporated by minimizing the following Lagrangian with respect to $\{y_j\}$ as well as μ_l:

$$\begin{aligned} F^* = &-\frac{1}{\beta}\sum_i p(x_i) \log \sum_l \lambda_l \mu_l e^{-\beta d(x_i, z_l)} \\ &+ \nu\left\{\sum_l \mu_l - 1\right\}. \end{aligned} \tag{20}$$

Again the resource locations come out be the same expression as in (11) except that $p(z_l|x_i)$ is given by (19). Thus, prescaling the weight distribution $p(z_l|x_i)$ does not change the form of the equations that follow (This scaling factor can be imagined to be generated by multiple instances of the same route).

The route weights μ_l are determined by setting $\frac{\partial F^*}{\partial \mu_l} = 0$

$$\Rightarrow -\frac{1}{\beta\mu_l}\sum_i p(x_i)p(z_l|x_i) + \nu = 0 \Rightarrow \mu_l = \frac{p(z_l)}{\beta\nu},$$

but $\sum_l \mu_l = 1$; therefore,

$$\frac{\sum_l p(z_l)}{\beta\nu} = 1 \Rightarrow \nu = \frac{1}{\beta} \text{ and hence } \mu_l = p(z_l), \tag{21}$$

where $p(z_l) = \sum_i p(x_i)p(z_l|x_i)$.

4.3 Multichannel Resource Locational Optimization

The baseline algorithms can also be adapted to problems that have multiple channels for each route. That is, those problems where resource nodes have multiple channels and the cost of using different channels can be different. For instance, this problem setting can accommodate a reconnaissance scenario where communication units can interact over multiple frequencies (channels), where high-cost channels carry larger bandwidth or have larger range. The problem in this scenario is to find a channel from each agent to the destination node that will minimize the communication cost, thus, posing a trade-off between cost against bandwidth or range.

Consider the communication network to have C distinct channels. Let χ_{lc} denotes a design parameter that gives a measure of the usage of cth channel ($1 \leq c \leq C$) by the lth route, and $w_c, 1 \leq c \leq C$ denote the bound on the extent of use of the cth channel; then the corresponding multichannel resource allocation problem is

$$\min_{Y \in \mathbb{R}^M} \sum_{i=1}^N p(x_i) \min_{\{z_{lc}\}} d(x_i, z_{lc})$$
$$\text{such that} \quad \sum_l \chi_{lc} = w_c,$$

where x_i, $p(x_i)$, Y, and z_l denote, respectively, element locations, element weights, resource node locations, and routes as in the baseline algorithm, z_{lc} denotes the cth channel in the lth route, and $d(x_i, z_{lc})$ represents the cost of using the cth channel along the lth route for an element x_i to communicate to the destination. For instance, we choose $d(x_i, z_{lc}) = (1 + \theta_c)d(x_i, z_l)$, where $\theta_c > -1$ is the channel cost parameter and $d(x_i, z_l)$ is same as in (3). We use the same procedure as we did to the baseline problem and introduce the weights $p(z_{lc}|x_i)$ and solve for

$$\min F = D - TH,$$

under the channel constraints, where

$$D = \sum_{i=1}^{N} p(x_i)p(z_{lc}|x_i)d(x_i, z_{lc}),$$

denotes the distortion and H represents the entropy of weights $p(z_{lc}|x_i)$.

The corresponding free energy term is given by

$$F^* = -\frac{1}{\beta}\sum_i p(x_i)\log\sum_{l,c}\chi_{lc}e^{-\beta d(x_i,z_{lc})} + \sum_c \nu_c\left\{\sum_l \chi_{lc} - w_c\right\}. \tag{22}$$

The resource locations are given by the same expression as in (11) except that $p(z_l|x_i)$ is given by

$$p(z_l|x_i) = \sum_c p(z_{lc}|x_i) = \sum_c \frac{\chi_{lc}e^{-\beta d(x_i,z_{lc})}}{\sum_k \chi_{lc}e^{-\beta d(x_i,z_{kc})}}. \tag{23}$$

The channel weights χ_{lc} are determined by setting $\frac{\partial F^*}{\partial \chi_{lc}} = 0$

$$\Rightarrow -\frac{1}{\beta\mu_l}\sum_i p(x_i)p(z_{lc}|x_i) + \nu_c = 0 \Rightarrow \chi_{lc} = \frac{p(z_{lc})}{\beta\nu_c},$$

but $\sum_l \chi_{lc} = w_c$; therefore,

$$\nu_c = \frac{\sum_l p(z_{lc})}{\beta w_c} \text{ and hence } \chi_{lc} = \frac{p(z_{lc})w_c}{\sum_l p(z_{lc})}, \tag{24}$$

where $p(z_{lc}) = \sum_i p(x_i)p(z_{lc}|x_i)$.

The iterative scheme presented in this section is implemented as follows:

- Step 1. Start at a high value of T (low β) and initialize all resource nodes $\{y_j\}$ at the weighted centroid of the elements $\{x_i\}$, i.e., $y_j = \sum_i p(x_i)x_i$. Assign $\lambda_l = 1$ or 0 depending on the resource node network.
- Step 2. Set $\mu_l = \mu_{al}\mu_{bl}\mu_{cl}$ or by $\mu_l = p(z_l)$ in (21) depending on the scenario.
- Step 3. Compute the probability distributions $p(z_l|x_i)$ using (19).
- Step 4. Compute new resource node locations $\{y_j\}$ using (11).
- Step 5. Increase β; Stop if β reaches a user-defined maximum value else goto Step 2.

5 Simulations and Results

For simulations, we use a data set of 150 elements $\{x_i\}$, formed by randomly selecting 15 locations spread over 300×300 area as centers of clusters and a normal distribution of 10 points was generated around each of these centers with random (uniformly distributed) variance. The destination-center location is set at $[250, 250]$ and five resource nodes $\{y_j\}$ at random locations are chosen. The algorithm was encoded using MATLAB and the

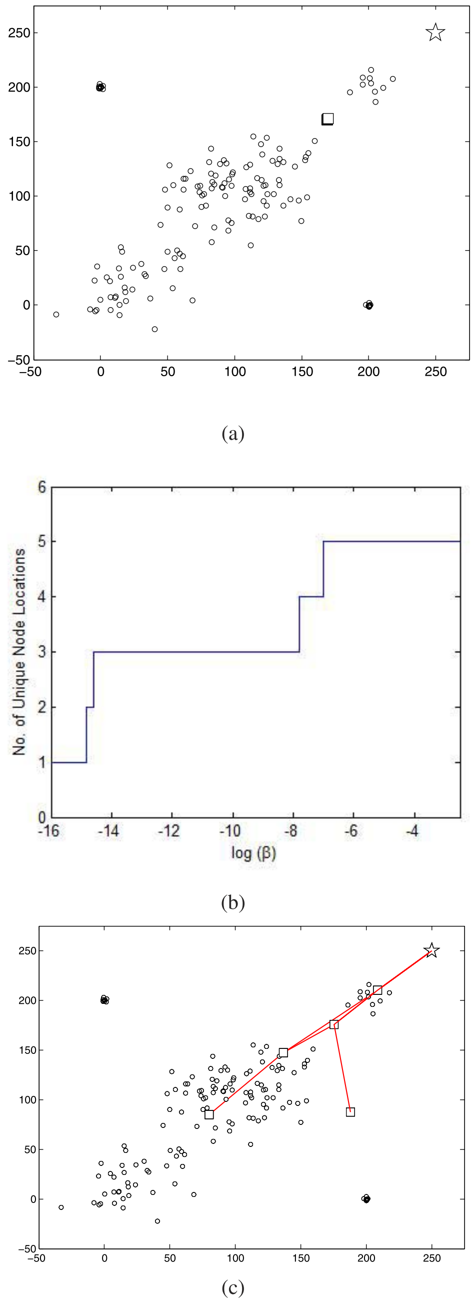


Fig. 5. (a) Above depicts the relocation of resources (hollow squares) at the centroid of the data set at low values of β. (b) Plots the number of distinct resource locations as a function of β. (c) Depicts the solution resource node locations at the final value of β.

simulation was conducted on an Intel Core 2 Duo T5470 1.6 GHz processor with 2 GB RAM. At low values of β, all resource nodes relocate themselves at the centroid of the data set (Fig. 5a). As the algorithm progresses and the value of β increases, the number of distinct resource increases (Fig. 5b). As the final value of β (the user-defined maximum value chosen as a stopping criterion for the algorithm) is reached, the solution resource node locations is obtained (Fig. 5c). Fig. 7a illustrates the final resource node locations for the single-hop scenario. It should be noted that as the value of β increases in a single-hop scenario, the resource nodes split up and their positions spread out between the destination center and clusters of elements. Figs. 7b and 7c illustrate the final resource node

TABLE 1
This Table Details the Performance of the Algorithm

Detail	single-hop	up-to-2-hops	up-to-3-hops	up-to-4-hops	up-to-5-hops
Communication Cost (basic)	29628.4173	21334.9742	17169.9097	14737.3356	13987.9972
Communication Cost (with μ_l)	29628.4173	21679.5668	16872.9539	14737.3356	13987.9972
% of 1-hop cost	100.00	73.17	56.95	49.74	47.21
% of two-step algorithm cost	150.97	110.74	89.12	76.49	72.61
Computational time (s)	7.6739	8.5917	10.4019	16.3858	24.8600
% of two-step computational time	77.37	86.62	104.87	165.20	250.63

The algorithm for multiple-hop scenarios are compared with the single-hop scenario and the two-step algorithm described in Section 3. The cost for the two-step algorithm is 19,264.656 units in 9.9189 seconds. The communication costs decrease as the number of hops are increased and the computational times correspondingly increase. This trade-off is recorded in this table.

locations for scenarios that allow up to three hops and up to five hops, respectively. In such multihop scenarios the resource nodes tend to diverge less and position themselves approximately collinear with the destination-center location, forming "bridges" for the hops to occur. The results obtained on running the algorithm for a single-hop-only to up-to-five-hops scenario are presented in Table 1. The scenario cost is computed using the distortion given by (6). The cost in row 1 is the cost of running the baseline algorithm as described in Section 3. The cost in the row 2 incorporates the *route-weight* parameter μ_l where the parameter values are not known a priori as described in Section 4.2. The computational time in seconds is tabulated in row 4. We provide a comparison of communication costs and computational times of the multiple-hop scenarios with the single-hop scenario, where the latter is a purely resource locational optimization problem without any routing considerations (see rows 3 and 5). Note that the costs in the multiple-hop scenarios are lesser than the cost in the single-hop scenario. For the example considered, the cost of the up-to-five-hops scenario is 47.21 percent of the single-hop scenario cost, while the computational time required for the former is about thrice that of the latter. Another benchmark that we use to compare our algorithm is the cost of the "two-step algorithm" described in Section 3 (see rows 4 and 6 in the table). As, expected, this benchmark outperforms the single-hop and up-to-two-hops scenarios, and performs worse than greater than up-to-three-hops scenarios. For instance, the up-to-five-hops scenario achieves only 72.61 percent of the cost for the two-step algorithm in less than thrice the computational time.

Fig. 6 studies the sensitivity of the cost benefit of multihop over single-hop scenario to the destination-center location r. The given area was divided into a 10×10 square grid with the center of a grid-cell chosen as the destination-center location. The algorithm was run for every new location of the destination-center, thus yielding data for 100 runs. The distance of the destination-center location from the centroid of the elements $\{x_i\}$ was noted for each simulation run and the results plotted in Fig. 6. Observe that as the distance between the destination-center location and the centroid of the elements increases, the percentage reduction in cost in a multihop scenario over single-hop scenario increases. This can be explained as follows: For a centroidal destination-center location, the data set can be partitioned into clusters that are located all around the centroid. Placing a resource node between the centroid and each cluster leads to a lower communication cost. In this resource node configuration, single-hop communication dominates over multihop. This diminishes the benefits offered by including a multihop routing scenario in the locational optimization algorithm. For a destination-center location away from the centroid of the data set, the advantages of multihop communication over single-hop become significant. With some clusters of elements located far away from the destination-center location, single-hop communication is very expensive. A relay of resource nodes located at intermediate positions acting like "bridges" in a multihop scenario reduce the cost substantially. The distribution of data points in the plot in Fig. 6 is substantially linear.

In Fig. 9, we show the incorporation of the connectivity parameters to enable scenarios with partially connected networks. Fig. 9a that allows 5 hops (the same scenario as in Fig. 7). The details of the solution are depicted in Table 2. Note that more than half (57.63 percent) of the total communication load is carried by the 4-hop route $b \rightarrow c \rightarrow d \rightarrow e$. Fig. 9b shows results for a partially connected network in which all 4-hop routes are disabled. The corresponding details are depicted in Table 3. Note that the communication cost deteriorates over the fully connected scenario, but only by 4.72 percent, even when all the 4-hop routes are disabled (which includes the one that carried more than half the communication load in the previous scenario). The algorithm achieves this by aptly rearranging the resource node locations (by bringing them closer to the destination center) so that the communication

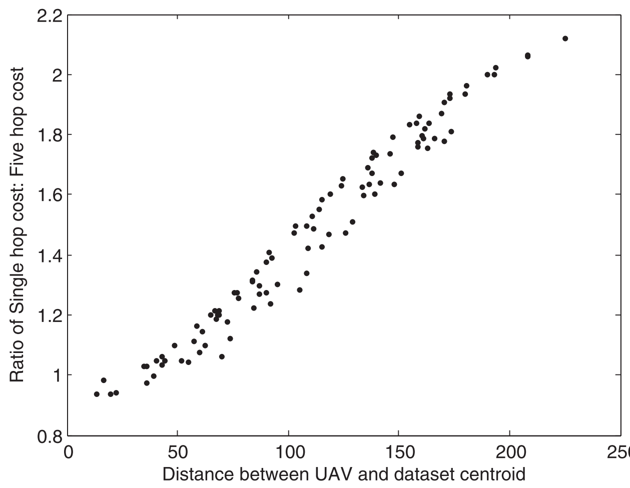


Fig. 6. Ratio of single-hop scenario cost to up-to five-hops costs allowed scenario as a function of the distance between the destination center and the centroid of the data set. Note that the cost benefit of allowing more hops is directly proportional to the distance.

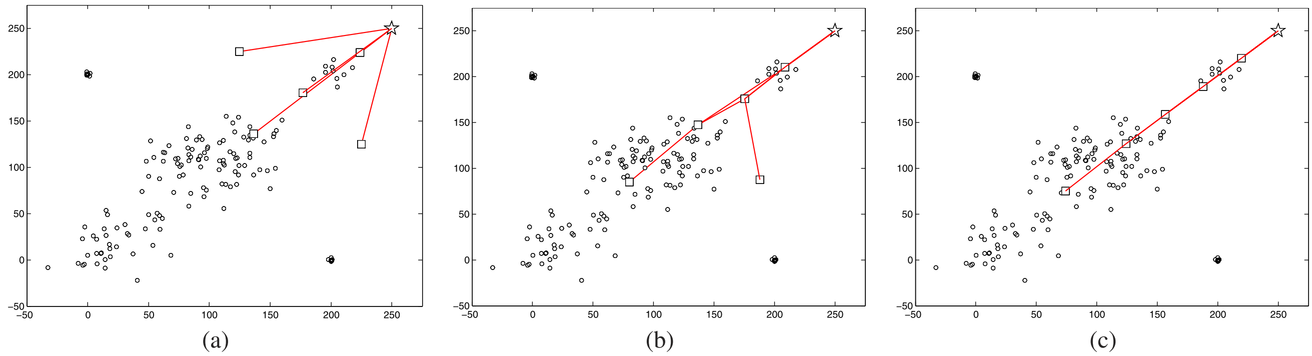


Fig. 7. Each figure in the above series depicts the final location of resource nodes (black squares) and routes, made of hops (red line segments), with each route originating at a resource node and terminating at the destination center (black star). The (a), (b), and (c), respectively, depict scenarios which allow only single-hop, up-to-three-hops, and up-to-five-hop communication. Note the tendency of the resource nodes to spread out around the centroid of the data set in single-hop scenario and to form "bridges" in scenarios allowing more than one hop.

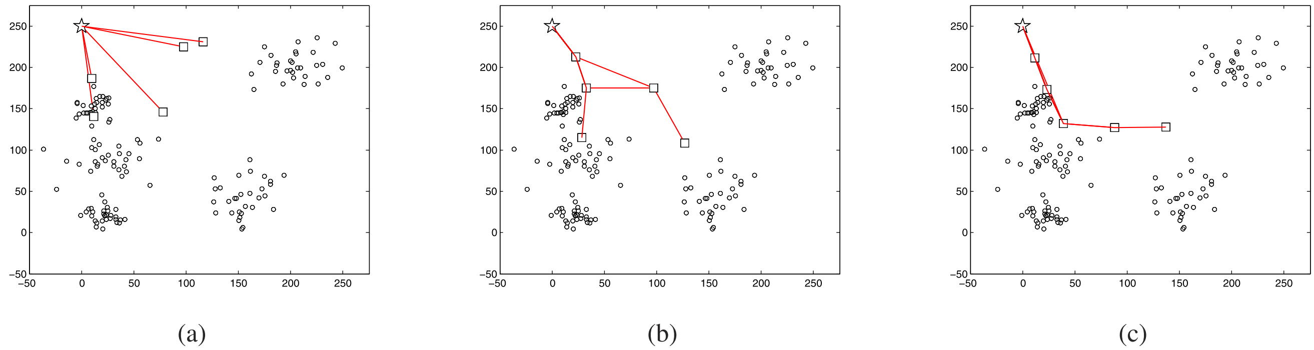


Fig. 8. Each figure in the above series depicts the final location of resource nodes (black squares) and routes, made of hops (red line segments), with each route originating at a resource node and terminating at the destination center (black star) for a different set of elements X_i and a different destination-center location. The (a), (b), and (c), respectively, depict scenarios which allow only single-hop, up-to-three-hops, and up-to-five-hops communication. Note, the tendency of the resource nodes to spread out around the centroid of the data set in a single-hop scenario and to form "bridges" in scenarios allowing more than one hop.

load previously carried by the 4-hop route is now mainly distributed to a 5-hop and a 3-hop route.

Table 4 tabulates the results of the multichannel resource locational optimization (discussed in Section 4.3) simulation. Three distinct channels are overlaid on each route and their relative weights are given by w_c. For the same relative weights, we implement two different sets of channel costs θ_c and tabulate the probability distributions of the channels which give the channel usage. The constraints considered are opposing in nature; The channel with higher weight (channel 3) is given higher cost whereas the channel with lower weight (channel 1) is given lower cost, thus leading to usage probabilities of the same order. As can be noted, the usage of a channel decreases as the relative channel cost increases.

We have performed simulations over larger data sets (over a magnitude higher number of elements and resource nodes), for a variety of different scenarios (Fig. 8 illustrates one such different scenario for a different set of elements X_i and a different destination-center location), and similar results are derived for all these cases. We have presented only the data set above for clarity in illustration as well as lack of space to present all the results.

6 Analysis and Discussion

6.1 Flexibility of the Algorithm

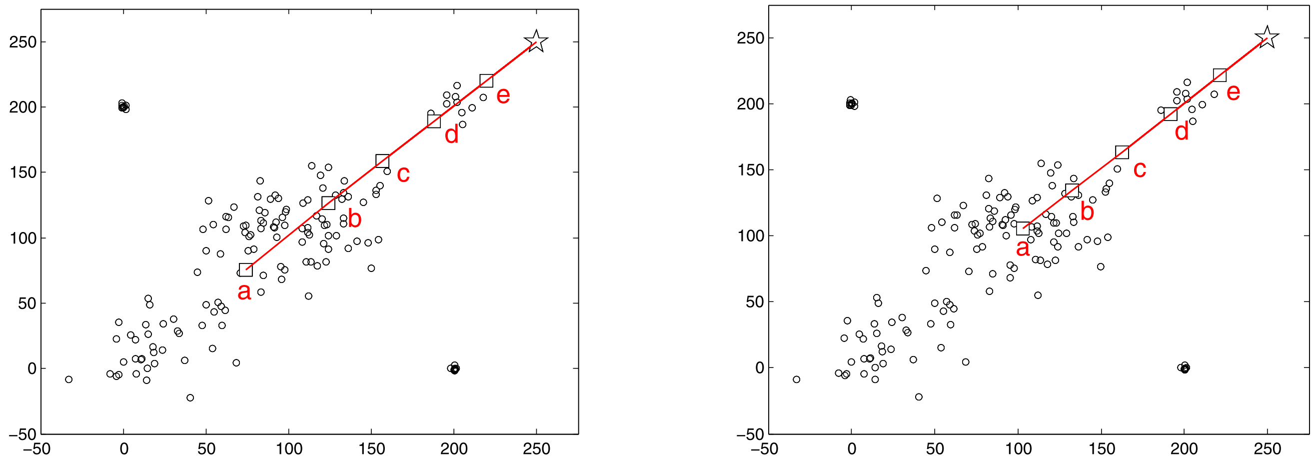
The algorithm is shown to be flexible and modifications were made to the algorithm to implement different relevant subscenarios at relatively lower analytical and computational cost addition. As described in Section 1, numerous applications, for example, in areas of wireless sensor networks, mobile communication and transportation, pose similar locational optimization and routing problems. However, the details of these problems differ in the specific constraints on network topology or the resources. The flexibility of the algorithm in terms of the distance (cost) function as well as adding additional

TABLE 2
Route Communication Load Distribution in a Fully Connected Network Scenario

Route	% of total communication load
e	6.89
c → d → e	8.49
b →c →d →e	57.63
a →b →c →d →e	26.99

TABLE 3
Route Communication Load Distribution in a Partially Connected Network

Route	% of total communication load
e	6.49
d → e	0.41
c →d →e	11.95
a →b →c →d →e	81.16



(a) (b)

Fig. 9. Each figure in the above series depicts the final location of resource nodes (black squares) and routes, made of hops (red line segments), with each route originating at a resource node and terminating at the destination center (black star). In (a) implementation of a fully connected network with up-to-five-hops is presented while (b) presents results for a partially connected network with up-to-five-hops allowed, where all 4-hop routes disabled.

constraints to the free energy term easily accommodates problem-specific constraints. Section 4 elaborated on the inclusion of different constraints to implement various modifications in the basic scenario outlined in Section 2.

This framework has the flexibility to include a variety of problems. For instance, in this paper our model included a single destination center at location "r." One can easily add another network layer to this framework by incorporating multiple destination centers (and determining their locations r_k) that in turn communicate to a common control center. Other avenues of generalization are in terms of having different distance functions as opposed to squared-euclidean distances considered in this paper and higher dimensional domains in contrast to the two-dimensional domain Ω considered in this paper. With these modifications, the underlying formulation remains the same albeit with some algebraic/computational complications.

6.2 Scalability

The inclusion of entropy term to make the algorithm insensitive to the initial choice of routes results in the computations being global in the sense that computing the location of each resource node requires values of all the elements. However, the contribution of "far off" elements becomes progressively lower as the parameter β is increased. In fact, the partitions are hard as $\beta \to \infty$ and consequently the contribution of site locations that are not "nearest neighbors" is zero. In this sense, computing the resource locations changes from being *truly global* to *truly local* as β is increased from zero to infinity. In [3], this feature, in the context of locational optimization, is exploited to evolve a scalable algorithm that is a close approximation of the deterministic annealing algorithm. We propose the following modification that will make the proposed algorithm scalable.

6.2.1 Numerical Round-Off

At every iteration of the parameter β, the computations are global in the sense that the calculation of lth route is determined by *all* the element locations x_i in Ω. However, as the algorithm progresses with higher values of β, the probability distributions $p(z_l|x_i)$ take approximately binary values 0 or 1 (and indeed take values 0 or 1 at $\beta = \infty$). We exploit this property to make the algorithm scalable. We choose a threshold $\delta > 0$ and all locations x_i that satisfy $d(z_l, x_i) < \delta$ are ignored in the computation of z_l. This saves computational time thereby making the algorithm scalable. Since small values of $d(z_l, x_i)$ imply small values of $p(z_l|x_i)$, the truncated terms do not have a significant effect on the calculation of z_l in the subsequent steps of the algorithm. The choice of δ as well as the effect of truncation on route computations is part of our ongoing work.

6.2.2 Markovian Structure

One important point to note is that the number of design variables (routes) in the presented algorithm is large given by $\sum_{k=1}^{M} {}^{M}P_k$ which evidently increases combinatorially with the number of resource nodes M (Here, ${}^{M}P_k$ represents the number of possible permutations in arranging k letters from an alphabet of length M). This leads to a large number of computations even though the presented algorithm is significantly better than the divide-and-conquer strategies.

TABLE 4
Multichannel Simulation Results for Three Channels

Channel	Relative weights w_c	Channel Cost 1 $(\theta_c)_1$	Channel Cost 2 $(\theta_c)_2$	Probability Distribution 1	Probability distribution 2
Channel 1	1	-10^{-3}	-10^{-3}	0.1445	0.2630
Channel 2	5	0	0	0.2469	0.6615
Channel 3	20	10^{-3}	10^{-2}	0.6087	0.0755

In the previous section, we have proposed ways to decrease the number of computations for a given M. We can potentially decrease the computations even further by using different models that describe the communication flow. For instance, the routing decisions can be arrived at by posing the communication as a Markov process, where the decision of a hop is dependant only on the current node location of the communication packet and is independent of the route it has taken to arrive to the current resource node. This formulation reduces the number of design variables significantly where it grows only as a polynomial function of M.

7 Conclusions

In this paper, we presented an algorithm for combined locational and multihop routing optimization. The proposed approach makes solving the combined node location and routing problem feasible and simulation results on test cases show near-optimal results. It offers larger flexibility for the algorithm to move on the solution space, thus offering a better chance at tracking the global minimum. We incorporated additional parameters to implement different scenarios maintaining the basic skeleton of the algorithm. The efficiency of the algorithm was presented in the simulation results, with quantitative comparisons to purely locational optimization with single-hop routing results.

Acknowledgments

This work was partially supported by US National Science Foundation ECS 0449310 CAR and Sprague/Swanson grants from UI at Urbana-Champaign.

References

[1] P. Sharma, S. Salapaka, and C. Beck, "A Maximum Entropy Based Scalable Algorithm for Resource Allocation Problems," *Proc. Am. Control Conf.*, pp. 516-521, 2007.

[2] P. Sharma, S. Salapaka, and C. Beck, "A Scalable Deterministic Annealing Algorithm for Resource Allocation Problems," *Proc. Am. Control Conf.*, pp. 3092-3097, 2006.

[3] P. Sharma, S. Salapaka, and C. Beck, "A Scalable Approach to Combinatorial Library Design for Drug Discovery," *J. Chemical Information and Modeling*, vol. 48, no. 1, pp. 27-41, 2008.

[4] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks: Variations on a Theme," *Proc. Mediterranean Conf. Control and Automation*, July 2002.

[5] S. Salapaka and A. Khalak, "Locational Optimization Problems with Constraints on Resources," *Proc. 41st Allerton Conf.*, pp. 1240-1249, Oct. 2003.

[6] S. Salapaka and A. Khalak, "Constraints on Locational Optimization Problems," *Proc. IEEE Control and Decision Conf.*, pp. 1741-1746, Dec. 2003.

[7] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210-39, Nov. 1998.

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

[9] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129-137, Mar. 1982.

[10] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, first ed. Kluwer, 1991.

[11] S. Kirkpatrick, C. Gelatt, and M. Vechhi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.

[12] J.N. Al-Karaki and A.E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Comm.*, vol. 11, no. 6, pp. 6-27, Dec. 2004.

[13] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 16, no. 1, pp. 51-62, Feb. 2008.

[14] T. Moscibroda, M. Wattenhofer, R. O'Dell, and R. Wattenhofer, "Virtual Coordinates for Ad Hoc and Sensor Networks," *Proc. Joint Workshop Foundations of Mobile Computing (DIALM-POMC '04)*, pp. 8-16, 2004.

[15] N.S. Kulkarnit, B. Ramantt, and I. Gupta, "On Demand Routing Protocols for Mobile Ad Hoc Networks: A Review," *Proc. IEEE Int'l Advance Computing Conf. (IACC '09)*, pp. 586-591, 2009.

[16] C. Liu and J. Wu, "Efficient Adaptive Routing in Delay Tolerant Networks," *Proc. IEEE Int'l Conf. Comm.*, 2009.

[17] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, Nov. 1984.

[18] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The Planar k-Means Problem Is np-Hard," *Proc. the Third Int'l Workshop Algorithms and Computation (WALCOM '09)*, pp. 274-285, 2009.

[19] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-Hardness of Euclidean Sum-of-Squares Clustering," *Machine Learning*, vol. 75, no. 2, pp. 245-248, 2009.

[20] P. Drineas, A. Friexe, R. Kannan, S. Vempala, and V. Vinay, "Clustering Large Graphs via the Singular Value Decomposition," *Machine Learning*, vol. 56, pp. 9-33, 2004.

[21] E.T. Jaynes, "Information Theory and Statistical Mechanics," *Physical Rev.*, vol. 106, pp. 620-630, 1957.

[22] E.T. Jaynes, *Probability Theory—The Logic of Science.* Cambridge Univ., 2003.

[23] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication.* Univ. of Illinois, 1949.

[24] L.D. Landau and E.M. Lifshitz, *Statistical Physics, Part 1*, third ed., vol. 3. Oxford, 1980.

Nachiket V. Kale received the BTech and MTech degrees in electrical engineering, communications, and signal processing from the Indian Institute of Technology, Bombay (IIT Bombay), India, in 2007 and the MS degree in aerospace engineering from the University of Illinois at Urbana-Champaign (UIUC). His masters' research includes wireless sensor networks and optimization. He is currently a doctoral candidate at the Department of Aerospace Engineering at UIUC. His current research interests include aerodynamics and flow control.

Srinivasa M. Salapaka received the BTech degree in mechanical engineering from the Indian Institute of Technology, Chennai, in 1995 and the MS and PhD degrees in mechanical engineering from the University of California, Santa Barbara, in 1997 and 2002, respectively. From 2002 to 2004, he was a postdoctoral associate in the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge. Since 2004, he has been a faculty member in the Mechanical Science and Engineering Department, University of Illinois, Urbana-Champaign. His areas of current research interest include controls for nanotechnology, combinatorial resource allocation, and numerical analysis. He received the National Science Foundation CAREER Award in 2005. He is a member of the IEEE.