

Self-Organization of Sensor Networks with Heterogeneous Connectivity

Arun Prasath, Abhinay Venuturumilli, Aravind Ranganathan and Ali A. Minai

Abstract Most research on wireless sensor networks has focused on homogeneous networks where all nodes have identical transmission ranges. However, heterogeneous networks, where nodes have different transmission ranges, are potentially much more efficient. In this chapter, we study how heterogeneous networks can be configured by distributed self-organization algorithms where each node selects its own transmission range based on local information. We define a specific performance function, and show empirically that self-organization based on local information produces networks that are close to optimal, and that including more information provides only marginal benefit. We also investigate whether the quality of networks configured by self-organization results from their generic connectivity distribution (as is argued for scale-free networks) or from their specific pattern of heterogeneous connectivity, finding the latter to be the case. The study confirms that heterogeneous networks outperform homogeneous ones, though with randomly deployed nodes, networks that seek homogeneous out-degree have an advantage over networks that simply use the same transmission range for all nodes. Finally, our simulation results show that highly optimized network configurations are as robust as non-optimized ones with respect to random node failure, but are much more susceptible to targeted attacks that preferentially remove nodes with the highest connectivity, confirming the trade-off between optimality and robustness postulated for optimized complex systems.

Arun Prasath

ECE Department, University of Cincinnati, Cincinnati, OH 45221 e-mail: news4arun@gmail.com

Abhinay Venuturumilli

ECE Department, University of Cincinnati, Cincinnati, OH 45221 (Currently at Microchip Technology, Chandler, AZ 85224, USA) e-mail: abhinay.venuturumilli@microchip.com

Aravind Ranganathan

CS Department, University of Cincinnati, Cincinnati, OH 45221 e-mail: rangana@email.uc.edu

Ali A. Minai

ECE Department, University of Cincinnati, Cincinnati, OH 45221 e-mail: Ali.Minai@uc.edu

1 Introduction

Self-organization is seen increasingly as an attractive alternative to design for engineering large-scale complex systems such as sensor networks, robot swarms, multi-agent systems, self-reconfiguring robots and smart structures [1]. As systems grow in size, self-organization provides an inherently scalable, flexible and robust way to obtain effective functionality without the need for global communication or control. However, most theoretical work on self-organization has focused on physical systems or their simplified models (e.g., cellular automata, percolation models, sandpiles, etc.), where the process is driven by simple rules that have little to do with domain-dependent performance considerations. In contrast, systems for engineering applications must accomplish goal-directed tasks, and their self-organization rules must be based on domain-specific considerations such as bandwidth, capacity, cost, energy resources, etc. Ultimately, the success (or failure) of such self-organization procedures must be judged by whether the resulting system has close to optimal *performance* — an issue seldom considered for abstract models such as cellular automata or sandpiles.

This is seen most clearly in the area of complex networks, where much of the work has focused on systems obtained through purely structural self-organization rules such as preferential attachment [2, 3, 4, 5] or rewiring [6, 5], and the resulting systems have been characterized primarily in terms of their global structural properties such as degree distribution or connectedness. Though functional attributes such as robustness have been studied extensively [7, 8], they are evaluated *post facto*, and not explicitly integrated into the self-organization algorithm itself. Even in studies that have systematically considered the optimization of functional properties [9, 10, 11, 12], the approach has been to *first* generate ensembles of networks using algorithms parameterized by generic quantities such as connectedness, attachment preference, degree distribution, etc., and *then* to characterize the results to find the best set of parameters (see [13] for an exception). However, as pointed out cogently by Doyle and coworkers [14, 15, 16, 17, 18], even the best networks produced in such cases are typically far from optimal with regard to specific application domains and problem instances. The central challenge for complex systems engineering [1] is to find application-specific self-organization algorithms that have the desirable attributes of their abstract counterparts but are based on concrete system properties [20, 21]. In particular, the goal is to find self-organization algorithms capable of producing systems with *optimally structured heterogeneity*, which is how custom-designed systems achieve their performance [16].

In this chapter, we consider a relatively simple but important class of systems — wireless sensor networks — and present a set of self-organization rules that try to optimize a specific, application-relevant performance criterion. We show empirically that the resulting networks are indeed close to optimal, that their performance derives from the *specific* structuring of their heterogeneity rather than from simple generic attributes, and that they represent *atypical* samples in the overall configuration space. These results are of broad interest in the context of complex networks (and, indeed, complex systems) because they partially bridge the divide between

purely open-loop self-organization [2, 3, 4, 5, 6, 5] and explicit design. This work demonstrates that, in a reasonably complex and practical network, it is possible to obtain most of the benefits of optimized design through self-organization based only on local information. While we focus only on geometric networks (i.e., networks where connectivity is based on geometric neighborhoods in physical space), the ideas may extend to other distributed systems such as swarms [19], where agents make decisions based on local information while seeking to achieve a globally desirable configuration.

A major issue of interest in the complex networks literature is the robustness of such systems to random node failure and targeted attack. Networks with power-law degree distribution are much more robust to random failure than random networks, but more susceptible to targeted attack [2, 22, 7, 23]. While it arises from inherent structure in scale-free networks, Carlson and Doyle have argued that this “robust-yet-fragile” attribute also characterizes all highly optimized systems, which achieve their enhanced performance by trading off robustness to likely problems against fragility to unlikely ones [14, 15, 16, 17]. We test this empirically on networks generated through explicit optimization (using genetic algorithms), networks produced through heuristic self-organization methods, and those obtained by applying simple uniform rules (e.g., identical out-degree for all nodes). Our results show that, in the case of geometric networks, all configurations lose fitness with both random failure and targeted attack. However, the non-optimized configurations are affected in the same way in both cases while the optimized configurations show very different responses. Specifically, the latter lose much more fitness under targeted attack than under random failure. These results are in line with the “robust-yet-fragile” view of optimized systems.

2 Background and Motivation

Geometric networks are networks where nodes are distributed in a metric space, and each node connects to all others within a certain range, called the *connection radius* of the node. Such networks arise naturally in the context of wireless networks (including sensor networks), but, to some degree, neuronal networks in the brain can also be considered geometric [24]. Three attributes are of particular interest for such networks:

1. *Connectedness*: It is usually essential to have global connectivity, i.e., have the network form a single connected component. For example, in wireless sensor networks, information is typically obtained locally and processed collaboratively, which requires connectedness.
2. *Network diameter*: This is the average length of the shortest paths between all node pairs — typically measured in the number of edges traversed (termed hops). Since each hop in a network usually entails some cost (e.g., transmission energy, wait time, etc.), a small diameter is considered good because it implies more efficient communication over longer distances.

It is usually more practical to use the *average inverse shortest path length* (*AISPL*), which averages the inverse distances, and can readily represent disconnected node pairs with a value of 0 [13].

3. *Mean in-degree*: The number of incoming edges to a node usually represent the “loading” faced by that node, and determines the capacity needed by the node to handle its load. Thus, smaller mean in-degree is considered a desirable attribute. For example, in wireless networks, the number of nodes that can be heard at the location of node i , and the mean in-degree of the network is a simple measure of congestion interference faced by that node.

In this chapter, we address the problem of obtaining connected geometric networks where the diameter and mean in-degree are minimized.

Most of the literature on geometric (typically wireless) networks has focused on *homogeneous networks*. These can be divided broadly into two classes: 1) *Degree-homogeneous (DH) networks*, where all nodes have the same out-degree, and 2) *Radius-homogeneous (RH) networks*, where all nodes have the same connection radius. It should be noted that each type of homogeneity (radius or degree) usually implies heterogeneity in the other parameter (degree or radius, respectively), but we reserve the term *heterogeneous networks* for those systems where neither radius nor degree are explicitly homogenized.

There is considerable research on degree-homogeneous networks — especially on determining the minimum number of neighbors necessary for a network to be connected [25, 26, 27, 28]. In particular, Xue and Kumar [28] show that for a network with N nodes, $C \log N$ neighbors for each node ensure connectivity, where $C > 1$. However, recent work indicates that such rules are very sensitive to inhomogeneities in node distribution [29]. In any case, choosing a fixed number of neighbors is a device for maintaining connectivity rather than minimizing network diameter or mean in-degree, which is our primary concern.

In radius-homogeneous networks, there is a critical radius above which the network is globally connected with high probability [30, 31]. This is called the *percolation radius* [32], ρ_{perc} . The simplest way to obtain a connected network with minimal mean in-degree is for all nodes to use a connection radius equal to (or just above) ρ_{perc} . However, this leads to a large diameter. Conversely, diameter can be minimized if all nodes use a radius much larger than ρ_{perc} , but this increases mean in-degree. Given the spatial distribution of nodes, one can find an “optimal” radius that represents the best compromise of diameter and mean in-degree, producing the optimal RH network. However, it has been shown that significantly better performance can be obtained by allowing heterogeneity, i.e., letting each node choose its own radius to maximize overall performance [33, 34, 35]. Unfortunately, assigning these radii is a very difficult combinatorial optimization problem in large systems, which has prompted several proposals for self-organized approaches [36, 37, 38, 39, 40, 41, 42, 43].

In the present chapter, we focus on the simplest possible class of heterogeneous networks, where only two radius choices are available — one smaller than ρ_{perc} and the other larger. The problem is to assign the appropriate value to each node such

that a specific function of diameter and mean in-degree is optimized. The key issues are:

- To show whether — and to what extent — this heterogeneity allows optimization beyond that offered by the homogeneous case.
- To propose efficient, scalable self-organization algorithms for configuring near-optimal heterogeneous networks, and validating their performance.

Networks with two classes of nodes differentiated by the spatial extent of their connectivity are interesting from both abstract and concrete perspectives. Since nodes in geometric networks must connect with all other nodes within their transmission radius, these networks cannot have selective “short-cut” links in the sense of small-world connectivity [6], and even obtaining power-law connectivity [22] becomes quite difficult and contrived. However, many interesting properties of small-world and power-law network models actually arise from the existence of *hub nodes* — highly connected nodes that shrink the network’s diameter and enhance its robustness [2, 22, 7, 23]. In the networks we study, the large radius nodes correspond to such hubs, and in this sense, these networks are the geometric analog of the classic complex network models. This chapter addresses the issue of whether they provide the same sort of advantages in terms of robustness and efficiency.

From a concrete viewpoint, two-level heterogeneous networks can be used to model many natural and artificial systems. For example, neuronal networks in the cerebral cortex consist of pyramidal cells with spatially wide-ranging connectivity and interneurons with more restricted range [44], and “hub-like” architectures in the cortex have been proposed as the crucial substrate of cognitive processes [45, 46]. In this chapter, we use wireless networks as the motivating case. Thus, the connection radius for a node is related to its transmission power, with larger radii requiring greater energy. Since nodes in wireless networks are typically energy-limited, there is considerable incentive to minimize diameter without increasing radii, i.e., transmission power, more than necessary [47, 48, 49, 50]. However, it should be noted that we do *not* explicitly try to minimize mean radius or total radius, and use the radii only as tunable parameters to minimize diameter and mean in-degree. We explicitly avoid specifying details such as handshake protocols, buffer sizes, etc., to maintain the generic character and scope of this study, which is intended to focus on the broad issue of self-organized network structures.

It should be noted that wireless networks with short-range and long-range nodes can arise naturally in many situations. For example, in chapter C3 of this volume [51], Cui et al. describe a network with short-range underwater nodes augmented by surface radio nodes with larger transmission radius.

3 System Description

We consider a system with N nodes distributed randomly in a unit square. Each node, i , has a connection radius r_i , and projects outgoing connections to every node

j within this radius. This defines the network $G(V, E)$, where V is the set of all nodes and E the set of all edges representing connections between nodes. Because nodes can have different radii, the network is heterogeneous, and the edges are directed. We assume that each node has a unique (randomly generated) identifier, but this is used only for local identification and there is no globally available lists of node IDs. We also assume that nodes are aware of their position in a global coordinate system, which may be based on GPS or self-organized within the network [52, 53, 54]. The position of node i is denoted by $l_i = (x_i, y_i)$, and the set $L \equiv \{l_i\}$, $i = 1, \dots, N$, is termed a *layout*. For simplicity, we assume that each node can take radius values from a finite set, $R = \{\rho_k\}$, $k = 1, \dots, M$. Thus, for a layout with N nodes, there are M^N possible assignments of radii. Each such assignment, $Q(L) \equiv \{r_i\}$, $i = 1, \dots, N$, is called a *configuration*, and corresponds to a specific graph, $G(V_Q, E_Q)$, where V_Q is the set of nodes in the underlying layout, L , and E_Q is the set of edges induced on this layout by Q (Here (and henceforth), we have omitted writing the argument L for notational clarity.) The in-degree of node i in configuration Q is denoted by $f_i^I(Q)$, and its out-degree by $f_i^O(Q)$.

The AISPL for a configuration, Q , is defined as:

$$H(Q) = \frac{1}{N(N-1)} \sum_{i,j \in V(Q)} 1/d_{ij}$$

where d_{ij} is the distance in hops from node i to j , and $1/d_{ij} = 0$ if there is no path from i to j . The mean congestion for a configuration Q is defined as:

$$C(Q) = \frac{1}{N} \sum_{i \in V(Q)} f_i^I(Q)$$

Given a layout, L , the goal is to find a configuration Q^* that maximizes the *fitness function*, $\phi = H(Q)/C(Q)$, thus trying to minimize congestion while keeping the network diameter as small as possible.

Whisperers and Shouters:

As discussed above, we focus in this chapter on the case of $M = 2$ (see [40, 41] for the $M > 2$ case.) Accordingly, we define two classes of nodes: 1) *Whisperers* are nodes with a small transmission radius, ρ_w , while *shouters* have a larger radius, ρ_s . To obtain a uniform parametrization, the two radii are expressed in terms of the nominal percolation radius, ρ_{perc} as $\rho_w = \alpha \rho_{perc}$ and $\rho_s = \beta \rho_{perc}$, where $\alpha < 1 < \beta$. The nominal ρ_{perc} is calculated as in [31]. Each node, i , maintains two adjacency lists for the nodes within its transmission range. Nodes within whisperer range of i are listed in its *whisper-adjacency list*, A_i^w , while the nodes in shouter range but not in whisperer range, called the *ring nodes* of i , are listed in the *ring-adjacency list*, A_i^r . These adjacency lists are the primary source of information as each node autonomously decides its transmission radius using local information in order to get close to an optimum ϕ value.

4 Self-Organization Algorithms

The scalability and flexibility of self-organization methods derives from three attributes:

1. *Decentralization of action*: Global organization emerges as a result of interaction between autonomous elements rather than by global prescription.
2. *Simplicity of decisions*: The decision space in which each element operates is very simple, and only has a few degrees of freedom.
3. *Locality of information*: The information used by each element in its decision-making is local (or limited in some other way).

The first two of these roughly determine what are termed *design degrees-of-freedom* (DDOF) [55]. Any successful self-organization algorithm must satisfy these conditions, and be able to generate near-optimally structured, heterogeneous configurations with high performance.

We present a series of self-organization algorithms based on a specific heuristic principle. While the algorithms are similar in terms of the decentralization of action and simplicity of decision criteria, each algorithm in the sequence takes increasingly more neighborhood information into account.

Each node in the system faces a binary choice of whether to be a whisperer or a shouter. The heuristic principle on which our algorithms are based is as follows: *If a node determines that it can reach all its ring-adjacent neighbors indirectly with connection radius ρ_w , it becomes a whisperer. Else, it must become a shouter.* When each node makes its choice using this criterion, the connectivity of the network is ensured, as many nodes as possible become whisperers, and shouters are deployed only when they are truly useful (i.e., would create connectivity beyond what is possible with whisperers alone). The nodes apply this criterion asynchronously and iteratively until the network is relaxed. The key difference among the algorithms is in the information used to determine whether the criterion is satisfied.

The heuristic described above tries to increase fitness in two ways: 1) It reduces the denominator of the fitness function (congestion) explicitly; and 2) It keeps the numerator of the fitness function (AISPL) relatively high by preventing disconnection. Though the first operation (reducing congestion) tends to *decrease* the AISPL and thus works against the second operation, this effect is largely swamped by the benefit of reduced congestion — especially in larger networks. More complex heuristics can mitigate this problem somewhat, but we use the current heuristic in the interest of simplicity.

The algorithms are as follows:

4.1 Basic Self-Organization (BSO) Algorithm

This is the basic algorithm with the least amount of information available to nodes. The process begins by each node transmitting a message giving its identifier and

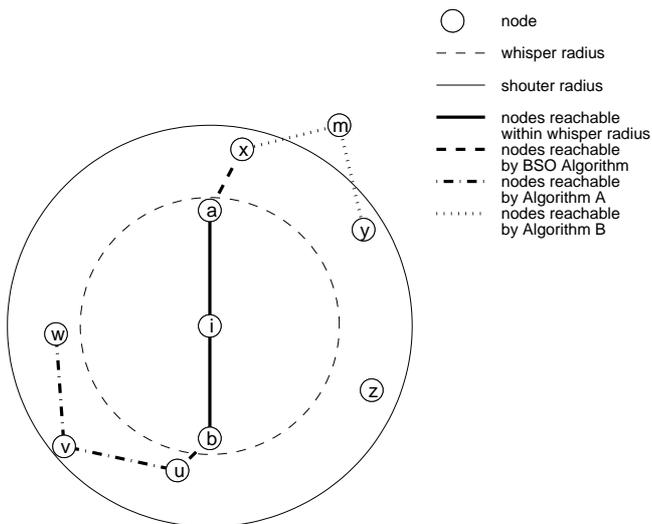


Fig. 1 A schematic representation of node i running the self organizing algorithm. Nodes a and b are whisper-adjacent to i , while u, v, w, x, y and z are ring-adjacent

its position (a so-called “hello” message) at shouter transmission power. As a result, all nodes acquire a list of nodes within shouter range of themselves, and their locations. From this, each node, i , can obtain its whisper-adjacency list, A_i^w , and its ring-adjacency list, A_i^r . Using these lists, each i determines if all nodes in A_i^r are within ρ_w of some node in A_i^w . If all nodes satisfy this criterion, $r_i = \rho_w$, else $r_i = \rho_s$.

This algorithm represents the most pessimistic and naive version of the heuristic principle stated above: Pessimistic because it assumes that all neighbors choose whisperer radii, and naive because it checks only for 1-hop links from nodes in A_i^w to those in A_i^r . Because of this, the algorithm makes maximally conservative choices, resulting in guaranteed connectivity but more shouters than needed to achieve it. The latter leads to high AISPL but also relatively high congestion.

4.2 Self-Organization Algorithm A

This algorithm relaxes the “naive” assumption in the BSO algorithm slightly. Each node obtains additional information on the current radius of the whisper-adjacent nodes. If a node, j , whisper-adjacent to i is a shouter, node i checks if the nodes in A_i^r are within ρ_s of j rather than within ρ_w . Node i also checks whether a ring-adjacent node not satisfying this condition is within ρ_w of a ring-adjacent node that does (see Figure 1). Node $r_i = \rho_s$ only if some node in A_i^r fails these tests. The additional cost of this step is minimal since the additional information used is the radius choices of i 's whisper-adjacent nodes. The result is that every node that was a

whisperer under BSO remains a whisperer and some that were shouters now become whisperers. Thus, Algorithm A tends to reduce congestion, but also reduces AISPL.

4.3 Self-Organization Algorithm B

As shown in Figure 1, it is possible that a node ring-adjacent to i is reachable, but requires a path through a node that is not a neighbor of i (about which i has no information in the previous algorithms). Algorithm B addresses this issue, and also relaxes the “pessimistic” assumption of the BSO algorithm. Here, each node, i , also obtains two additional items of information: 1) The current radius choices of all its ring-adjacent nodes; and 2) The whisper adjacency lists of all its ring-adjacent nodes. This information is obtained through additional communication following the initial “hello” message. Using the first item of information, i checks for connectivity from nodes in A_i^w using the *actual* current radii of neighbors rather than assuming whisperer radii for all ring-adjacent nodes. It also checks if each ring-adjacent node that is not reachable from A_i^w has an adjacent node that can be reached from a reachable ring-adjacent node. If so, the previously unreachable node is marked reachable because it has a path from A_i^w through a non-neighbor *link node* (see Figure 1). If any ring-adjacent nodes for i are still not reachable, i sets $r_i = \rho_s$, otherwise $r_i = \rho_w$. As with Algorithm A, this further reduces the total number of shouters, decreasing congestion further but also decreasing AISPL.

4.4 Self-Organization Algorithm C

The final algorithm we consider augments Algorithm B by using the *actual* radii of the link nodes described above. This requires that i have information about radii of nodes that are not within its ρ_s radius. This is still possible to obtain locally by having the ring-adjacent neighbors of i forward the radius information about their neighbors (which they have in Algorithm B) to i . Algorithm C has the effect of further reducing the number of shouters over Algorithm B.

Obviously, this progression of algorithms can be continued by having each node obtain information about a wider area around it, but that also diminishes scalability which is the main reason for using self-organization. Thus, we focus on simulations done with Algorithms A, B and C. As pointed out above, each successive algorithm reduces the number of shouters in the network, and an interesting issue is to see whether the gain due to reduced congestion is more than the loss due to reduced AISPL.

To address the main issue of the chapter (i.e., optimality), we also explicitly obtain optimized configurations for each layout using a genetic algorithm [56]. The algorithm is run repeatedly with high-mutation episodes, etc., to ensure reasonably that the configurations obtained are, indeed, very close to optimal in terms of fit-

ness. This is a crucial part of our study. In most cases previous proposals for self-organization of wireless networks [57, 58, 59, 60], there was no attempt to check how the results obtained compared with the best achievable. However, we are also constrained by the computational difficulty of obtaining optimal configurations, and this has been done only for relatively small networks (200 nodes). Given that the self-organization algorithms produced solutions close to optimal in this case, we expect that the same will hold true for larger systems. This is verified indirectly by comparing the fitness of randomly generated configurations with those obtained through self-organization.

Results from our simulations are presented and discussed in the next section.

5 Simulation, Results and Discussion

The simulations described below were done to address the following specific questions:

1. Are networks optimized through heuristic self-organization comparable in quality to those obtained by direct optimization?
2. Are the heterogeneous networks obtained through heuristic and direct optimization significantly different from the best radius-homogeneous and degree-homogeneous networks?
3. Are the optimized networks significantly different from statistically similar randomly generated networks?
4. How robust are the various optimized and homogeneous networks to node failure and targeted attack?

5.1 Simulations

Simulations were run for networks with 200 nodes and 1000 nodes. The whisperer radius was chosen as 0.8 times percolation radius ($\alpha = 0.8$) and shouter radius as 1.25 times percolation radius ($\beta = 1.25$). Percolation radius for 200 node network was set to 0.11, giving $\rho_w = 0.088$ and $\rho_s = 0.1375$. Percolation radius for 1000 node network was set at 0.0463, giving $\rho_w = 0.03704$ and $\rho_s = 0.05785$.

The genetic algorithm (GA) (see Appendix), which was run only for the 200 node case for computational reasons, used a population size of 20, and was run for 4000 generations. It used the fitness ϕ , which was also used to evaluate all other networks. Each solution in the GA population was encoded as a vector of length N (i.e., number of nodes), with the i th element indicating whether the node i was a whisperer or a shouter. New solutions were generated using two-point crossover of parents selected through an elitist method. Mutations were applied by switching whisperers to shouters and vice-versa with a probability of 0.05. To exclude the

possibility for premature convergence, the GA was run 3 times for each layout. Each run was “punctuated” by mutation shocks (high-mutation iterations) whenever the algorithm converged to a putative optimum, with at least 3 such punctuations in each run. Only if the algorithm converged repeatedly to very similar fitness values within and across runs was the best fitness achieved by the GA treated as optimal for a layout.

Both degree-homogeneous and radius-homogeneous networks were compared with optimized networks. For each layout, a range of degree and radius choices were evaluated for the DH and RH cases, respectively, and the best one was used for the comparison.

To compare optimized networks with randomly generated ones, it is necessary to choose the latter in a way that makes the comparison justified. We did this by scrambling the radius assignments for networks obtained through the GA for the 200-node case and through Algorithm C for the 1000-node case. Thus, the ratio of shouters and whisperers in all *scrambled networks* is identical to that in the best available configurations for those layouts. The only difference is in the assignment. The relative difference in quality between the two is, therefore, entirely the result of specific assignment rather than the generic attribute of radius distribution — reflecting the argument made by Doyle and others [16, 17] that optimal systems achieve their performance through *specific configuration*, and are atypical rather than generic within their class.

In addition to the scrambled networks generated as above, we also generated random networks with the fraction of shouters distributed around the value for the optimized cases rather than set equal to it. These networks are termed *random networks*.

5.2 Results and Discussion

5.2.1 Performance Comparison between Algorithms

Figure 2 shows the best fitness achieved by the GA and the three heuristic optimization algorithms (A, B and C) over 10 different 200-node layouts. Clearly, the GA outperforms the heuristic algorithms by a small amount in all cases, but the latter come close to achieving the optimal fitness. Also, there is a small but consistent improvement in performance from Algorithm A through Algorithm B to Algorithm C.

While it is important that the heuristic algorithms produce configurations with close to optimal fitness, the utility of these algorithms must ultimately be judged in comparison with simpler (and faster) non-optimizing algorithms. Figures 3 and 4 show results comparing Algorithm C — the best of the heuristic algorithms — with the two homogeneous configuration methods — RH and DH — and with scrambled networks to provide a baseline for performance. Clearly, Algorithm C outperforms all the non-optimizing configurations consistently for both 200-node

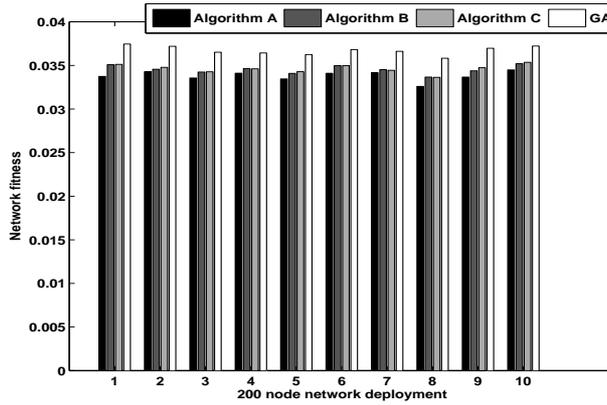


Fig. 2 Comparison of fitness for networks produced with Algorithms A,B and C and the GA for 10 different 200-node layouts. The graphs show that self-organized networks produce networks close to those obtained through the GA. Note that performance improves further as the self-organized algorithms use more information

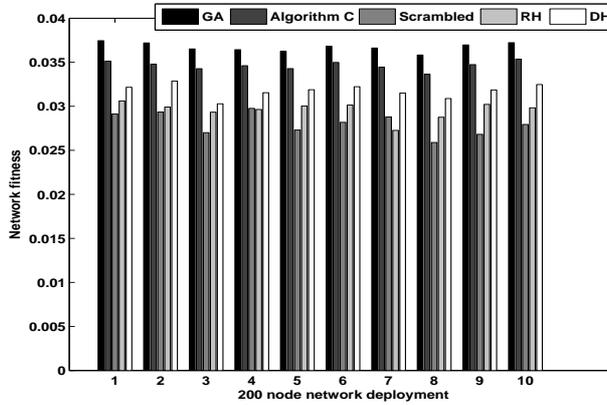


Fig. 3 Comparison of GA, Algorithm C, scrambled and homogeneous networks in the 200-node case, showing that self-organized networks are better than scrambled or homogeneous ones. The layouts are the same as in Figure 2

and 1000-node layouts, though it is interesting to note that degree-homogeneous (DH) configurations are also quite good, and much better than radius-homogeneous (RH) configurations which are not much better than the scrambled case. This is presumably because degree-homogeneity is a stronger guarantor of connectivity than radius-homogeneity.

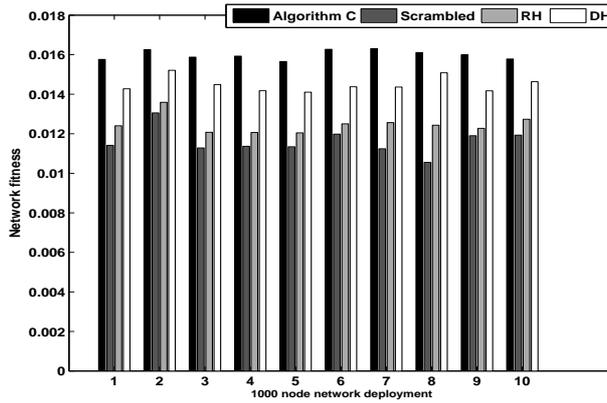


Fig. 4 Comparison of Algorithm C, scrambled and homogeneous networks in 1000-node case, showing the advantage of self-organized networks over scrambled or homogeneous ones. The x-axis indexes 10 independent random layouts of 1000 nodes each

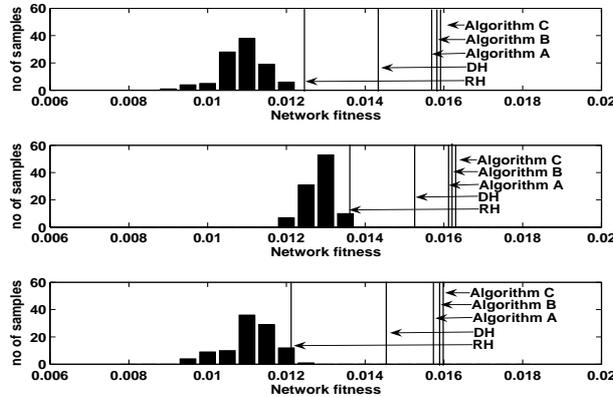


Fig. 5 Fitness distribution for 100 randomly scrambled networks for a particular layout, obtained from the best available configuration for the layout. The vertical lines indicate the best fitnesses obtained through various optimization procedures.

5.2.2 Comparison with Non-Optimized Networks

As discussed earlier, it has been argued that optimized configurations in application systems must be atypical, i.e., they must fall well outside the space of generic configurations. To test this for our optimized configurations, we randomly scrambled the radius assignments for the best optimized network in each case to generate a population of networks that had the same radius distribution as the optimal. Figure 5 shows the fitness distribution for this class of 1000-node networks as well as the fitnesses obtained for the DH and RH cases and the optimized cases. Clearly, the optimized algorithms all produce fitness well outside the generic range, indicating that the optimization has produced statistically significant benefits. Of the two homoge-

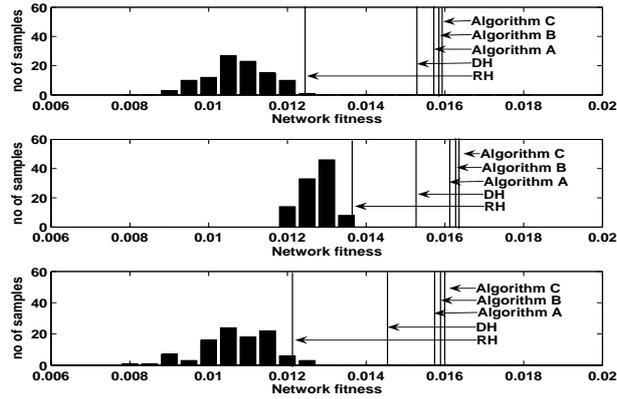


Fig. 6 Fitness distribution for 100 randomly generated networks for a particular layout, with shouter fractions distributed in a 30% window around that of the best available configuration for the layout. The vertical lines indicate the best fitnesses obtained through various optimization procedures.

neous cases, RH networks tend to fall in the upper tail of the scrambled network distribution, indicating that they are just very good random samples. However, the DH networks are well outside the generic fitness range, though still significantly worse than optimized networks.

Since the scrambled networks all have the same whisperer-shouter ratio, it is worth investigating whether varying this parameter may produce better configurations. To check this, we generated random networks for each layout with shouter-fractions ranging from 15% below that in the optimal case to 15% above it. The results for these random networks are shown in Figure 6, and are virtually identical to those for scrambled networks. The main conclusion from both these comparisons is that high-quality (near-optimal) networks cannot be produced by sampling and sifting randomly generated configurations — even those with carefully chosen shouter-fractions — and an explicit optimization process must be used. This emphasizes the necessity for efficient, scalable self-organizing optimization algorithms such as Algorithm C.

5.2.3 Robustness Evaluation

To consider the issue of robustness, each network configuration was subjected to systematic deterioration using random failure and targeted attack models. In the random failure case, randomly chosen nodes were removed from the network along with all connections to and from these nodes, while targeted attack involved removing nodes in decreasing order of out-degree. Figures 7 and 8 show the *percentage loss* of fitness as an increasing number of nodes is removed. Clearly, all configurations except those produced by Algorithm A have very similar response to random failure (Figure 7), but the responses are very different for targeted attack. As seen in

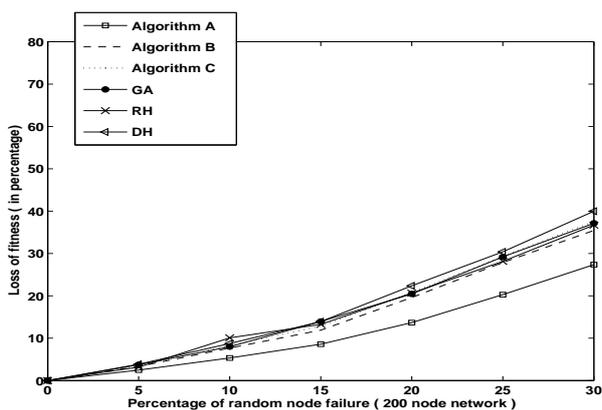


Fig. 7 Robustness of networks based on fitness loss with random node failure

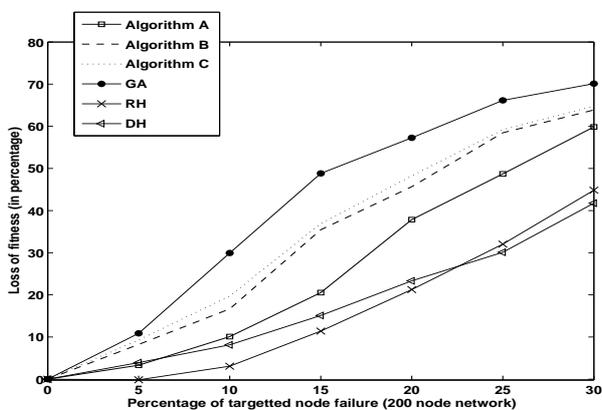


Fig. 8 Robustness of networks based on fitness loss with targeted node failure

Figure 8, the optimized configurations lose fitness much more rapidly in this case. Indeed, the robustness appears to depend inversely on the degree of optimization, with the GA-derived networks faring the worst followed by Algorithms C, B and A, with the two homogeneous cases showing the greatest robustness. This is seen very clearly in Figure 9, which plots the fitness of networks with 25% node loss against the fitness of the undamaged networks. Clearly, highly fit optimized networks show a strong divergence in their response to random failure and targeted attack while the non-customized homogeneous networks are almost identically affected in the two cases. Note that, by definition, the random and targeted failure cases are identical for DH networks.

Interestingly, Algorithm A configurations, which are the most robust in the random failure case, behave like non-optimized configurations for low levels of targeted attack but become more similar to optimized configurations for higher attack levels.

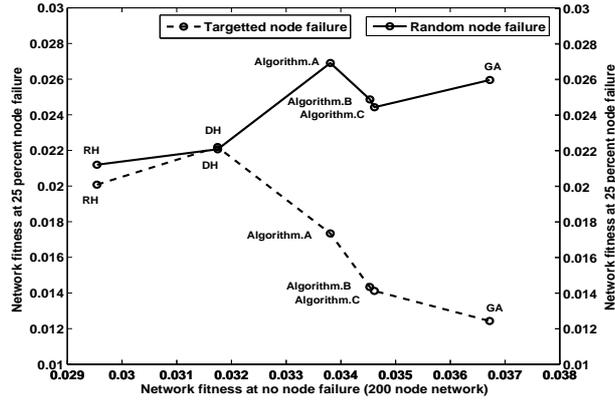


Fig. 9 Plot of fitness at 25% node failure against fitness in the no-failure case for different networks. The plots clearly show that the heterogeneous networks respond very differently to random and targeted failures while non-optimized homogeneous networks have virtually identical responses

Thus, overall, Algorithm A appears to be the best choice for balancing the benefits of optimization with robustness.

6 Conclusion

In this study, we have investigated heterogeneous geometric wireless networks with two types of nodes: Whisperers, projecting over a small radius, and shouters projecting over a larger radius. Several interesting conclusions about these heterogeneous networks have emerged from the study:

1. Given a node layout, very specific assignments of node type are needed to achieve optimal performance, which is significantly better than that of the best homogeneous networks.
2. The performance of optimized networks is not just a consequence of the fraction of hub nodes (shouters) in the population, but depends sensitively on the precise choice of these nodes. Thus, optimal configurations are *atypical* rather than generic in the space of configurations with the same radius distribution.
3. It is possible to find simple, efficient and scalable self-organization algorithms to produce near-optimal network configurations.
4. Highly optimized network configurations are as robust as non-optimized ones with respect to random node failure, but are much more susceptible to targeted attacks that preferentially remove nodes with the highest connectivity. All heterogeneous networks thus show a strong “robust-yet-fragile” effect [14, 15, 16, 17].
5. Of all the optimization procedures investigated, Algorithm A, which used the least amount of neighborhood information, provided the best balance between

performance and robustness. The results suggest that most of the benefits of heterogeneity can be obtained with minimal loss of robustness by using simple self-organization algorithms rather than more complex ones.

The study has also produced an interesting conclusion about homogeneous networks, showing that, with randomly deployed nodes, networks that seek homogeneous out-degree are much better than networks that simply use the same connection radius for all nodes — though both are worse than heterogeneous configurations. Interestingly, there is evidence that neuronal networks in the cerebral cortex configure themselves by trying to equalize the synaptic input for each neuron rather than equalizing the size of the axonal arbors (i.e., projection radius) [24].

Appendix

In order to assess the quality of the results obtained through self-organization, we generated optimal (or near-optimal) configurations for the 200-node layouts studied in the simulations. A genetic algorithm was used to produce these optimized configurations. The algorithm comprised the following steps.

Given a layout L with N nodes, set $generation = 1$:

1. Generate an initial population of n_{pop} configurations by assigning whisperer and shouter radii randomly to each node in n_{pop} copies of the layout. Thus, each configuration is represented by a binary string, $\zeta^k = \{\zeta_i^k\}$ of n_{pop} bits, where $\zeta_i^k = 1(0)$ means that node i in configuration k is a shouter(whisperer). These strings are the *chromosomes* representing the respective configurations.
2. Evaluate the fitness of each configuration in the current population using the fitness function defined above.
3. Select $0.8n_{pop}/2$ fittest configurations and $0.2n_{pop}/2$ randomly chosen configurations as *parents* for the next generation.
4. Randomly pair the parents into $n_{pop}/2$ couples, and generate two *offspring* from each couple using two-point crossover, i.e., both chromosomes in the couple are divided into three parts by breaking them at the same two randomly chosen points, and recombined into two new chromosomes of length N .
5. Select the $n_{pop}/4$ fittest configurations and $n_{pop}/4$ randomly chosen configurations as *survivors* from the current generation, i.e., not including the new offspring.
6. Remove the remaining members of the current generation and replace them with the offspring to obtain a new population of size n_{pop} .
7. In all but the two fittest survivors:
 - If mutation is to be normal, randomly switch 0.05% of the bits from 1 to 0 or vice versa.

- If the current generation is selected for a *mutation shock*, randomly switch 0.25% of the bits from 1 to 0 or vice versa. The mutation shock is intended to break the system out of a local optimum and prevent premature convergence.
8. If termination condition (maximum number of generations) is not met, increment *generation* by 1 and repeat from step 2.

References

1. D. Braha, A.A. Minai, and Y. Bar-Yam, Eds., *Complex Engineered Systems: Science Meets Technology*, Springer/NECSI, 2006.
2. A.-L. Barabási and R. Albert, “Emergence of scaling in random networks”, *Science*, vol. 286, pp. 509–511, 1999.
3. A.-L. Barabási, R. Albert, and H. Jeong, “Mean-field theory for scale-free random networks”, *Physica A*, vol. 272, pp. 173–187, 1999.
4. R. Albert and A.-L. Barabási, “Topology of evolving networks: Local events and universality”, *Physical Review Letters*, vol. 85, pp. 5234–5237, 2000.
5. R. Albert and A.L. Barabási, “Statistical mechanics of complex networks”, *Reviews of Modern Physics*, vol. 74, pp. 47–97, 2002.
6. D.J. Watts and S.H. Strogatz, “Collective dynamics of “small-world” networks”, *Nature*, vol. 393, pp. 440–442, 1998.
7. R. Albert, H. Jeong, and A.-L. Barabási, “Error and attack tolerance of complex networks”, *Nature*, vol. 406, pp. 378–382, 2000.
8. M.E.J. Newman, S.H. Strogatz, and D.J. Watts, “Random graphs with arbitrary degree distribution and their applications”, *Physical Review E*, vol. 64, pp. 026118, 2001.
9. B. Shargel, H. Sayama, I.J. Epstein, and Y. Bar-Yam, “Optimization of robustness and connectivity in complex networks”, *Physical Review Letters*, vol. 90, pp. 168701, 2003.
10. A.X.C.N. Valente, A. Sarkar, and H.A. Stone, “Two-peak and three-peak optimal complex networks”, *Physical Review Letters*, vol. 92, pp. 118702, 2004.
11. G. Paul, T. Tanizawa, S. Havlin, and H. E. Stanley, “Optimization of robustness of complex networks”, *European Physical Journal B*, vol. 38, pp. 187–191, 2004.
12. T. Tanizawa, G. Paul, R. Cohen, S. Havlin, and H. E. Stanley, “Optimization of network robustness to waves of targeted and random attacks”, *Physical Review E*, vol. 71, no. 4, pp. 047101, Apr. 2005.
13. A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish, “Improving network robustness by edge modification”, *Physica A*, vol. 357, pp. 593–612, 2005.
14. J. Doyle and J.M. Carlson, “Power laws, highly optimized tolerance, and generalized source coding”, *Physical Review Letters*, vol. 84, pp. 5656–5659, 2000.
15. J.M. Carlson and J. Doyle, “Highly optimized tolerance: Robustness and design in complex systems”, *Physical Review Letters*, vol. 84, pp. 2529–2532, 2000.
16. J.M. Carlson and J. Doyle, “Complexity and robustness”, *Proceedings of the National Academy of Sciences USA*, vol. 99 Suppl. 1, pp. 2539–2545, 2002.
17. L. Li, D. Alderson, R. Tanaka, J.C. Doyle, and W. Willinger, “Towards a theory of scale-free graphs: Definition, properties, and implications (extended version)”, Tech. Rep. CIT-CDS-04-006, Engineering and Applied Science, California Institute of Technology, 2005.
18. J.C. Doyle, D.L. Alderson, L. Li, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger, “The “robust yet fragile” nature of the internet”, *Proceedings of the National Academy of Sciences USA*, vol. 102, pp. 14497–14502, 2005.
19. E. Bonabeau, M. Dorigo, and G. Theraulaz, Eds., *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.

20. R. Nagpal, "Engineering amorphous systems, using global-to-local compilation", in *Complex Engineered Systems: Science Meets Technology*, D. Braha, A.A. Minai, and Y. Bar-Yam, Eds., pp. 291–306. Springer/NECSI, 2006.
21. A.A. Minai, D. Braha, and Y. Bar-Yam, "Complex engineered systems: A new paradigm", in *Complex Engineered Systems: Science Meets Technology*, D. Braha, A.A. Minai, and Y. Bar-Yam, Eds., pp. 1–21. Springer/NECSI, 2006.
22. R. Albert, H. Jeong, and A.-L. Barabási, "Diameter of the world wide web", *Nature*, vol. 401, pp. 130–131, 1999.
23. S.-H. Yook, H. Jeong, and A.-L. Barabási, "Modeling the internet's large-scale topology", *Proceedings of the National Academy of Sciences USA*, vol. 99, pp. 13382–13386, 2002.
24. A. Van Ooyen and J. Van Pelt, "Activity-dependent outgrowth of neurons and overshoot phenomena in developing neural networks", *Journal of Theoretical Biology*, vol. 167, pp. 27–43, 1994.
25. L. Kleinrock and J. Sylvester, "Optimum transmission radii for packet radio networks or why six is a magic number", in *NTC '78; National Telecommunications Conference, Birmingham, Ala., December 3-6, 1978, Conference Record. Volume 1. (A79-40501 17-32)* Piscataway, N.J., Institute of Electrical and Electronics Engineers, Inc., 1978, p. 4.3.1-4.3.5., 1978, pp. 431–435.
26. L. Takagi, H. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals", *IEEE Transactions on Communications* 32 (3) (1984) 246-257, vol. 32, pp. 246–257, 1984.
27. J. Ni and S.A.G. Chandler, "Connectivity properties of a random radio network", *IEE Proceedings - Communications*, vol. 141, pp. 289–296, 1994.
28. F. Xue and P.R. Kumar, "The number of neighbors needed for connectivity of wireless networks", *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.
29. O. Ferrari, G. Tonguz, "Minimum number of neighbors for fully connected uniform ad hoc wireless networks", in *Proceedings of IEEE International Conference on Communications*, June 2004, pp. 4331–4335.
30. Y.-C. Cheng and T.G. Robertazzi, "Critical connectivity phenomena in multihop radio models", *IEEE Transactions on Communications*, vol. 37, pp. 770–777, 1989.
31. P. Santi, "The critical transmitting range for connectivity in mobile ad hoc networks", *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 310–317, 2005.
32. D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, Taylor & Francis, London, UK, 1994.
33. C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network", in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002, pp. 80–91.
34. C. Bettstetter, "On the connectivity of wireless multihop networks with homogeneous and inhomogeneous range assignment", in *Proceedings of the IEEE Vehicular Technology Conference*, 2002.
35. E. Duarte-Melo and M. Liu, "Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks", in *Proceedings of IEEE GLOBECOM 2002*, November 2002.
36. R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment", in *Proceedings of IEEE INFOCOM 2000*, 2000, pp. 404–413.
37. S. Borbash and E. Jennings, "Distributed topology control algorithm for multihop wireless networks", in *Proceedings of the 2002 World Congress on Computational Intelligence*, 2002.
38. N. Li and J.C. Hou, "Localized topology control algorithms for heterogeneous wireless networks", *IEEE/ACM Transactions on Networking*, vol. 13, pp. 1313–1324, 2005.
39. G. Srivastava, P. Boustead, and J. Chicharo, "Connected fixed node degree based topologies in ad hoc networks", in *Proceedings of the 12th IEEE International Conference on Networks (ICON 2004)*, 2006, pp. 1330–1340.
40. P. Ranganathan, A. Ranganathan, A. Minai, and K. Berman, "A self-organizing heuristic for building optimal heterogeneous ad hoc sensor networks", in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC '06)*, 2006, pp. 774–779.

41. P. Ranganathan, A. Ranganathan, K. Berman, and A. Minai, "Discovering adaptive heuristics for ad-hoc sensor networks by mining evolved optimal configurations", in *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation (CEC'06)*, 2006, pp. 3064–3070.
42. A. Venuturumilli and A.A. Minai, "Obtaining robust wireless sensor networks through self-organization of heterogeneous connectivity", in *Proceedings of the 6th International Conference on Complex Systems*, 2006.
43. Y. Wang, X. Wang, D.P. Agrawal, and A.A. Minai, "Impact of heterogeneity on coverage and broadcast reachability in wireless sensor networks", in *Proceedings of the 15th International Conference on Computer Communications and Networks*, 2006, pp. 63–67.
44. D.J. Felleman and D.C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex", *Cerebral Cortex*, vol. 1, pp. 1–47, 1991.
45. O. Sporns, G. Tononi, and G.M. Edelman, "Theoretical neuroanatomy: Relating anatomical and functional connectivity in graphs and cortical connection matrices", *Cerebral Cortex*, vol. 10, pp. 127–141, 2000.
46. O. Sporns and G. Tononi, "Classes of network connectivity and dynamics", *Complexity*, vol. 7, pp. 28–38, 2002.
47. V. Raghunathan, C. Schurgers, S. Park, and M.B. Srivastava, "Energy-aware wireless microsensor networks", *IEEE Signal Processing Magazine*, pp. 40–50, 2002.
48. I.F. Akyildiz, W. Su, Y. Sankaasubramaniam, and E. Cayirci, "A survey on sensor networks", *IEEE Communications Magazine*, pp. 102–114, 2002.
49. W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", in *Proc. of the Hawaii International Conference on System Science, Maui*, 2000.
50. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: An energy-efficient coordination algorithm for topology maintenance in ad-hoc wireless networks", in *Mobile Computing and Networking*, 2001, pp. 85–96.
51. J.-H. Cui, R. Ammar, Z. Shi, Z. Zhou, S. Ibrahim, and H. Yan, "Enhancing underwater acoustic sensor networks using surface radios: issues, challenges and solutions", in *Sensor Networks*, G. Ferrari (ed.), 2009.
52. N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, and N.S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks", *IEEE Signal Processing Magazine*, vol. 22, pp. 54–69, 2005.
53. S. Gezici, Z. Tian, G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios", *IEEE Signal Processing Magazine*, vol. 22, pp. 70–84, 2005.
54. S. Kim, A.P. Brown, T. Pals, R.A. Iltis, and H. Lee, "Geolocation in ad hoc networks using DS-CDMA and generalized successive interference cancellation", *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 984–998, 2005.
55. D. Reynolds, J.M. Carlson, and J. Doyle, "Design degrees of freedom and mechanisms of complexity", *Physical Review E*, vol. 66, pp. 016108, 2005.
56. M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1998.
57. H. Zhang and A. Arora, "gs³: Scalable self-configuration and self-healing in wireless networks", *Computer Networks*, vol. 43, pp. 459–480, 2003.
58. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks", in *Proceedings of SenSys'03*, Los Angeles, CA, November 2003.
59. B. Krishnamachari, S. Wicker, R. Bejar, and C. Fernandez, "On the complexity of distributed self-configuration in wireless networks", *Telecommunication systems*, vol. 22, pp. 33–59, 2003.
60. H. Gupta, S.R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution", in *Proceedings of MobiHoc'03*, Annapolis, MD, June 2003.