Int. J. Sensor Networks, Vol. 1, Nos. 3/4, 2006 241

Self-organisation of sensor networks using genetic algorithms

Rahul Khanna*

Intel Corporation,
2111 NE 25th Avenue,
Hillsboro, OR 97124, USA
E-mail: rahul.khanna@intel.com
*Corresponding author

Huaping Liu

School of EECS, Oregon State University,
Corvallis, OR 97331, USA
E-mail: hliu@eecs.oregonstate.edu

Hsiao-Hwa Chen

Institute of Communications Engineering,
National Sun Yat-Sen University, Taiwan
E-mail: hshwchen@ieee.org

Abstract: In this paper we propose a reduced-complexity Genetic Algorithm (GA) for optimisation of multihop sensor networks. The goal of the system is to generate optimal number of sensor-clusters with Cluster-Heads (CHs). It results in minimisation of the power consumption of the sensor system while maximising the sensor objectives (coverage and exposure). The GA is used to adaptively create various components such as cluster-members, CHs and next-cluster. These components are then used to evaluate the average fitness of the system based on the sequence of communication links towards the sink. In addition, the mechanism supports dynamically changing coverage, task requirements, failures, incremental redeployment and reconfiguration.

Keywords: sensor networks; Genetic Algorithms (GA); network optimisation.

Reference to this paper should be made as follows: Khanna, R., Liu, H. and Chen, H-H. (2006) 'Self-organisation of sensor networks using genetic algorithms', *Int. J. Sensor Networks*, Vol. 1, Nos. 3/4, pp.241–252.

Biographical notes: Rahul Khanna is currently a Staff Architect at Intel Corporation, USA. He has authored technical papers in the areas of predictive modelling, motion estimation, wireless and security. He holds six patents and 12 pending.

Huaping Liu received his BS and MS degrees from Nanjing University of Posts and Telecommunications, Nanjing, PR China, in 1987 and 1990, respectively and his PhD from New Jersey Institute of Technology, Newark, NJ, in 1997, all in electrical engineering. From July 1997 to August 2001, he was with Lucent Technologies, New Jersey. He joined the School of Electrical Engineering and Computer Science at Oregon State University in September 2001 and is currently an Associate Professor. His research is in the area of wireless networks with a focus on CDMA, ultra-wideband and MIMO OFDM techniques.

Hsiao-Hwa Chen is currently a full-time Professor in National Sun Yat-Sen University, Taiwan. He has authored or co-authored over 180 technical papers in major international journals and conferences and five books in the areas of communications, including *Next Generation Wireless Systems and Networks* (498 pages) published by John Wiley and Sons in 2006.

1 Introduction

Advances in low-power digital integration and Micro-Electro-Mechanical Systems (MEMS) have paved the way for micro-sensors (Akkaya and Younis, 2005; Akyildiz et al., 2002; Katz et al., 1999; Min et al., 2001; Rabaey et al., 2000). These sensors are equipped with data processing capabilities along with sensory circuits. Sensor data are processed on these individual sensors and transmitted to the target (sink). Low-cost integration and small sizes of these sensors have generated special interest in the area of disposable-sensors. These are randomly deployed, infrastructure-less, data-centric sensors that cannot be charged or replaced. Queries to these sensors are addressed to nodes which have data satisfying the same condition. These disposable sensors find their uses in the areas of disaster recovery, target identification, reconnaissance, medical applications (Celler et al., 1994), defence applications (Marcy et al., 1999) and intrusion detection, etc. However, these sensors are constrained in energy, bandwidth, storage and processing capabilities. Large number of such sensors along with these constraints creates a sensor-management problem. At the network layer it amounts to setting up the energy-efficient route that transmits the non-redundant data from source to the sink in order to maximise the battery (and sensor's) life. This is done while adapting to changing connectivity due to failure of some nodes and new nodes powering up.

Clustering of a network to minimise the distance is an NP-hard problem (Fudenberg and Tirole, 1991; Pardalos and Wolkowicz, 1994). In this paper we develop an evolutionary algorithm (Goldber, 1989) that divides the randomly deployed sensors into an optimal number of independent clusters with Cluster-Head (CH) and optimal route. CH collects data from those sensors that belong to the cluster and sends them to the sink in a compressed manner via the most cost-effective route. It is assumed that while the sensors may be deployed in a non-hospitable environment, the sink is a stationary component that is located at a safe location.

Genetic Algorithm (GA) is a stochastic search technique that mimics the natural evolution proposed by Charles Darwin in 1858. GA has been successfully applied to a wide range of combination problems. They are particularly useful in applications involving design and optimisation, where there are a large number of variables and where procedural algorithms are either non-existent or extremely complicated.

In this paper, we undertake energy-efficient sensor-network design using GA approach. Deployment of this network can be done, for example, by dropping a large number of disposable Sensor Nodes (SNs) in a random fashion. The goal is to develop a long-lasting sensor network containing nodes with non-renewal and limited energy resource. To achieve this goal we discover clustered topology with optimal routes to the sink. These clusters have the ability to fuse the collected data at the CH, which are then routed to the sink using one or more hops. Therefore, GAs are designed with two objectives:

1. discover the optimal clusters with cluster members and CH and
2. discover low-cost path to the sink using one or more hops.

2 Proposed GA solution

The system consists of an initialisation module and an adaptation module. The initialisation module helps in coding of gene for each sensor. This gene contains the identification of each sensor and any other specific information. This information may be related to sensor objectivity, next-hop, cluster-domain, etc. The initialisation module also initiates temporary clusters of the sensors with a domain identification and CHs. The adaptation module is responsible for cluster adaptation and load adaptation. Cluster adaptation is responsible for creating accurate cluster boundaries due to addition, deletion or modified sensor objectives. Load adaptation is responsible for creating optimal routes from CHs to the sink. Adaptation modules are governed by a fitness function that is specific to the network objective in a load-balanced network. It prevents the flow of redundant information while maximising the network bandwidth usage and battery life.

It is interesting to note that two competing objectives are required to create an energy-efficient sensor network. While cluster membership will keep on changing because of dead or depleted nodes, routes to sink will keep on changing to avoid high-cost paths (like multiple clusters using the same CH to route the data to the sink). Therefore, we use Multiobjective GAs (MOGAs) (Horn et al., 1994). Simple GA converges to a single solution. In problems where there are several, often conflicting objectives, a MOGA is used which evolves a set of solutions (the population) towards the Pareto-optimal front where trade-off analysis can be performed to select a suitable solution.

2.1 Node selection chromosome representation

The chromosome of the GA contains all the building blocks to a solution of the problem at hand in a form that is suitable for the genetic operators and the fitness function. Each individual SN is represented by a 3-bit binary number called 'gene'. These three-bit genes which define the feature of the node are called 'allele' and are represented as follows:

000 – Node Inactive (powered off).

001 – Node chosen as CH.

010 – Node chosen as Inter-Cluster Router (ICR).

100 – Node chosen as Sensor (NS).

Each cluster is represented by a CH and cluster-members are represented by inactive/active sensors and ICRs. CH is responsible for data-fusion from various node-sensors and ICR is responsible for routing cluster data (from CH) to the sink.

For example, in a 25-node system, the number of bits required to represent the complete system would be $3 \times 25 = 75$. Therefore, the size of the string would be 60-bits. For the scenario shown in Figure 1, this string likes as follows:

100 000 001 100 001 100 100 100 001 100 100 001 100
100 100 000 001 100 100 001 010 010 010 010 010

Upon completion of the GA, a function is assigned to each node. Once the functions are assigned, each type of nodes then performs the following functions:

2.1.1 Inter-cluster routers

a. Each router starts listening to 'sink' or 'Lx router', where $x = 0, 1, 2, \ldots$, represents the number of hops between sink and itself.

b Each router finds out the next-hop energy requirements to the sink and/or Lx routers that it can listen to by exchanging data and bounds checking. This step also involves exchanging a conflict-free Proximity Unique ID (PUID) with other neighbouring routers (Section 3).

c Each router temporarily designates itself as Lx, where $x = 0, 1, 2, \ldots$, based on the next hop it sends the data to. $L(x)$ can send data to only an $L(x - 1)$ that is closer to the sink.

d Each router then sends the neighbouring routers (and/or sink) information (from step-b) to the sink using the temporary router chosen in step-c.

e Upon cost-analysis using a parallel GA, the sink will designate a primary and fail-over path to each router and send this information using the node it received that information from. This is a periodic process that repeats at a pre-defined interval.

f Lx routers will update its next-hop information by replacing the temporary next-hop with that provided by the sink. Lx routers will receive this information periodically from the sink.

g Lx routers will start advertising (router advertisement) its presence with the cost of using this path at regular intervals. This cost is evaluated using the following metrics:

  i average data flowing through this router (dynamic)

  ii energy requirements to reach next hop (static).

h Average cost of using the next-hop (static) Lx routers will trigger an attention message when the battery reaches an attention state (battery condition in quantized steps). This attention message is carried to the sink using the current path (updated in step-f). The sink will use this message as a trigger point for re-configuration and running a new instance of node-selection GA. In the new instance, the failing node is permanently marked ‘Powered Off (000b)’.

2.1.2 *Sensor nodes*

a Each SN starts listening to the available (CH advertisement).

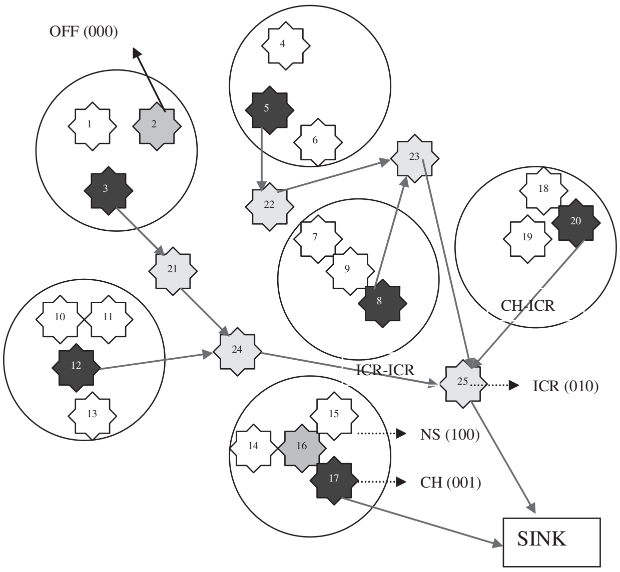
b Each SN will calculate the cost of communicating with the available CHs.

c Each sensor will attach to a CH based on the cost as calculated in step-b and become the part of that cluster. This step also involves receiving the unique PUID from the CH (Section 3).

d Each sensor will update the chosen CH with the sensor data. These data include the SN-CH cost of all CHs it evaluated in step-b.

e SN will trigger an attention message when the battery reaches an attention state (battery condition in quantised steps). This attention message is carried to the sink using the CH (step-b). Sink will use this message as a trigger point to reconfiguration and running a new instance of node-selection GA.

Figure 1 SN clustering. Each node is assigned function as a result of GA and the resulting chromosome structure. For the example below, chromosome structure is 100 000 001 100 001 100 100 100 001 100 100 001 100 100 100 000 001 100 100 001 010 010 010 010 010.



2.1.3 *Cluster head*

a Each CH starts CH advertisement to invite nodes (SN).

b Each CH sends the NS-ICR data received from the sensors to the sink.

c Each CH listens to the router advertisement and selects the ICR en-route to the sink. This step also involves receiving the unique PUID from the ICR (Section 3).

d These CHs can participate in data fusion. The resulting information is then communicated to the sink using the selected router. Sink returns back the Application Unique ID (AUID) (Section 3) to the CH during the set-up (or reinitialisation) phase.

2.1.4 *Sink*

Sink is an entity where all the event data collection and dissemination take place. This information is then processed for sensor related functions.

Sink also receives the statistical and status information from routers and CHs. This information is processed in the following manner:

a It collects the information regarding valid router-router (ICR-ICR) communication. This information includes energy requirements for communication and the corresponding Globally Unique Identification Number (GUID).

b It evaluates the average data that pass through each router by processing the data received by the sink (from SN).

c It evaluates the cost of NS-CH communication for all valid links. This information is passed by the CH during the setup-operation.

d It allocates an application specific AUID (Section 3) to the CH to uniquely identify the message trace.

e Listens to any alert message (battery conditions).

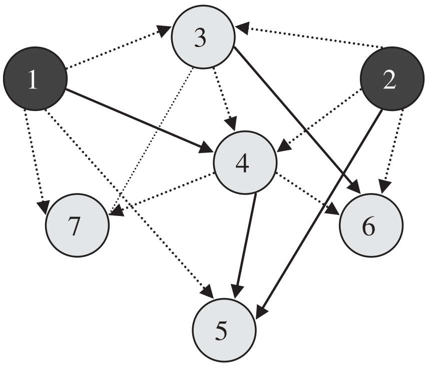
f Performs a GA to evaluate the optimal route using the fitness function based on parameters obtained in steps a and b. This is triggered based on periodicity or an alert event.

g Performs a GA to designate functional unit to each node using the fitness function based on the parameters obtained in step-c. This is triggered based on alert event.

2.2 *Route selection chromosome representation*

Route-selection GA uses a different chromosome structure than that used in node-selection GA. Characteristics of route-selection chromosomes are given as follows: Each node (CH and ICR) is represented by $\log_2(N)$ bits, where N is the maximum number of ICR nodes that can be reached by this node. Hence an individual in this case is represented by a string that consists of all such nodes with representation to the next ICR. For example, (0010) (0010) (001) (010) represents R12, R22, R31, R42 connections, where R_{xy} are the yth route of the xth node (Figure 2).

Figure 2 Chromosome structure of the route. For example, for nodes 1,2,3,4, chromosome string is represented by (0010) (0010) (001) (010); for node 1, route 2 is selected that connects to ICR 4; for node 2, route 2 is selected that connects to ICR 5; for node 3, route 1 is selected that connects to ICR 6



2.3 *Node selection fitness function*

The node selection fitness function is a weighted function that measures the quality or performance of a solution, in this case a specific sensor network design. This function is maximised by the GA system in the process of evolutionary optimisation. A fitness function must include and correctly represent all or at least the most important factors that affect the performance of the system. The major issue in developing a fitness function is the decision on which factors are the most important ones. We use the following measure.

2.3.1 *Coverage fitness*

A SN has an objective to maximise the blanket coverage where the objective is to maximise the total detection area. In many applications of sensor networks, the number of neighbouring nodes plays an important role. If the network is to be connected, the number of neighbours of each node needs to grow at $\Theta(\log n)$, where n is the number of nodes in the network (Xue and Kumar, 2004). Based on the density of deployment ρ, each node will have at least K neighbours and optimal isotropic communication range (R_c) with its neighbours (SNs and CHs) (Poduri and Sukhatme, 2004). Coverage Fitness (CF) depends on the percentage of nodes that have atleast K neighbours.

$$\mathrm{CF} = \sum_i \left(\min\left(1, \frac{N_i}{K}\right)\right) \tag{1}$$

where N_i is the number of fully connected SNs in cluster i.

2.3.2 *Cluster-head fitness*

SNs connected to each CH should be uniformly distributed. This prevents CH overloading. Cluster-Head-Fitness (CHF) defines the fitness based on the uniformity of the SNs and CHs:

$$\mathrm{CHF} = 1 - \min\left(1, \left(\sum_n \frac{|\rho_n - \rho|}{\rho}\right)/N\right) \tag{2}$$

where n is the CH number, N is the number of CHs in the system, ρ_n is the number of nodes attached to this CH and ρ is the average number of nodes per cluster in a system calculated as

$$\rho = \frac{\text{Total SNs}}{\text{Total CHs}} \tag{3}$$

Any cluster consisting of more than ρ SNs will be penalised.

2.3.3 *Node communication fitness*

A node needs power p to communicate with another node that is d distance away. The power required to communicate with the CH can be computed using the path loss expressed as (Li et al., 2001)

$$\mathrm{PL}(d) = \mathrm{PL}_0 + 10\mu \log_{10}\left(\frac{d}{d_0}\right) + S \tag{4}$$

where d is the distance between the sensors, d_0 is a reference distance typically chosen as $1m$ for sensor networks, PL_0 is the path loss at the reference distance d_0, μ is the path loss exponent, typical in the range of 2–4 and S is a zero-mean Gaussian random variable that gives the deviation in path loss from its average value.

For example, SNs calculates these values p by responding to CH advertisements that it can listen to during setup-operation. These values are then sent to the sink via a temporary low-cost path chosen by the SN during the set up phase. The Node Communication Fitness (NCF) function is obtained as

$$\mathrm{NCF} = 1 - \min\left(1, \sum_i \sum_j \left(\max\left(0, \frac{p_{ij} - p_t}{p_t}\right)\right) \Big/ N\right) \tag{5}$$

where p_{ij} represents inter-node communication energy relationship (as measured by individual SN), p_t represents energy threshold, and N is the number of SNs in the system.

2.3.4 *Battery status fitness*

Anytime SN communicates with the CH or CH communicates with ICR, there is a penalty paid in terms of battery usage. Battery is also consumed during the sensing operation or other related functions. Each node alerts the sink about its battery status (Q) when it crosses the quantised limit (or thresholds). These thresholds will be used to penalise the use of those nodes for operations that consume more battery power. Penalty for using the node with a low battery capacity depends upon the type of node and its usage. For example, a node with low battery capacity will have a greater penalty for ICRs than the CH. Similarly CH will have a greater penalty than the SN. Therefore penalty suffered by each node depends upon the battery status and the type of node assignment. The battery status fitness function is expressed as

$$\mathrm{BF} = 1 - F(Q, \text{Node Type}) \tag{6}$$

where $F(\cdot)$ is the penalty with $0 \leq F(Q, \text{Node Type}) \leq 1$.

2.3.5 *Router load fitness*

ICRs participate in routing the traffic originating from CHs or other ICR to the sink. Routers are penalised if they cater to more than the average number of CH and ICR. This avoids overloading routers. The Router Load Fitness (RLF) function is expressed as

$$\mathrm{RLF} = 1 - \sum_{n} \frac{\frac{|\varrho - \varrho_n|}{\varrho}}{N} \tag{7}$$

where n is the ICR index, N is the number of ICR in this system, and ϱ and ϱ_n are given as

$$\varrho = \frac{\text{Total (CH + ICR)}}{\text{Total ICR}} \tag{8a}$$

$$\varrho_n = \text{Connected CH} + \text{ICRCost}_n \tag{8b}$$

where ICRCost_n (nth ICR) is updated as a result of GA function (part of MOGA) that evaluates the *cost of using* a router using route selection fitness function (Section 2.4).

2.3.6 *Sensor effector fitness*

Sensor Effector Fitness (SEF) is the fitness measure that interprets the power consumed by the sensory action of clusters. The net effect of SEF is to rearrange the SNs so that the sensor data transmission is uniformly optimised by fusion, elimination or compression methods. Similar packets from multiple nodes can generate large amount of redundant data that can be aggregated to reduce transmissions. Aggregation is done by clustering or reclustering the nodes in order to perform data aggregation to save energy and is influenced by the following factors:

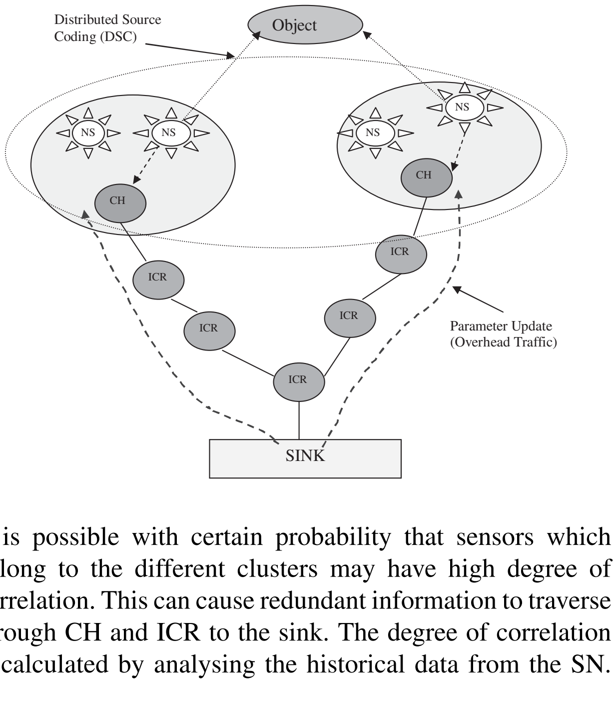
a Eliminating the duplicate data within the cluster (Heidemann et al., 2001). Furthermore, the sensor generating the duplicate data transitions to a low-duty-cycle state based on a leaky bucket hurestics.

b Compressing the data by fusing the correlated information within the cluster. This operation is performed by the CH that also updates the SN's spatial or temporal policy parameters.

c Compressing the data by correlating the information across the clusters. Wirelessly transmitting and receiving bits is the most energy consuming operation done by the nodes; therefore, by reducing the amount of bits that must be sent can significantly extend their lifetime (Chou et al., 2002). Mechanisms such as *Slepian-Wolf Distributed Source Coding (DSC)* (Cover and Thomas, 1991) can compress the content at the original sources in a distributed manner without explicit routing-based aggregation. This operation can only be performed using the sink mediation. Sink identifies the extent of correlation and updates the CH with new NS policy coding parameters. CH then applies these coding parameters to the respective NS in the associated cluster. This operation is performed using a predetermined update timer to avoid excessive transmissions from the sink while helping to adapt to the changing correlations. Higher inter-cluster correlations introduces extra communication overhead for mediated traffic (Figure 3).

d Reducing the data globally by eliminating the duplicate paths, turning off the sensors with high degree of redundancy or simply reclustering.

Figure 3 An example of DSC, where an object is monitored from two SNs encapsulated across different clusters. Coding parameters are exchanged with the help of the sink that also monitors the correlation between packets arriving from different clusters. This also introduces the transmission overhead on intermediate routers and CHs



It is possible with certain probability that sensors which belong to the different clusters may have high degree of correlation. This can cause redundant information to traverse through CH and ICR to the sink. The degree of correlation is calculated by analysing the historical data from the SN.

This analysis is done in spatial, temporal or spatio-temporal domain. Such analysis will rank the sensor based on the power usage for sensory functions. NS power usage depends on the following factors:

a Degree of correlation within local sensors.

b Degree of correlation between CHs carrying the fused data.

c Compression credit that reduces the redundancy in the information with some overhead. This overhead is calculated by measuring the extra information carried as side-bands or additional communication required to convey the compression parameters to the NS. Various compression schemes are addressed by Goel and Imielinski (2001) and Lindsey and Raghavendra (2002).

By exploiting the redundancy between various sensors in spatial, temporal or spatio-temporal dimensions, the average energy consumption per cluster can be optimised greatly. This energy has to be uniformly distributed among clusters. For example, a cluster allocated on the blind side of the sensory environment may not have much use as a SN. Sensors in that clusters may very well belong to a different clusters and be used as a CH or an ICR.

$$\mathrm{SEF} = \lambda_1 \left(1 - \left(\sum_j \min\left(1, \frac{|\sum_i (E_{ij}) - E_{\mathrm{avg}}|}{E_{\mathrm{avg}}}\right)\right)/N\right) + \lambda_2 \left(1 - \left(\sum_j \sum_i \left(I_{ij} H_{ij}/H_{\max}\right)\right)/M\right) \tag{9}$$

where E_{ij} is the average sensory power consumption of the ith SN in jth cluster, E_{avg} is the average sensory power consumed by all clusters, N and M are the total number of clusters and SNs, respectively, H_{ij} is the number of hops between jth cluster and the sink, $H_{\max}$ is the maximum number of hops possible in the SN, $I_{ij} = 1$ if SN i of cluster j is correlated with another SN in a different cluster and $I_{ij} = 0$ otherwise. λ_1 and λ_2, which satisfy $\lambda_1 + \lambda_2 = 0.5$, are the contributions to the SEF fitness due to sensory power consumption and overhead traffic, respectively.

2.3.7 *Total node fitness*

Total Node Fitness (TNF) is the final fitness that is evaluated in the GA for the appropriate node assignment. It is described by

$$\mathrm{TNF} = \alpha_1 \mathrm{CHF} + \alpha_2 \mathrm{NCF} + \alpha_3 \mathrm{BF} + \alpha_4 \mathrm{RFL} + \alpha_5 \mathrm{SEF} + \alpha_6 \mathrm{CF} \tag{10}$$

where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = 1$ and α_i depends upon the relative significance of the component. These values can be made adaptive using an external heuristics.

2.4 *Route selection fitness function*

The second objective of the MOGA is to generate balanced routes based on node allocation using GA based on node fitness function. During setup-operation, both CHs and ICRs start sending the data on the most cost effective ICR. It is not guaranteed that the set-up connection will remain cost-effective over a period of time. GA predicts the optimal route topology based on the cost of using an ICR for the next sampling period. CH and ICR are updated with this information in each sampling period. Route fitness function takes into account the traffic patterns, battery capacity and transmission energy. This is accomplished because of the following properties of the sink:

a It is aware of the static routes that are either formed during the setup operations or updated during GA operations during a sampling period. This will help GA evaluate average load on each router (since destination of all communication is the sink).

b It is aware of the amount of data (bits) received from each cluster which then traverse through a static routes as in (a).

c Each ICR updates the sink of its battery capacity as soon as it crosses a threshold value.

d It is aware of the energy-cost of transmission to its nearest neighbours. This is proportional to the distance between ICH and its next hop (or sink). This information is sent by the router during the set-up phase.

Based on the predicted optimal route fitness, the sink will update the cost of using this ICR for the next sampling period. The Total Route Fitness (TRF) is given by:

$$\mathrm{TRF} = \mathrm{BF} + \mathrm{NCF} + \left(\sum_k ((\max(\mathrm{ICR}(j,k)) - \mathrm{Curr}(\mathrm{ICR}(j,k)))/\max(\mathrm{ICR}(j,k)))\right)/N \tag{11}$$

where

$\mathrm{ICR}(j,k)$ = Average bit-rate handled by ICR j that can communicate with node k (CH or ICR)

$\max(\mathrm{ICR}(j,k))$ = ICR with the highest bit-rate that can be communicated by node k

$\mathrm{Curr}(\mathrm{ICR}(j,k))$ = Current ICR that has been designated to communicate with node k

N = Total number of nodes (CH and ICR)

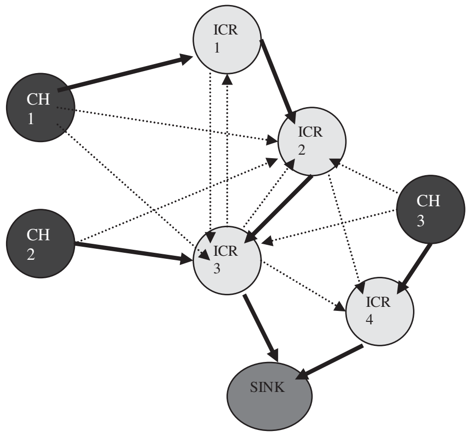
BF = Battery Fitness (BF) of the router in question

NCF = Node (CH-ICR or ICR-ICR) communication fitness

$$\mathrm{ICRCost}_n = 1 - \left(\sum_k ((\max(\mathrm{ICR}(j,k)) - \mathrm{ICR}(j=n,k))/\max(\mathrm{ICR}(j,k)))\right)/M$$

where k = all the nodes that can communicate with ICR n. For example in Figure 4, for ICR-2, k = {CH1, CH2, CH3, ICR1, ICR3}, M equals the total number of k nodes that can communicate with ICR n; for ICR-2, $M = 5$.

Figure 4 Sample output of the route-selection GA. Thick lines represent the low-cost selected route. Dotted lines represent other possible routes, but with higher relative cost (lower fitness). As the system conditions change, low-cost routes can become high-cost routes and vice versa. There is a high likelihood that ICR 3 may turn out to be a high-cost path if CH 2 becomes highly active



2.5 Node selection GA

Now that we have defined a node selection fitness, we can design the GA for node-selection that can be represented with the following steps using the GA operators. The process of GA takes place in the sink (or a similar centralized identity). This algorithm repeats itself upon multiple triggers. These triggers are related to battery alert, deteriorating route fitness alert, periodic action. Once the optimal fitness is achieved, the topology corresponding to that fitness is committed and the sensors are instructed to assume the new functions by relinquishing the old functions.

a *Initial population*: initial chromosomes strings are seeded partially randomly using a Random Number Generator (RNG) and partially using population of previous samples. Population uses the gene structure as defined in Section 2.1. This population is coded with gene structure as defined in Section 2.1.

b *Evaluation*: each chromosome string is evaluated for the fitness using the TNF function (for node assignment) as defined in Section 2.2.

c *Reproduction*: reproduction is a process in which individual strings are copied according to there fitness function values, which also means that individuals with larger fitness value will have a higher probability of contributing an offspring in the next generation. The algorithm uses the standard weighted roulette wheel method to select n individuals for reproduction to the mating pool. Since the TNF defines the fitness value, the chromosome with the highest fitness value means represents a better chromosome to take part in reproduction. N chromosomes will again be reproduced from the n chromosomes selected for reproduction using a crossover probability. During reproduction, we choose multiple cross-over points. Cross-over points and the locations are calculated using an RNG. As in this example, two chromosome strings having three random cross-over points will create a resultant chromosome after cross-over as below:

Parents:

100 000 001 100 001 100 100 100 001 100 **100 001 100 100 100 000 001** 100 100 001 010 010 010 010 010 **100 010 010 100 100** 010 100 010 001 001 **001 001 010 001 001 000 100** 010 001 100 010 000 001 100 010

Children:

100 000 001 100 001 **010 100 010 001 001** 100 001 100 100 100 000 001 **010 001 100 010 000 001 100 010** 100 010 010 100 100 **100 100 100 001 100** 001 001 010 001 001 000 100 **100 100 001 010 010 010 010 010**

d *Mutation*: newly reproduced N chromosomes are transferred to the mutation pool. The mutation operator mutates chromosome in the mutation pool according to mutation probability which will make it adaptive. We will choose a maximum mutation probability p_m. In any generation, mutation probability will be inversely proportional to the average fitness of the standard number of population in any generation. Therefore

$$p_g = \frac{\mathrm{p_m}1 - (N \times \mathrm{TNF_{avg}})}{\mathrm{TNF_{total}}}. \tag{13}$$

Mutation function uses function flip (toss of a coin) to decide whether to invert the bit or not.

e *Selection*: finally N chromosomes are chosen out of $2N$ chromosomes according to their fitness values. These chromosomes are carried over to the next generation. $2N$ chromosomes consist of N parent chromosomes and N children.

2.6 Route selection GA

Route selection GA is similar to the node selection GA with the following exceptions and an extra trigger point. This algorithm repeats itself at a regular interval to ascertain the acceptable thresholds of route-loads during the constant usage of the sensor-system.

a *Initial population*: initial population is chosen partially randomly using an RNG and partially using population of previous samples. Population uses the gene structure as defined above.

b *Evaluation*: each chromosome is evaluated for the fitness using the TRF function (for route selection) as defined in Section 2.4.

c *Node selection trigger point*: route-selection GA keeps on making attempts to achieve the most cost-effective path for a given topology (as selected in node-selection) in which there can be multiple paths. At certain point certain fitness threshold is reached beyond which further conversion to higher fitness may not be possible. This condition can happen due to battery condition and bad node assignments. This will cause a node-reassignment alert, which in turn will cause the node-selection GA to run again with changed conditions.

As seen above, the route selection can sometimes act as a resisting factor for the node-selection. While nodes may have been assigned the functions based on a high fitness factor, it may not be suitable for routing the packets in a multihop system. This will cause a reconfiguration (running route selection GA) again until both objectives reach an acceptable convergence point. This is a dynamic process and keeps on repeating over the life-time of the system.

3 Naming convention

Naming convention is an important ingredient of the sensor network system architecture. An optimal naming convention will reduce the messaging overhead as well as facilitate collaborative signal processing. Each sensor supports three types of naming conventions:

a *Globally Unique Sensor ID* is the uniquely identifiable node identification number that is hard-coded in the sensor hardware. This ID is used during the set-up phase (or reinitialisation phase).

b PUID is the unique identification of the nodes contained within the neighbourhood of a cluster (NS ID , CH ID, and ICR ID). This ID is used among the neighbouring nodes for link-layer data exchange and collaborative signal processing. These IDs are allocated dynamically in the following manner:

  i ICR(s) self-allocate its own PUID during the set-up (or reinitialisation) phase that involves negotiation between the neighbouring ICR(s). This ID is used for ICR-ICR link-layer forwarding and not contained in the message header.

  ii CH(s) receives its PUID from ICR(s) during the CH-ICR binding operation (Section 2.1.3). This ID is used for CH-ICR link-layer forwarding and not contained in the message header.

  iii NS(s) receives its PUID from CH(s) during the NS-CH binding operation (Section 2.1.2). This ID is used for NS-CH link-layer forwarding. This ID can optionally be carried in the message header when the message flows through the ICR.

c AUID is allocated by the sink to the CH for message identification. This ID enables the sink to uniquely identify the origin of the message up to the node level. This is a data-layer ID that is carried in the message header and filled in by the CH. This ID is for application use only and system topology has no visibility into its construction or usage. AUID can be attribute-named data that can enable in-network processing with filters, supporting data aggregation, nested queries and similar techniques that are critical to reduce network traffic and conserve energy (Heidemann et al., 2001).

The addressing in this manner reduces the overhead due to addressing bits during the message transmission. This method is scalable to the size of the network. Smaller networks with fewer nodes will use less addressing bits than the larger networks.

4 Overhead traffic

The GA used to perform *Node Selection* and *Route Selection* is targeted with two competitive objectives running at the sink. The introduction of this layer as a separate protocol aids in using the snoop data for predicting the fitness data. It is important to note that the sink builds up the *Node Database* tree by snooping the routed data and the setup-data (initialisation step) consisting of node ID, transmission distance (to ICR and other neighbouring nodes), associated CH and routes to the sink (primary and fail-over routes). This data is further evaluated at the sink for the creation of additional data-points consisting of various fitness categories defined in Sections 2.3 and 2.4. While most of the fitness data can be indirectly inferred from the regular data (normal data and setup data), battery loss due to coverage cannot be measured using this method. Moreover, changes in the associations and designations require extra messaging between sink and nodes. Hence various sources of the overhead traffic are:

i *Set-up messaging*: this is mandatory messaging [2.1] required for initial set-up of the network. These messages are exchanged once during initialisation-step and in extreme cases can also be triggered by sink.

ii *Sink alert messaging*: this message is performed when sink determines the need for new allocations for nodes and routes as a result of performing GA.

iii *Node alert messaging*: this messaging is initiated by the nodes towards sink, in order to identify critical information. Currently, this messaging is used for alerting when:

  a Battery levels falls below the thresholds.

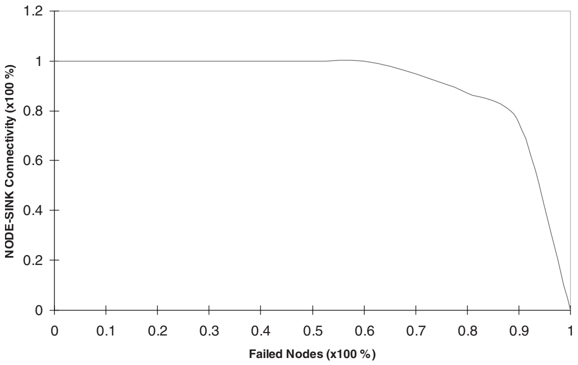
  b A fail-over route is chosen for future routings. Each ICR identifies itself (starting with the failed over ICR).

iv *Node battery status*: the node battery status data flow through the preestablished path using the extra bits in the SN messaging data. Three extra bits in the header gives eight quantisation levels.

v *Correlation data timer*: this timer updates the spatial or temporal correlation parameters of the SNs that compresses the overall data in a distributed coding (Cover and Thomas, 1991). An update has to traverse through all the ICR(s) en-route to the cooperating nodes residing in different clusters. This is a low duty-cycle operation that can adapt to the changing environment thus limiting the expenditure involved in update overhead.

As described, most of the overhead traffic is caused during the setup-operation. A small percentage of messages also flow as Alerts between source and sink. Most of the cost of indirect inference and GA execution is pushed to the sink. As a result, cost of overhead-traffic on the nodes is greatly reduced. Increase in the number of nodes (large networks) shows a longer delay in *Initialisation Setup* and *Message Propagation*. Longer delays are mitigated by subdividing a large network into smaller domains with identifiable boundaries (Figure 5).

Figure 5 Percentage of nodes connected to the sink in the event of node failure



5 Sensor construction

A self-organisation mechanism using a contention-free TDMA medium access protocol can be used for sensor networks (Sohrabi and Pottie, 1999). During the set-up phase, nodes interact in a random access manner. As the clusters are formed during the set-up phase or reclustering event (upon GA trigger), incremental TDMA schedules are formed for SNs, CHs, and ICRs. Reclustering as a result of GA evaluation (and the corresponding alert) will modify the existing TDMA schedules to accommodate the newly formed clusters. Incremental determination of the TDMA schedules is based on the known range limitations on the radio where certain nodes are expected to be outside the region of radio interference with the current node. At the same time, as a result of distributed scheduling, some nodes with similar schedules may interfere with each other. This can be remedied by using multiple channels or spreading codes, that will reduce the overhead due to transmission management. Typical sensors used in this scenario have the following characteristics (Figure 6):

a *Transmitter*: transmits the data at various power levels. It should have the ability to quickly enter and exit from one power-state to another.

b *Receiver*: receives the data targeted toward it. It additionally comprises of a low-power listening engine. Sensor radio consumes almost the same energy as when transmitting, searching for the next packet. To reduce energy consumption during the non-transmitting periods, energy-aware protocols are used. These protocols involve extra transmission overhead in terms of extra attention bits and reduced sampling of the radio channel.

c *Power control*: controls the power to be transmitted according to the function assigned.

d *Function control*: performs the protocol actions related to the function. These actions are related to identification, advertising, power-control, performing bindings and associations with other nodes and sensing, etc.

e *Test control*: performs the test functions. Test control is transparent to the function control and does not interfere with its working. This control is required to simulate a future topology while not interfering with the current one. GA will make use of this function to evaluate the fitness before committing this topology to all nodes.

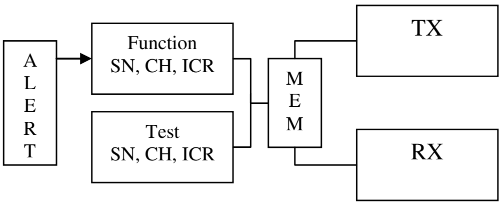
f *Alert generation*: generates an alert action to the sink upon any critical/warning or quantised event (like battery depletion). Alert data are used by the sink in the evaluation of the fitness parameters.

g *Memory*: limited memory is required to collect the data payload related to sensing, test-data or route-queues. A buffer overflow can cause the packets to be dropped. A temporary overflow memory allocation can be received from the neighbouring node that belongs to the same cluster. Such allocation can be expensive as it comes at the cost of transmission overhead.

h *Adaptive duty cycling*: all ICR nodes put themselves in a low-duty-cycle state when they are not required to transmit any data through them. A Wake-on-Wireless (WoW) signal (Shih et al., 2002) can wake a sensor from the low-power state to the high-power state.

These characteristics are required for proper functioning of the self-organising sensor network using GAs. Most of these characteristics are related to the functional adaptation of the sensor based on function allocation by the sink without interrupting the sensing operation.

Figure 6 Illustration of sensor construction



6 Numerical results

Experimental set-up consists of 100, 225, 400, 625 nodes placed at random positions in a 30 × 30 space. Each of the nodes picks up a random coordinate between (0, 0) and (30, 30) and assigns itself an UUID and a random battery capacity between 0 and 15. Once all the nodes have placed themselves in the listen mode, GA is run with the following parameters:

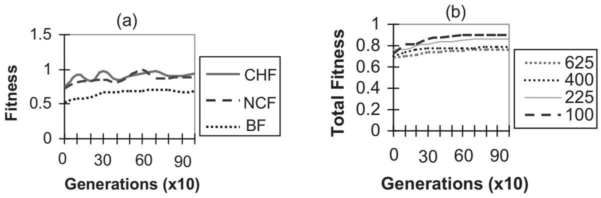
- Population size = 0.75 (number of nodes)
- Crossover rate = 0.8 (n − point cross-over)

- Mutation rate = 0.004
- Number of generations = 1000.

The experiment is simulated in an environment where each node acts as a Linux thread. Once GA run has completed, it assigns a function to each of these threads. These threads then start acting as independent nodes and initiate the node specific protocol. Each of these independent threads is capable of simulating battery depletion and transmission energy. In the experiment, it is assumed that there are no obstructions in the sensor transmit/receive path.

As seen in Figure 7(b), convergence points are dependent upon the number of nodes being optimised. In all four cases convergence is reached within the first 500 generations. After that point improvement in the fitness is minimal. We can call this an 80% fitness point. After this point we may use a deterministic approach to achieve further fitness. As seen in Figure 7(a), CH fitness, NCF and battery fitness increase monotonically with the number of generations. The same is true for the total fitness (Figure 7(b)), which is a function of all the individual fitness. Also, as seen in Figure 5, complete connectivity between non-extinct nodes and the sink can be maintained until 65% of the nodes die. While the death of sensors will reduce the coverage, the presence of efficient routing will reduce the number of orphan nodes.

Figure 7 (a) Fitness chart for CHF/NCF/BF for 100 nodes; (b) Total fitness chart for 100, 225, 400 and 625 nodes set-up (right)



Another important data point is the effect of isotropic communication range R_c of SNs on the average power consumption as a result of cluster set-up. Since the clustering decision is based on the density of the sensor deployment and the non-overlapping communication distance, a non-optimal distance can be expensive. This is due to the fact that the non-optimal distance will cause multiple collision, extra address bits (PUID) for unique identification in a denser environment and increased messaging. As seen in Figure 8, the optimal communication distance is reached at the minimum point in the valley. If the distance is less than the optimal, it requires more sensors to fulfill the coverage requirements. At the same time, we also observe an increased ICR messaging due to increase in the number of hops (ICR). If the distance is more than the optimal, then it reduces the coverage by turning off the neighbouring sensors. This in turn increases the communication traffic due to decrease in data aggregation and fusion.

Longterm network sustainability depends upon the function distribution based on the total residual energy of the nodes. A bad allocation create pockets of no connectivity or connectivity with limited coverage. A suboptimal function allocation can also cause frequent reclustering and a practically unstable system. This can also result in non-useful energy expenditure. Average residual energy of the CH is good measure of the effectiveness of the functional allocation of the nodes that tries to optimise the available power in the system. Reclustering process ensures that a node functioning as a CH is reprovisioned with a different function with less energy requirements. This mechanism prevents CH from depleting its energy levels quickly by exchanging the roles with its neighbouring nodes. In Figure 9, node population is increased from 100 nodes to 625 nodes for the same area. Average residual energy of the CH varies between 61% to 79% of its total energy. For each population, measurements are made at regular intervals till the system loses its connectivity (Figure 5). In all 18% change in residual energy is recorded for 600% change in the node density.

Figure 8 Average power consumption of the network with respect to variation in the communication distance R_c of the SNs.

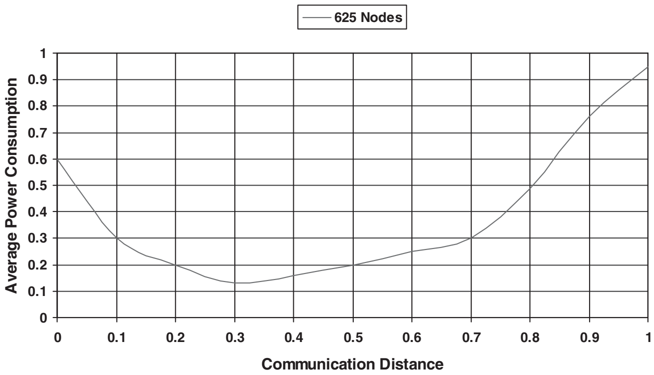
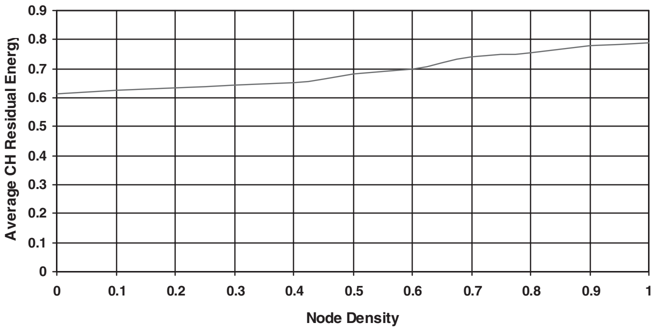


Figure 9 Normalised plot of node density (100–625 nodes) and the average residual energy of the CH over the period of full connectivity

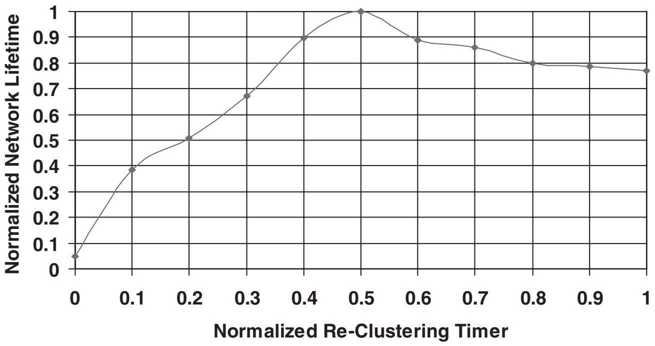


GA tries to balance the routes by optimally generating new routes based on the overall fitness (Section 2.4). This increases the overall lifetime of the system. ICR, CH and NS roles are exchanged by performing GA re-evaluation using a static timer or NODE alerts. This re-evaluation tries to rebalance the energy allocations based on the functional requirements as well as the historical traffic patterns, although with a set-up transmission penalty. This penalty decreases with the overall system usage, because SINK is able to calculate the NODE parameters based on the side-band information (Section 4). Figure 10 shows the effect of reclustering timer (re-evaluate GA) on the overall life expectancy of the network. Once the optimal set-up is reached, we do not observe any significant improvement in the life expectancy. At the same time we see a slow degradation in the life expectancy because of the increased transmission overhead. Therefore GA re-evaluation can be optimised by adapting the timer period based on the fitness matrix. We can evaluate the periodicity based on the following equation.

$$\text{Period} = K\left(1 - \left(\alpha_1 \sum_{n=t-x}^{n=t} |\text{TNF}_n - TNF_{n-1}| + \alpha_2 \sum_{n=t-x}^{n=t} |TRF_n - TRF_{n-1}|\right)/x\right) \quad (14)$$

where, K is the Maximum Timer period, t is the Current Instance, x is the number of past instances, TNF_n is the Total Node Fitness at the instance n of the GA re-evaluation (Equation 10), TRF_n is the Total Route Fitness at instance n of the GA re-evaluation (Equation 11) and α_1 and α_2 are the relative contributions and $\alpha_1 + \alpha_2 = 1$.

Figure 10 Normalised plot of Re-Clustering timer and the effect on the lifetime of the sensor network. Lifetime is measured from the start to the point where first sensor becomes unusable



7 Conclusion

In this paper we have presented a novel approach to design a self-organising network based on GAs. Sensors that are placed at random are assigned functions (sensing node, CH, router, or inactive-node) based upon the results of GA. The GA approach optimises the network to maximise energy usage along with battery conservation with route optimisation. It can be shown that the periodic run of a GA will help conserve the overall energy of the system with maximum operability. As it can be seen from Figure 7(b), individual components tends toward maximising their fitness with the passing generations in a uniform manner. That shows that the goal of maximising the system fitness along with individual component fitness can be achieved with a considerably reduced complexity. The algorithm also prevents the over-optimisation of an individual fitness component at the cost of other components. One of the challenges in GA is to be able to converge in the shortest time possible. As an extension of this paper, we will show the applicability of demand-based, mixed model where we run GA until convergence and then run traditional algorithms (e.g. TABU, directed diffusion, etc.) to achieve the target fitness. We will also research the prediction of system usage and the resulting topologies based on historical trends. The derivatives of these trends can then be used to define an individual fitness along with the current fitness parameters which will improve upon the uniform SN usage assumption. As a part of future research, we will continue to work on improvements related to the security and the corresponding overhead. We also plan to address the challenges involved in the identification of domain boundaries in large networks which can be partitioned into multiple small network domains capable of performing GA. Another aspect that needs research is the ability to reduce the ICR traffic on cross-cluster aggregation (or fusion). While aggregation parameters are conveyed to the CHs using sink mediation, this is less effective in a fast-changing environment because of overhead traffic originating from the sink.

References

Akkaya, K. and Younis, M. (2005) 'A survey on routing protocols for wirelsss sensor networks', *Ad Hoc Networks*, Vol. 3, pp.325–349.

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) 'Wireless sensor networks: a survey', *Computer Networks*, Vol. 38, pp.393–422.

Celler, B.G., Hesketh, T., Earnshaw, W. and Ilsar, E.D. (1994) 'An instrumentation system for the remote monitoring of changes in functional health status of the elderly at home', *Proceedings of 16th IEEE Annual International Conference Engineering Medicine and Biology Society*, Baltimore, MD, USA, pp.908–909.

Chou, J., Petrovic, D. and Ramchandran, K. (2002) 'Tracking and exploiting correlations in dense sensor networks', *Proceedings 36th Asilomar Conference Signals, Systems and Computers*.

Cover, T.M. and Thomas, J.A. (1991) *Elements of Information Theory*, John Wiley.

Fudenberg, D. and Tirole, J. (1991) *Game Theory*, MIT Press.

Goldberg, D. (1989) *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Inc.

Goel, S. and Imielinski, T. (2001) 'Prediction-based monitoring in sensor networks: taking lessons from MPEG', *ACM Computer Communication Review*, Vol. 31, pp.82–98.

Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D. and Ganesan, D. (2001) 'Building efficient wireless sensor networks with low-level naming', *Proceedings ACM Symposium Operating Systems Principles*, Chateau Lake Louise, Banff, Alberta, Canada, pp.146–159.

Horn, J., Nafpliotis, J.N. and Goldberg, D. (1994) 'A niched Pareto genetic algorithm for multiobjective optimization: evolutionary computation', *Proceedings First IEEE Conference Computational Intelligence*, pp.82–87.

Katz, R.H., Kahn, J.M. and Pister, K.S.J. (1999) 'Mobile networking for smart dust', *Proceedings Fifth Annual ACM/IEEE International Conference Mobile Computing and Networking (MobiCom'99)*, Seattle, WA.

Li, Q., Aslam, J. and Rus, D. (2001) 'Hierarchical power aware routing in sensor networks', *Proceedings DIMACS Workshop on Pervasive Networking*.

Lindsey, S. and Raghavendra, C.S. (2002) 'PEGASIS: power-efficient gathering in sensor information systems', *Proceedings Aerospace Conference*.

Marcy, H.O., Agre, J.R., Chien, C., Clare, L.P., Romanov, N. and Twarowski, A. (1999) 'Wireless sensor networks for area monitoring and integrated vehicle health management applications', *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Portland, OR, USA.

Min, R., Bhardwaj, M., Cho, S-H., Shih, E., Sinha, A., Wang, A. and Chandrakasan, A. (2001) 'Low power wireless sensor networks', *Proceedings of the International Conference VLSI Design*, Bangalore, India.

Pardalos, P. and Wolkowicz, H. (1994) 'Quadratic assignments and related problems', *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 16.

Poduri, S. and Sukhatme, G.S. (2004) 'Constrained coverage for mobile sensor networks', *Proceedings IEEE International Conference Robotics and Automation*, New Orleans, LA, USA, pp.165–172.

Rabaey, J.M., Ammer, M.J., da Silva, Jr., J.L., Patel, D. and Roundy, S. (2000) 'PicoRadio supports ad hoc ultra low power wireless networking', *IEEE Computer*, Vol. 33, pp.42–48.

Shih, E., Bahl, P. and Sinclair, M. (2002) 'Wake on wireless: an event driven energy saving strategy for battery operated devices', *Proceedings ACM/IEEE International Conference Mobile Computing and Networking*, Atlanta, GA.

Sohrabi, K. and Pottie, G.J. (1999) 'Performance of a novel self-organization for wireless ad-hoc sensor networks', *Proceedings IEEE Vehicular Technology Conference*.

Xue, F. and Kumar, P.R. (2004) 'The number of neighbors needed for connectivity of wireless networks', *Wireless Networks*, Vol. 10, No. 2, pp.169–181.