An Interactive Genetic Algorithm for Mobile Sensor Networks

Ali NOROUZI, Faezeh SADAT BABAMIR, A. HALIM ZAIM

Department of Computer Engineering, Istanbul University, Avcilar, Istanbul, Turkey,
norouzi@cscrs.itu.edu.tr

Abstract: In this paper, we describe an interactive approach to design mobile sensor networks. The node-mobility aspect requires an online or interactive algorithm to determine the optimal network-coverage solution for a given area of interest. Hence, we develop a real-time genetic algorithm to find the suitable direction of node locomotion, considering either coverage of the target area or estimation of the optimum energy consumption. The main purpose is to provide a solution that can extend the network lifetime. The simulation results indicate that the proposed fitness function achieves our objectives.

Keywords: Mobile Sensor Networks, WSN, Genetic Algorithm, Network Life time, Optimum Energy Consumption.

1. Introduction

Wireless sensor networks (WSNs) include numerous unsupervised devices capable of sensing, computation and communication. These energy-restrained devices are expected to be used for many different kinds of applications [1]. For instance, WSNs can be used for environment and habitat monitoring, traffic measurement on roads, vehicle tracking and personnel tracking inside buildings. Even though WSNs have a variety of applications, their deployment usually has two common objectives: (a) obtaining the maximum area coverage for a specific number of nodes and (b) prolonging the operational life of the individual nodes [2].

A mobile sensor network is a WSN with locomotion capabilities, consisting of several nodes with sensing, computation and communication functions [1]. This mobility aspect presents a design challenge in unknown environments. A genetic algorithm (GA) could be an innovative approach to simultaneously optimise coverage and lifetime problems. Network lifetime could be defined as the period of time it takes for the first node, or a fraction of all the nodes in the network, to be depleted of their energy.

The aforementioned GA method is applicable for the optimisation of both dynamic environments and dynamic network topologies [3]. In this paper, we develop a real-time GA in order to design a network with maximum coverage and minimum energy consumption, which results in extended network lifetime. In most cases, a basic sensor node consists of five main components: (a) a power supply that is considered to be the only energy source, (b) a controller with memory, (c) a sensing device, (d) a communications system and (e) a mobile platform for mobile wireless sensor nodes. All these constituent parts of the architecture consume energy; especially during the spreading procedure of mobile sensor nodes, which adjusts node position with a trade-off between area coverage and energy usage [4].

A GA is a heuristic search technique that is used to automatically find optimal solutions, while trying to avoid local maxima [5]. This method is inspired from nature and has numerous applications in model checking [3]. Also, it is suitable for solving non-linear optimisation problems and for finding the probable global optimisation value of a fitness function. Fundamentally, a GA comprises three important components: recombination, mutation and a fitness function. Many researchers have concentrated on the fitness function, which operates on chromosomes. In our paper, as in [6], we also propose a fitness function (the main procedure of GAs) to estimate an optimal solution. The optimal solution(s) is selected according to two important parameters. Energy consumption is one the most important parameters for measuring the efficiency of network positioning and can be divided into three parts: the energy usage of transmission for each node, receiving packets by cluster head and data transmission from cluster head to sink. In this paper, we assume that the energy consumption of transmission for each node represents the energy usage. Network coverage is another important parameter to consider in measuring the size of a chromosome. In this work, the size of a chromosome depends on its energy consumption and network coverage [7].

The remainder of this paper is organised as follows: (a) Section 2 reviews related work, (b)

Section 3 shows our contribution, (c) in Section 4 we propose a solution and elaborate on details of our algorithm, (d) Section 5 presents our simulation and an analysis of our results and (e) Section 6 summarises the conclusions.

2. Related Work

In this section, we describe some examples of previous research on network coverage optimisation problems. One of the first research works on the coverage optimisation problem was presented by Gage [8] in 1992. He proposed a blanket coverage method to achieve a statistical arrangement of sensor nodes leading to an extended coverage of an area, similar to that of interest in the current work.

Howard et al. [9] presented a GA-based approach that used a repulsive behaviour to spread the nodes throughout the resource. Here, a virtual "force" algorithm was utilised for node placement in order to avoid the problem of local-optimum solutions encountered by many researchers [10, 12]. They considered a complete model network in which only offline planning is required for deployment.

Kalpakis et al. [10] proposed the Maximum Lifetime Data gathering Algorithm (MLDA), which is able to maximise the lifetime of a network with fixed locations of the nodes and the base station. This algorithm finds the edge capacities that allow maximum transmission flow by running a linear program.

The PEGASIS protocol, an extension of the LEACH algorithm, in which every node transmits aggregated data to its nearest neighbour and the transmission is repeated until the data packet is delivered to the base station[13]. The PEGASIS method provides an advantage over LEACH in its robustness to node failure. Pan et al. [14] subsequently presented a two-tiered structure that improves energy efficiency by local hierarchical clustering.

Later, Hussain et al. [12] proposed an approach using a GA method to obtain an optimum solution, including energy consumptions, transmission schedule, etc. Qu et al. [2] introduced a sensor relocation method based on a multi-objective GA. These objectives were to maximise coverage area and minimise energy usage within the network.

3. The Proposed Approach

The nature of sensor nodes and special properties of a dynamic environment challenge the optimisation problem because of two main limitations for mobile sensor networks. First, the offline design of network topology based on a static map is not appropriate for a dynamic environment with changing geographical features. Second, energy consumption is a more significant issue in dynamic networks due to the mobility of sensor nodes. The sensor nodes have limited energy for monitoring the resource and thus, in communicating with each other for long periods and moving from one place to another, a minimum energy consumption approach is highly desirable. To solve these problems, we instigated a real-time genetic algorithm, because: firstly, a real-time or more exactly, interactive GA can facilitate optimisation solutions of the dynamic environment; secondly, the algorithm takes power consumption into account when nodes are selected for movement and/or monitoring activities; and thirdly, the GA operates randomly at different levels, yielding a convergence close to the static global optimum in coverage.

3.1. Assumptions

In this work, we aim to propose a fitness function that is adapted for a real-time GA. This function operates on chromosomes that present different network topologies. A network of n nodes is represented by a chromosome of n bits or gens. Nodes of a network are represented as bits (gens) of a chromosome. Each chromosome in a population represents a possible solution of network topology. Chromosomes permute information by crossover and generate new chromosomes. In a GA, fitness is assessed by the function defining our desire goal(s). Moreover, the chance of an individual chromosome surviving in a crossover depends on its fitness value. The chance of survival is higher for better fitness values. The fitness of a chromosome is defined by several parameters, such as speed, energy consumption and single hop delay. A population includes several chromosomes. The fittest chromosomes will be chosen to produce the next population. Also, the initial population is a large number of random network topologies. The fitness function evaluates members of the current population to choose those most suitable to be transformed into the future generation. A new population is produced by replacing some members of the population [1].

In general, a GA retains a certain number of the best individuals from each generation and mixes them up to form the new generation. Therefore, the new generation will have some of the individuals from previous populations and others that have evolved through the crossover and mutation functions [1]. Sometimes, a couple of parent chromosomes are dropped, even though they have high priorities, due to an incompatible crossover, which leads to the production of a bad chromosome. In this case, such parents are marked with a tag field for use in other crossovers.

In this work, we will introduce an interactive genetic algorithm. To run this algorithm, a powerful processor is required, which can be installed in the base station. Moreover, the base station needs to know the location of each sensor. Then, the base station runs the optimisation algorithm in order to decide in which direction the sensors should move. Furthermore, in this research, a case in which each node has a scanning laser range-finder sensor and an omnidirectional camera to monitor its distance from nearby nodes and obstacles is considered.

The sensor network assumptions are listed in the following. In addition, both the sensing and multi-objective GA models are described.

3.2 Network assumption

The following assumptions are assumed for network sensors:

1. All sensors can communicate with the base station directly or by multi-hopping.
2. Sensors' coverage is defined as a circle centred on this sensor.
3. All sensors have GPS or other location devices. They can move to any position (with known coordinates) within their mobility range.
4. Sensors cannot sense through or move across boundaries and obstacles that are considered as walls.

3.3 Sensing model

The first objective considered in our algorithm is coverage. The assumed sensing model is a binary model, which is supposed to be covered as much as possible. This means that the area within a certain distance from a sensor can be counted as 100% covered and the area beyond the sensing range will be set at less than 50% covered, because it cannot be covered by this sensor [1]. The sensing field is considered to be a grid. The coverage of the whole area is proportional to the grid points that can be covered. Considering the grid points *(x,y)* and *(x_i,y_i)*, the possibility that they can be sensed by a sensor node A*s_i(x_i,y_i)* can be described by Equation 1. A*s_i(x_i,y_i)* means node A is located at *(x_i,y_i)*. *R_s* is the number of sensor nodes in the network.

$$\begin{cases} 1 \quad As_i = \dfrac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{R_s} \times 100 \therefore R_s \neq 0 \\ 0 \quad Otherwise \end{cases} \qquad (1)$$

Relation 1: Possibility of sensing points (x, y) and (x_i, y_i) by node As_i

Another objective of the function is to save the sensor's energy by minimising the distance travelled by. For a sensor network with *R_s* sensor nodes, the average distance travelled is equal to:

$$S_i = \frac{\sum_{i=1}^{N} | G_i - G'_i |}{R_S} \Leftrightarrow G_i \neq G'_i \qquad (2)$$

Relation 2: Average distance travelled

Where G(*x*, *y*) and $G'_i(X'_i, Y'_i)$ are the corresponding initial and final points of a sensor node; hence, this is the distance between grid points *(x,y)* and *(x'$_i$,y'$_i$)*. To manage all sensor nodes, we designate a selected subset of association nodes. Initially, all sensor nodes are grouped into the set called cluster, which in turn is managed by association nodes called Cluster Heads (CHs). The developed algorithm comprises five steps, as explained in Algorithm 1. This algorithm summarises the GA-based process for dynamic environmental monitoring. All nodes transmit aggregated data to the common Base Station (BS) through their corresponding CHs. The base station is a constantly powered resource that processes the algorithm and transmits the results to all nodes.

3.4 Initialisation

Initialisation is the first step of the algorithm, at which the value of *t_i* is set to initialise the algorithm. Nodes are clustered and CHs are determined. Due to the locomotion of the sensor nodes in a mobile network, we utilise the concept of the Direction Vector (DV) to specify the situation of each node, as done in previous studies [3]. The DV is an n-dimensional vector ($n \in \mathrm{N}$) scaled to the interval [0, 1]. This value indicates the probability of nodes moving in different directions. Clearly, the summation of this dimension is 1.0.

```
Deployment (Mobile Network)

1. Initialization
        1.1. Determine Direction Vector (DV_i);
        1.2. Cluster network;
        1.3. Determine Member Vectors (MV_i);
1.4. Set $t_i$ as termination parameter of GA;
1.5. Calculate current coverage:
    $\sum_{i=1}^{membernodes} \mathrm{cov}\,erage(i)$
2. Aggregation Data
2.1. Collect monitored data;
2.2. Update $MV_i$;
2.3. Consider $MV_i$ in coverage(i);
2.3. Interrupt system when episode is over;

3. Fitness
3.1. If $\sum_{i=1}^{membernodes} \mathrm{cov}\,erage(i) < \mathrm{cov}\,erage_{t\arg et}$ then
Compute updated coverage();
3.2. Assign score according to the new coverage and energy
        consumption;
4. Selection();
5. Crossover();
6. Termination
6.1. If the given $t_i$ is over or target coverage is satisfied then
        return final results;
Else, go to Step 2;
```

Algorithm 1. Proposed GA-based algorithm for mobile sensor network

Simultaneously, the GA explores node movement using the DV and the locations of the CHs. Cluster heads must know the status of each node, including its remaining power level. Therefore, we introduce a concept called the Member Vector (MV), which consists of a member-node identifier (ID) and its energy level. After monitoring the environment and collecting data from the nodes, every sensor member periodically transmits its aggregated data to the corresponding CH and then the MV will be updated. Any CH has a set of MV_i (i ∈ Number of member nodes), which allows the real-time GA to calculate the energy required to deliver the optimum network coverage.

3.5 Aggregation data

Consider Algorithm 1, step 2. This part shows the interactive or real-time property of our algorithm. Moreover, when the algorithm is processing, at each round, new information regarding the nodes and network is collected to consider the most up-dated variables for the fitness function. Therefore, the scores that are assigned to the chromosomes are based on updated variables, including MV_i.

3.6 Selection

During each successive generation, a new population is selected for mating from among the current members based on fitness. Fitter individuals are similarly rated, leading to the preferential selection of the best solution. Most of the fitness functions are designed with a stochastic part to choose some smaller, less fit members, in order to help maintain the diversity of the population [15]. Among several available selection methods, the Roulette Wheel was chosen to distinguish appropriate individuals, with a probability given by:

$$P_i = \frac{F_i}{\sum_{j=1}^{n} F_j} \qquad (3)$$

Relation 3. Selection method of proposed algorithm

Where F_i and n are the chromosomal fitness and population size, respectively. According to the Roulette Wheel, each assigned value in the continuous numerical interval is between 0 and 1.

3.7 Fitness function

The fitness function is one of the most essential components of a GA. Using this function, a score can be assigned to any chromosome to distinguish those that are fittest among the population. Individuals of high quality survive for the next generation. Also, along with a small group of low-value members, they are the parents of subsequent generations. In our case, the score awarded by the fitness function is dependent on two different objectives. The first is the coverage criterion, which acts to increase the score and the second is the amount of energy spent in order to cover and monitor the area. Higher energy consumption decreases the total fitness value; therefore, the fittest members are those that are able to make an appropriate trade-off between the two objectives. The proposed fitness function based on the coverage and energy criteria is given as:

$$\text{Fitness}(x) = \textstyle\sum^{n}_{j=1} [(\text{coverage}_{x,j,e} – \text{coverage}_{x,j,e-1}) \\ – (\text{energy}_{x,j,e} – \text{energy}_{x,j,e-1})] –\text{energy}_{CH}*p \qquad (4)$$

Relation 4: Proposed fitness function

Where *coverage* is the area covered, *energy* is the sum of the remaining power in the nodes and *p* is the total number of data packets. The index *n* indicates the number of active nodes taking part in the transmission of data packets and *e* is the step counter. This function was formulated in an attempt to consider all conditions and thus, allow a comprehensive comparison among the chromosomes.

3.8 Termination

Consider Algorithm 1. This algorithm interactively (in online fashion) calculates MV_i, which should yield t_i at the maximum. T_i is determined by the user (in this algorithm, step 1.4.) and depends on the desired objectives, i.e., further generations could obviously be used to estimate a more accurate solution but according to the user's opinion, this value can be determined. However, as the network locomotion is live, iteration should be limited. Sending MV_i to the corresponding member nodes requires energy to transmit and loads the new MV. This task might be done at long intervals to conserve energy and thus, extend the network lifetime. Finally, the algorithm is terminated when a given *t* value is reached or the entire target coverage is obtained.

4. Implementation

To implement the algorithm, we utilise a network simulator to assess the proposed algorithm in two respects. First, the GA-based portion can be implemented by a Java editor. In this case, the Java Genetic Algorithm Package (JPAC) should be installed to test the algorithm in a consistent manner. Next, OMNET++ can be utilised to trace the movement of the nodes in a virtual environment (Figure 1). The parameters are summarised in Table 1.

Figures 2 and 3 show a comparison between the proposed algorithm and those using the LEACH (Low Energy Adaptive Clustering Hierarchy) protocol with respect to network energy and lifetime over 200 time units (years).

Figure 1. Example of virtual environment

Table 1. Parameters of the tested virtual environment

Network size	100 m × 100 m
Node numbers = n	200
Initial energy	2J
Node energy	50 nJ/bit
Uncovered area	10 × 10 m^2
BS distance	200 m
Packet size	200 bits

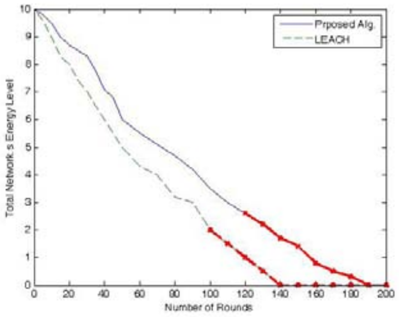

Figure 2. Energy consumption rate over the lifetime of a virtual environment

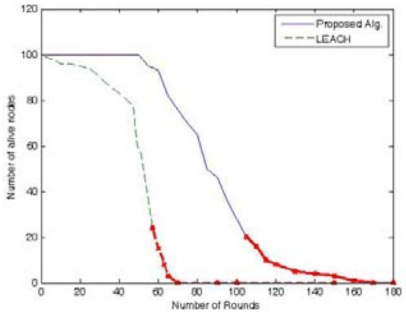

Figure 3. Comparison of coverage between proposed and LEACH algorithms

In Figure 2, the unified energy consumption by the CHs results in a short lifetime in the LEACH protocol. Also, this figure shows that the removal time for the first node (the death of the first node), due to its low-energy status, is delayed compared with the LEACH protocol. Additionally, the network remains in working order as long as a minimum number of nodes are alive. Generally, due to use of the algorithm fitness function that considers the energy status of the nodes and the distance between the CHs and BS, the remaining individuals yield a cluster formation with uniform energy consumption that significantly extends the lifetime of the network. Figure 3 shows a coverage comparison between LEACH and the proposed algorithm over time. Moreover, the topology discovered by our algorithm covers the total environment until round 60. After that, the coverage will be gently reduced until round 180. As shown, the beneficial coverage rounds are 100 more than the LEACH algorithm and complete coverage lasts until round 60.

5. Conclusions and Future Work

In this work, we presented a method that utilises an interactive GA. Moreover, an online approach is used to direct the locomotion sensor nodes to determine optimal locations through consideration of the amount of remaining power. Finally, the obtained solution simultaneously maximises environmental coverage and minimises power-consumption metrics as objectives. The simulation results confirm that the proposed fitness function fulfils these objectives. Also, it is shown that the network lifetime has been extended and coverage improved compared with the LEACH protocol. As future work, we plan to investigate a GA-based algorithm that requires fewer transmitted directions over longer time intervals to further reduce energy costs incurred by receiving and loading new directions.

REFERENCES

1. POURKABIRIAN, A., A. T. HAGHIGHAT, **Energy-Aware, Delay-Constrained Routing in Wireless Sensor Networks through Genetic Algorithm**, IEEE.
2. QU, Y., S. V. GEORGAKOPOULOS, **Relocation of Wireless Sensor Network Nodes Using a Genetic Algorithm**, 978-1-61284-080-2/11/IEEE, 2011.
3. SUEN, Y., **A Genetic Algorithm based Mobile Sensor Network Deployment Algorithm**, Technical report, Department of Electrical and Computer Engineering, The University of Texas at Austin, 2004.
4. NOROUZI, A., A. SERTBAS, **An Integrated Survey in Efficient Energy Management for WSN using Architecture Approach**, International Journal of Advanced Networking and Applications, vol. 3, no. 1, 2011, pp. 968-977.
5. GOLDBERG, D., B. KORB, K. DEB, **Messy Genetic Algorithms: Motivation, Analysis, and First Results**, The Clearinghouse for Genetic Algorithms (TCGA), Report 89003, 1989.
6. CANTU-PAZ, E., **A Survey of Genetic Algorithms**, Calculateurs Paralleles, Reseaux et Systems Repartis, vol. 10(2), 2002.
7. ROMOOZI, M., M. VAHIDIPOUR, M. ROMOOZI, S. MAGHSOODI, **Genetic Algorithm for Energy Efficient and Coverage-preserved Positioning in Wireless Sensor Networks**, in Proceedings of IEEE Conference, ICICCI 2010, pp. 22–25, DOI:10.1109/ICICCI.2010.10
8. D GAGE, **Command Control for Many-Robot Systems**, Unmanned Systems Magazine, vol. 10, no. 4, 1992, pp. 28–34.
9. HOWARD, A., M. MATARÍC, G. SUKHATME, **Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem**, in International Symposium on Distributed Autonomous Robotics Systems (DARS02), June 2002.
10. KALPAKIS, K., K. DASGUPTA, P. NAMJOSHI, **Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks**, in IEEE International Conference on Networking, August 2002, pp. 685–696.
11. DASGUPTA K., K. KALPAKIS, P. NAMJOSHI, **An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks**, in IEEE Wireless Communications and Networking Conference, vol. 1, 2003, pp. 1948–1954.
12. HUSSAIN, S., A. W. MATIN, O. ISLAM, **Genetic Algorithm for Energy Efficient Clusters in Wireless Sensor Networks**, in Fourth International Conference on Information Technology: New Generations (ITNG 2007) , April 2007, pp. 147–154.
13. LINDSEY, S. C. S. RAGHAVENDRA, **PEGASIS: Power-efficient Gathering in Sensor Information Systems**, in Proceedings of the IEEE Aerospace Conference, vol. 3, March 2002, pp. 1125-1130.
14. PAN, J., L. CAI, Y. T. HOU, Y. SHI, S. X. SHEN, **Optimal Basestation Locations in Two-Tiered Wireless Sensor Networks**, IEEE Transactions on Mobile Computing (TMC), vol. 4, no. 5, 2005, pp. 458–473.
15. BABAMIR, F. S., A. HATAMIZADE, S. M. BABAMIR, M. DABBAGHIAN, A. NOROUZI, **Application of Genetic Algorithm in Automatic Software Testing**, in Second International Conference on Network Digital Technology, Charles University, Prague, vol. 2, July 2010, pp. 545–552.