MGA: A Solution Strategy to the Sensor Selection Problem in WSNs

Luhutyit Peter Damuut
School of Computer Science & Electrical Engineering
University of Essex
CO4 3SQ, England
pdamu@essex.ac.uk

Dongbing Gu
School of Computer Science & Electrical Engineering
University of Essex
CO4 3SQ, England
dgu@essex.ac.uk

Abstract — Selecting a finite subset out of a given universe of discuss to accomplish some tasks or meet some well-defined objective(s) is often modeled either as an optimization problem or solved through exhaustive search. While the former approach is the norm in most applications, the later is suitable for those applications where the size of the problem space is relatively small. In this paper, we consider a problem domain comprising homogeneous sensors deployed randomly for the purpose of environmental surveillance application. Specifically, we investigate the schemes applicable in determining which subset of sensors to select at any time period in order to optimize a predefined objective by employing and extending the conventional genetic algorithm (GA) technique through the incorporation of a more robust bivariate gene combination, comprising both binary and continuous values to encode chromosomes in the solution space. Simulation results show the effectiveness of this technique in the problem domain.

Keywords – Sensor, Selection, Optimization, Genetic Algorithm

1. INTRODUCTION

Sensor selection is imperative in online applications involving large volumes of sensory data that need to be integrated quickly in order for decisions to be taken with reasonable level of accuracy. In information theory [1], it is expected that more information is obtainable with increase in the number of sensors. As plausible as this assertion appears, it is well known that information gathering incurs costs ranging from hardware costs to operational costs. Many sensor applications are often constrained by limited time and resources. An important issue for online information fusion is selecting a subset of the most informative nodes, which optimally reduces the uncertainty in the parameter being measured (i.e., the event of interest) at minimal costs.

To dynamically determine the best choice of sensors to select requires the complete enumeration and evaluation of all the possible subsets of the total number of nodes in the network, which is computationally very expensive and infeasible especially when the size of the nodes is large. Moreover, searching for an optimal subset is a well-known NP-hard problem, because the search space increases exponentially with the number of nodes [2]. To find a solution to this type of problem, it is often reduced to that of single optimal sensor selection and then greedy heuristics applied. While this method could be efficient, there is often no guarantee of optimality.

To effectively reduce uncertainty and cost, strategies that consider several observations before making a decision on the selection are often adopted. In [3], for example, an efficient method for identifying a subset of active sensors that are more relevant and cost effective is presented. Obviously, the task of selecting any ideal combination from a finite pool of possibilities is a well-researched area due to its relevance in everyday life. In fact, this problem is widely known as an optimization problem which has many forms and flavours depending largely on the application area. Note that most optimization problems work well with continuous values. And when an optimization problem has a combination of both integers and continuous values, it is referred to as mixed integer optimization which can be solved using several branch and bound techniques [4]. In this paper we are interested in modelling the sensor selection problem leveraging on GA techniques. Therefore, it is pertinent to define some key terms associated with GA.

a.) Population: Population refers to a group of chromosomes representing a complete set of solution to a defined problem. In some literature, the chromosomes are called individuals each encoded differently depending on the problem domain. In most GA implementations, the initial population is often randomly generated from where newer ones evolve, usually through a combination of different processes such as selection, cross-over and mutation.

b.) Fitness: In genetics, an organism’s fitness refers to its ability to pass on its character traits to its offspring, while in GA, fitness is measured by a function which defines the problem to be solved. The chance of an individual to mate largely depends on its fitness value.

c.) Selection: Selection is the process of choosing from the current population, which chromosomes to mate, mutates or retain in the next generation. When two parent chromosomes are selected to mate, new offspring is created consisting of the genes from both parents. Fitter chromosomes stand better chances of selection for mating or retention in the next generation than less fitter ones, depending on the selection criteria in use. In this paper, we consider Tournament selection [5], where two
needed to keep the network functional is highly crucial in number of active nodes, at any given time, to those strained by energy limitations, therefore limiting the stem from the fact that sensor networks are highly con-

The justification for sensor selection algorithms often include heuristics designed to solve coverage optimization problems [8], target tracking and localization [9], maximum information gain [1], minimum mean-squared error solutions [10], among others.

In [11], a decision-theoretic solution to the sensor selection problem is presented where a Partially Observable Markov Decision Processes (POMDPs) is used as a frame-work for sequential decision making under uncertainty. In particular their work is based on selecting a small subset of a set of camera sensors used to track a moving person.

Modelled as a relaxed form of a convex optimization problem, SSP has been solved using centralized or decentralized approaches [12]. The authors of this work propose a decentralized heuristic approach to select a subset of sensor readings and measure the performance of their model using the volume of the covariance matrix of the estimation error. This approach for sensor selection could be unsuitable for applications involving sensors with limited processing power.

The authors of [13] proposed the application of binary particle swarm optimization techniques to the problem of selecting a subset of sensors from a large set of sensors for the purpose of minimizing the error in parameter estimation. Their motivation is based on the need for in-creased computational efficiency of parameter estimation and energy consumption of the nodes in the network.

The task of selecting sensors has been applied to state estimation of dynamic systems [14], identification of large space structures [15]. Similarly, a mobile robot localization technique based on sensor selection is presented in [16].

The researchers in [17] study the problem of sensor selection for target localization. In [18], a method for selecting an optimal set of sensors for the purpose of tracking a moving target is presented. Also, an algorithm for sensor selection is presented in [19] for estimating the health status of an aircraft gas turbine engine.

Most sensor selection problems have been proven to be NP-hard [3] because of the exponential increase in the complexity of the problems with increase in the number of sensors. In fact, the SSP is intrinsically combinatorial in all its ramifications making it infeasible to solve using exhaustive search technique. Therefore, a lot of the existing solutions to the sensor selection problems are based on heuristic optimization techniques.

Contrary to what obtains in the literature, our approach to solving the SSP leverages on GA processes namely; fitness function evaluation, crossover and mutation to choose nodes in the evolving population until ensuring the overall increase in the network’s lifespan.

The rest of the paper is organized as follows: Section II presents related literature on the subject. In section III an analysis of event detection schemes is presented, while Section IV provides a detailed description of our mixed GA solution strategy to the sensor selection problem. Section V provides the details of the implementation of our strategy together with simulation results. We thereafter conclude the paper in section VI highlighting our contributions and providing an insight into other dimensions to which the problem can be further tackled going forward.

II. RELATED WORK

Sensor Selection Problem (SSP) refers to the task of determining a subset, say k out of a complete set of n possible sensors, k, n ∈ R^+, such that a well-defined objective function is optimized. The objective function, in this context, could take various forms depending on what the network is designed to achieve and at what network layer the objective is based. Examples of objective functions include minimizing energy consumption in the network [6], maximizing the reliability of event detection [2], maximizing area coverage [7], minimizing information latency [1] among many others.

The justification for sensor selection algorithms often stem from the fact that sensor networks are highly con-

DOI 10.5013/IJSSST.a.14.05.05 35
ISSN: 1473-804x online, 1473-8031 print

...
no changes occur in the best fitness value for a predefined number of times.

III ANALYSIS OF SENSOR SELECTION SCHEMES

Consider a network of sensors that has been deployed randomly at a set of locations \( V \) with the task of monitoring some phenomenon (e.g., temperature). Constraints on the nodes such bandwidth or energy typically require us to select a subset, say \( k \) of these sensors for activation, according to some utility function. The activated sensors then send their measurements to data fusion center for onward processing.

First, we review the offline sensor selection scheme in which the utility function is clearly specified in advance, showing how near-optimal sensor selection can be implemented. Thereafter, an attempt is made to address the more challenging setting in which the utility function is not specified in advance but is learned online from the sensory measurements.

A. The Offline Sensor Selection Scheme

In offline sensor selection algorithms, a set of sensors (say \( k \)) is chosen to maximize a known objective function \( Z = f(k) \), subject to some given constraint (e.g., number of sensors, \( n \leq k \)). One plausible choice of an objective function is sensing. In many applications, measurements are highly correlated allowing for predictions at the unobserved locations. Suppose, \( X_v = [X_1, \ldots, X_n] \) is a random vector over all sensory measurements in the network and let \( X_A = x_A \) be some measurements taken by a subset of the nodes, then the conditional distribution \( P(X_{\Lambda} | X_A = x_A) \) allows us to predict the values of the remaining nodes. In addition, the conditional distribution gives an indication of the uncertainty in the prediction.

Intuitively, we want to select those sensors that minimize the predicted uncertainty which is quantifiable by the mean squared error (MSE) given as follows:

\[
\text{MSE}(X_{\Lambda} | X_A) = \frac{1}{n} \sum_{a=1}^{n} (X_a - E[X_a | X_A])^2 |X_A|
\]

(1)

Where \( n \) is the cardinality of the sensors and \( A \) is the selected subset of nodes. However, because the measurements \( X_A \) is not always known in advance, we are constraint to base our optimization on the reduction in mean squared prediction error as given in (2).

\[
f_{\text{EMSE}}(A) = \text{EMSE}(\Phi) - \text{EMSE}(A)
\]

(2)

Where the term \( \text{EMSE}(\Phi) \) is the expected mean square error when no sensor is selected which by definition, has a value of 0 (zero).

Moreover, \( f_{\text{EMSE}} \) is monotonic, implying that given; \( A \leq B \leq V \), then \( f_{\text{EMSE}}(A) \leq f_{\text{EMSE}}(B) \). In other words, increasing the number of sensors correspondingly reduces the prediction error. That means, \( f_{\text{EMSE}} \) is maximized when all the sensor measurements are included. In practice, only a small set of the nodes, at most \( k \) is selected due bandwidth and energy constraints as shown in (3).

\[
A^* = \arg \max f_{\text{EMSE}}(A) : |A| \leq k
\]

(3)

This optimization problem is NP-hard, because there is no efficient ways to find an optimal solution. Greedy approaches that start with an empty set and then iteratively adds elements which maximally improves the utility to obtain a near-optimal solution.

One major challenge in offline approach to sensor selection is the requirement for the function \( f \) to be clearly specified in advance. In some applications, prior data, from pilot deployments could serve that purpose; while for others, no such prior data could be available. This leads to a situation where there is need for a decision on whether sensors need to be activated to collect data in order to learn a model, whereas the model in question is supposed to drive the sensor selection process.

B. The Online Sensor Selection Scheme

In a typical online sensor selection scheme, the objective function is not specified in advance, and needs to be learned during the monitoring process.

Suppose that we intend to monitor the environment for a specified number of time steps, say \( T \) rounds. If at each round \( t \), a set \( S_t \) of the sensors is selected to transmit their measurements to the fusion center, then the sensing quality \( f(S_t) \) of the selected nodes determines the utility obtained.

In this paper, we assume that the objective function \( f_t \) is not bounded and that we have a strategy for calculating it. Our goal is to maximize the total utility obtained over the \( T \) rounds, or if there are no consecutive improvements in the utility for a prescribed time periods.

We aim to implement a strategy for selecting the sets \( S_t \) of sensors at each round, such that after a number of rounds the average performance of the sensor selection algorithm converges.

IV. THE MGA SOLUTION STRATEGY

We consider a scenario consisting of a wireless sensors...
network that has been randomly deployed at a well-defined location for the purpose of monitoring a physical phenomenon (Fig. 1.) An obvious constraint on the sensors, such as their battery power, presents a challenge for selecting a subset of these sensors to send their data to the information sink, such that the total energy costs is minimized, thereby prolonging the network’s lifespan.

The objective function (s) could be specified in advance based on the traditional offline method or the more challenging method, where the objective function is learned online from the sensors’ data. In our case, the objective function is clearly specified in advance. As is common the literature, we define our sensor selection problem as that of selecting a subset $k \geq 1$ of active sensors out of a total (say $n >> k$) of deployed sensors. We consider a sensors network composed of randomly deployed sensors, where the $k$ active ones are allowed to transmit sensed events (such as an oil leak) to a central server or base station.

### A. Problem Definition

Our approach to solving the sensor selection problem considers an initial population $S$ randomly generated, consisting of chromosomes representing a solution space as shown in equation (1). The binary part of each chromosome is given by the $b_{ij}; \{i, j\} \in \mathbb{R}^n$, while the continuous part of each chromosome is given by the $\alpha_{ij}; \{i, j\} \in \mathbb{R}^n$.

$$s = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix}$$

Equation (4) can also be represented as follows:

$$s = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} b$$

Each of the vectors $v_i$ is related to the binary part of the chromosomes as shown in equation (6) is

$$v_i = [b_{i1}, b_{i2}, \ldots, b_{in}]; \quad b_{ij} \in \{0; 1\}$$

where each element of the vector $v_i$ represents a sensor, the dimension of which, is equal to the total number of sensors in the network. Each $b_{ij}$ whose value is equal to 1 indicates that the sensor associated with that is selected, and those with values equal to 0 indicates that those sensors are not selected. Each row’s continuous values $0 \leq a_{ij} \leq 1$ represents the part of each chromosome whose entries are obtained from a uniform random number generator and represents the readings of the corresponding sensors. The binary part of each chromosome in the initial population is also obtained from the uniform random number generator as shown in equation (7).

$$b_{ij} = \text{round}(a_{ij})$$

The objective function evaluates each chromosome and returns a fitness value as shown in equation (8).

$$f_{it} = f[b_{i1}, b_{i2}, \ldots, b_{in}, \alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in}]$$

In effect, there is a utility function $F$, associated with any particular selection of the sensors as defined by the $b_i \in \{0, 1\}$, for all chromosomes in the population expressed as follows:

$$F = \sum_{i=1}^{n} w_i \cdot b_i$$

where $w_i$ is the energy cost associated with each sensor in the sensor field defined as follows:

$$w_i = \frac{E_i}{E_o} \cdot d_i^u$$

where $E_i$ is the energy reserve of the node $i$; $E_o$ is the initial energy configuration of the node and $d_i$ is the distance of node $i$ from the data sink and is the sensor signal attenuation constant associated with the sensor in use.

Therefore, for any given $k$, our SSP is formally defined as follows:

$$\min F(S) = \sum_{i=1}^{n} w_i \cdot b_i$$

Subject to $1^T b = k$ (11)

where $1^T$ is the 1 vector, $k \geq 1$ is the given number (i.e., the constraint).

The objective function in (11) is central to the selection in our MGA solution strategy because it is used to evaluate each chromosome to determine its suitability for selection or mating to create new offspring at every
For a homogeneous network of sensors with identical start-up energy and based on equation (10), we can rewrite equation (11) as follows:

\[ \min_{\mathbf{b} \in \{0, 1\}^k} F(b) = \sum_{i=1}^{k} \frac{\sum_{n=1}^{N} \frac{d_{in}}{a}}{d_{in}} \]

subject to \( \sum_{i=1}^{k} b_i = k \)

(12)

The mating process adopted in our implementation involves random selection of two parent chromosomes in the current population and a binary mask is then used to generate the offspring as shown below:

\[
\begin{align*}
p_1 &= b_{m1}, b_{m2}, \ldots, b_{m10} \\
p_2 &= b_{n1}, b_{n2}, \ldots, b_{n10} \\
msk &= x_1, x_2, \ldots, x_{10} \\
x &= o_1, o_2, \ldots, o_{10}
\end{align*}
\]

(13)

where \( p_1 \) and \( p_2 \) are the binary part of parents 1 and 2 respectively, \( msk \) is the binary mask and \( x \) is the binary part of the new offspring generated after the first stage of the mating process.

The continuous value part of the chromosome on the other hand is crossed by swapping the continuous values of the parent chromosomes as shown below:

\[
\begin{align*}
p_{1a} &= m_1, m_2, \ldots, m_{10} \\
p_{2a} &= n_1, n_2, \ldots, n_{10} \\
x_0 &= m_1, m_2, \ldots, m_{10} \\
x_0 &= m_1, m_2, \ldots, m_{10}
\end{align*}
\]

(14)

where \( p_{1a} \) and \( p_{2a} \) are the continuous value part of parents 1 and 2 respectively, \( x_0 \) is the continuous value part of the new offspring produced by the second stage of mating \( p_{1a} \) and \( p_{2a} \) together.

The new offspring formed after the two-stage crossover is given as follows:

\[ o_f = [x \ x_0] \]

(15)

Mutation of the population is introduced by randomly selecting any of the chromosomes in the current population in order to diversify the population. Similar to the process of crossover discussed earlier, mutation in our implementation is carried out at every generation and proceeds in two different stages. In the first stage, a bit from the binary part of a selected chromosome is flipped over, while in the second stage, the continuous value part of the selected chromosome is replaced with a new random value between the bounds of the continuous values in the selected chromosome.

Suppose \( b_{mn} \) is the selected bit for mutation, then the value \( b_{mn} \) will be changed to \( b'_{mn} \) at the first stage of the mutation as shown in equation (16):

\[ b'_{mn} = \text{not}(b_{mn}) \]

The second stage of the mutation involves the replacement of the selected continuous value part of the chromosome by a new value whose magnitude lies within the bounds of the values in the continuous value part of the chromosome as shown in (17).

\[ a'_{mn} = lb + (ub - lb) \times r \]

(17)

where \( lb \) and \( ub \) are the lower and upper bounds of the continuous value part of the randomly selected chromosome, respectively. The \( r \) in (17) represents a number obtained from the standard uniform random number generator.

A typically mutated chromosome in our implementation is as shown in (18).

\[ c' = [b_{mn}, b_{m3}, b'_{m3}, a_{m1}; a_{m2}; a'_{m3}; \ldots, a_{m10}] \]

(18)

where \( b'_{m3} \) and \( a'_{m3} \) are the mutated binary and continuous value parts of the selected chromosome respectively.

The complete listing of the MGA algorithm is given as follows:

**Input:** A matrix of the Coordinates of the Sensors), \( k \)

**Output:** \( F \) (Trace of best fitness values at each generation)

**for** \( (s_i \ P) \) **do**
- Generate initial population \( P \)
- **end for**

**Calculate Fitness Value** \( f_i \)

**Initialize Generation count:** \( g \leftarrow 0 \)

**Initialize global best fitness value:** \( g_b \leftarrow 0 \)

**while** Termination criteria not satisfied **do**

- Update generation: \( g \leftarrow g + 1 \)
- Select the fitter nodes for retention in \( g + 1 \)
- Cross over population
- Mutate the population

**for** \( (s_i \ P) \) **do**
- Update Fitness \( f_i \)

**end for**
- Update the trace: \( F \leftarrow g_b \)

**end while**

V. SIMULATION RESULTS

The initial setting of the simulation consists of 10 sensor randomly deployed in a 10x10 2D topology. Each of the sensor nodes is equipped with 2.0 J of energy. The matrix \( A \), stores the coordinates of the nodes while a record of each node’s distance from the data sink is maintained. The encoding of each chromosome
representing a solution space of the nodes is accomplished by a combination of both binary and continuous values as presented in section III. The MGA algorithm was allowed to run until a near optimal solution is attained, where there are no successive improvements in the best fitness value for a predefined number of times. This served as the stoppage criteria for the algorithm.

The mutation rate was fixed at 0.1% of the chromosomes. The result presented in Fig. 2, depicts a sample run of the MGA algorithm which only stops when there is no change in the fitness function for a fixed number of consecutive times. Although no maximum number of iterations is set at which to terminate the algorithm run, it is obvious from the figure that the algorithm converged after attaining about 5000 iterations.

We investigated the relationship between the mean fitness value and the number of nodes in the network while maintaining a fixed 10 x 10 sensor field topology, our algorithm showed no noticeable improvement in the fitness value after the number of nodes increased to 40. This implies that for this scenario, it is not advisable to use more than 40 nodes, otherwise, there is bound to be redundancy [20] in the network with significant consequences. In other words, deploying in excess of forty nodes, in this scenario, amounts to wastage of resources. Fig. 3 succinctly depicts the outcome of this finding.

While the mean values of the best fitness values remain relatively stable with increase in the number of deployed nodes in the network, there is a progressive decrease in the level of uncertainty prevalent in the simulated sensor readings as shown in Fig.4.

As expected, the time taken for the algorithm to converge is of the order of magnitude $n$. This relationship is shown in Fig. 6, where the average time for the algorithm to converge is proportional to the number of nodes in the network.
The average convergence time was computed after running the algorithm 10 times for each value of $n$ (i.e., the number of nodes).

In terms of the root mean square error, Fig. 7 shows how this is related to the level of uncertainty in the simulated sensor readings as the number of nodes in the network varies from 10 to 120.

This figure (i.e., Fig. 7) confirms that these two parameters are very closely related and are quite useful for measuring the performance of any optimization algorithm. Moreover, the result presented in Fig. 7 corroborates our earlier assertion that for this scenario, any further increase in nodes beyond 40 is not advisable, since no significant change in the level of uncertainty and hence, the root mean square error in the sensor readings, is noticeable beyond this point.

VI. CONCLUSION

In this paper, a mixed genetic algorithm optimization strategy for sensor selection has been proposed and implemented. The sensor selection problem was modeled as an optimization problem and solved using a variant of genetic algorithm implementation, where each chromosome in the population is encoded using both binary and continuous values. The binary part of each chromosome comprises of bit strings wherein a bit value of 1 signifies a selected node while a bit value of 0 indicates that the node associated with that bit is not selected in the current generation.

We showed through extensive simulations that it is possible to use mixed genetic algorithm to solve the sensor selection problem. In particular, we showed the positive correlation (i.e., the relationship) that exists between the mean fitness values and the root mean square error of the sensory readings of the selected nodes as the number of nodes is varied within a fixed, randomly deployed network topology. This positive correlation corroborates the position, shown in our earlier work [20] on redundancy in randomly deployed wireless sensor networks. Our choice of the Tournament selection scheme during the implementation of our algorithm stems from the fact that sensory measurements are intrinsically error-prone hence there is no guarantee, even under ideal conditions.

It must be stated however, that this paper does not consider the case of heterogeneous nodes (e.g., with the nodes equipped with different power configurations) in the network and where the population either shrinks or increases as it evolves from generation to generation.

ACKNOWLEDGMENTS

This work was funded in part by Petroleum Technology Development Fund (PTDF) under grant number 10PHD056 and in part by the University of Jos.

REFERENCES


