Joint Optimization of Local and Fusion Rules in a Decentralized Sensor Network

Nithya Gnanapandithan, Balasubramaniam Natarajan
WiCom Group, Department of Electrical and Computer Engineering, Kansas State University, USA
Email: nithya@relmbk.com, bala@ksu.edu

***Abstract*— Decentralized sensor networks are collections of individual local sensors that observe a common phenomenon, quantize their observations, and send this quantized information to a central processor (*fusion center*) which then makes a global decision about the phenomenon. Most of the existing literature in this field consider only the data fusion aspect of this problem, i.e., the statistical hypothesis testing and optimal combining of the information obtained by the local sensors. In this paper, we propose a Parallel Genetic Algorithm (PGA) for optimizing the probability of global detection error performance of a parallel decentralized sensor network. Specifically, we use the PGA to simultaneously optimize both the fusion rule and the local decision rules. We show that our approach provides results comparable to those obtained by using a GA and gradient-based algorithm from previous work by Aldosari and Moura, with reduced complexity. We consider both the cases of identical (homogeneous) and non-identical (heterogeneous) sensors and demonstrate that our algorithm converges to the same optimal solution in both cases. We also discuss the effect of the quality of the initial solution on the convergence of the PGA.**

***Index Terms*— Decentralized sensor networks, distributed detection, optimal fusion rule, genetic algorithms**

I. Background

Distributed detection networks are detection schemes where group-decision making is employed. In other words, a number of entities are collectively used in the decision making process. The obvious advantage of such a scheme would be the increased reliability and the redundancy inherent in it. One area where distributed detection is widely used is in sensor networks.

Sensor networks are collections of individual or local sensors that observe a common phenomenon and collectively produce some globally meaningful information. Sensor networks have a wide array of applications including military, scientific, industrial, health-care, agriculture, and domestic applications. Traditionally, multi-sensor systems consisted of a number of local sensors which sense the common observation and communicate all their data to a central processor, which then performs optimal decision making using some conventional technique. Such a system is known as a centralized multi-sensor network. One of the challenges faced in the design of such centralized sensor networks is the limited power available in the sensors and the communication bandwidth constraints. One way of reducing the bandwidth requirement is to perform some preliminary processing of the data at each local sensor and then send the condensed information to the central processor (*fusion center*). Decentralized sensor networks is the name given to such networks which are becoming increasingly popular. The reasons for this popularity are the relatively low cost of sensors, the redundancy inherent in multiple sensors, the availability of high speed communication networks, and increased computational capability [1]. These advantages have lead to significant research activity in this area [2].

The three major topologies used for decentralized signal processing are the *parallel*, *serial* and *tree* topologies [1]. Figure 1 shows the general structure of a decentralized sensor network with parallel topology. Here, H is the phenomenon that is being observed by the sensors S_1 through S_N. The observation of sensor s_i is denoted by y_i and u_i is the local decision that it makes based on the information in y_i. The local decisions of all the N sensors are transmitted to the fusion center which then makes the global decision u_0 based on the information from all the sensors.

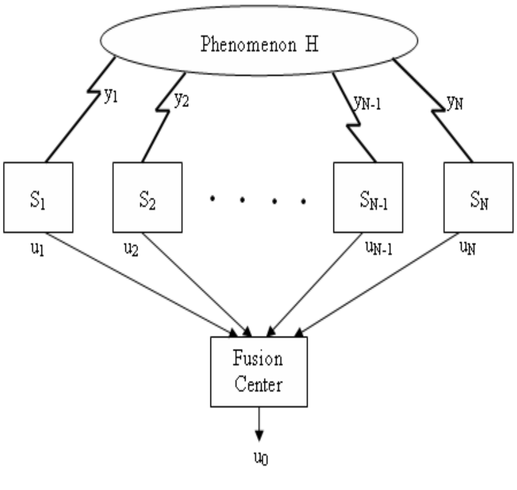

Fig. 1. Distributed sensor network - Parallel topology

Throughout this thesis, we consider a distributed sensor

This paper is based on “Parallel Genetic Algorithm based Optimal Fusion in Sensor Networks,” by N.Gnanapandithan and B.Natarajan, which appeared in the Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, USA, January 2006. © 2006 IEEE.

network having this parallel topology.

Figure 2 shows the general form of a decentralized sensor network having serial topology. Here, we have used the same notation as in the case of the parallel topology. In serial topology, there is no fusion center as in the case of the parallel network scheme. Here, each sensor generates its decision or quantized information based on its own observation and the quantized information received from the previous sensor, i.e., the ith sensor uses its observation y_i and the information u_{i-1} from the $i-1$th sensor to generate its quantized information u_i. The first sensor in the network, S_1, uses only its observation to generate its quantized information. The decision of the last sensor S_N is taken as the global decision about which of the two hypotheses is true.

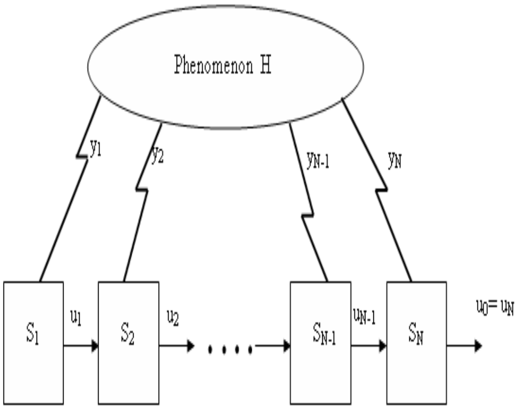

Fig. 2. Distributed sensor network - Serial topology

Figure 3 shows an example of a decentralized sensor network with tree topology. As we can see, the tree topology resembles a directed acyclic graph with the fusion center as the root of the tree. The information from all the sensors flows through a unique path to the fusion center.

As mentioned earlier, unlike the fusion center in centralized sensor networks, the fusion center in decentralized networks has only partial information about the observations. This results in a loss of performance in decentralized networks as compared to centralized networks [6]. Thus, one of the major challenges in the design of decentralized systems is to make this performance loss as small as possible by optimally processing the information at the sensors and efficiently combining them at the fusion center. This involves developing computationally efficient algorithms for processing the information at the sensors and for combining the information at the fusion center. Thus, the main objective of this work is the design of decentralized sensor networks is to find the optimal local and global decision rules assuming uncorrelated sensor observations.

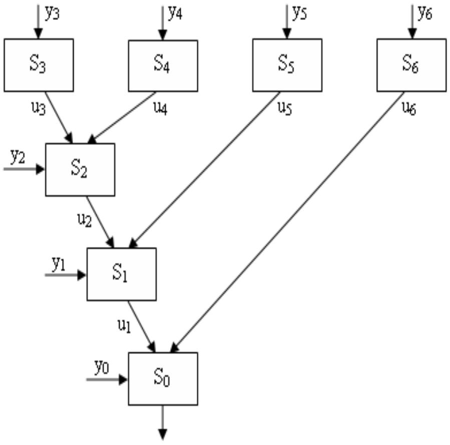

Fig. 3. Distributed sensor network - Tree topology

This paper is organized as follows: Section II provides a brief overview of related work and highlights the contribution of this paper; In Section III, we present the system model followed by the problem definition in Section IV. The optimization algorithm is detailed in Section V and simulation results are provided in Section VI. Conclusions are presented in Section VII.

II. Related Work

As mentioned in the previous section, distributed sensor networks has been an area of active research within the wireless research community. However, most of the work in this field tend to focus on the data fusion aspect of the problem, which consists of statistical hypothesis testing and combining of the information from all the local sensors. For instance, in [4], Irving and Tsitsiklis demonstrated that there is no loss in optimality if the same decision rule is used in both sensors of a two-sensor distributed network. In [7], the authors analyzed the AND and OR fusion rules for distributed sensor systems and showed that the choice between the two rules depends on the desired false alarm rate as well as the parameters of the probability distributions under both hypotheses. In [8] and [9], the authors have analyzed the constant false alarm rate (CFAR) models of distributed sensor networks. Chamberland and Veeravalli demonstrated that having a set of identical binary sensors is asymptotically optimal as the number of observations goes to infinity [5]. The authors in [10] presented an adaptive fusion model for distributed sensor networks, which estimates the probabilities of detection and false alarm by a simple counting rule. While in [11], the authors investigated the impact of various system parameters on the detection performance of decentralized sensor networks. However, most of these prior works turn to asymptotic assumptions and information-theoretic performance measures to simplify the analysis and design of sensor networks [5]- [12].

This results in the abstraction of important details of the problem such as the structure of the fusion rule. Although there are a few studies that avoid the use of asymptotic assumptions (e.g., [1], [13]), these are mostly limited to simple networks and fail to provide an insight into the structure of the optimal fusion rules. In [14], Alsodari and Moura have adopted a non-asymptotic approach to optimize both the sensing and fusion side with respect to probability of error. Their work uses a gradient-based approach for optimizing the thresholds of the local detectors and a genetic algorithm (GA) for optimizing the fusion rule. This method requires the repeated computation of the gradient along the direction of each of the $N(L-1)$ variables (N being the number of sensors and L being the number of local decision classes), until the algorithm converges. This leads to high computational cost.

In this paper, we propose a computationally efficient alternative to the method proposed in [14] using a *Parallel Genetic Algorithm*(PGA). In our algorithm, both the local thresholds as well as the fusion rule are simultaneously optimized within a single GA. We consider a parallel topology for the decentralized sensor network where, (1) the sensor observations are uncorrelated (although our PGA approach can be extended to the correlated case easily); (2) there is no communication among the local sensors, and (3) the local detectors feed their quantized decisions to a single fusion center (figure 4) through an error-free channel. In this paper, we concentrate on the design of the fusion center and the local decision rules that is optimal in a global probability of error sense. We compare our results to those obtained using the gradient-based approach outlined in [14]. Unlike [14], where only heterogeneous sensors are considered, we optimize the fusion rule for the case of both heterogeneous and homogeneous sensors. We show that both the cases of homogeneous and heterogeneous sensors converge to the same fusion rule and the same minimum probability of error. In addition to the results presented in [15], we also illustrate the effect of the quality of initial solution on the convergence of the GA. Our results show that our PGA approach converges to the same majority-like fusion rule as the gradient-based approach of [14]. The advantage of our approach is a great reduction in the computational complexity. In the next sections, we describe our decentralized sensor network model in detail and define the optimization problem that we consider in this chapter.

III. Sensor Network Model

We consider a parallel fusion network shown in figure 4, which has N local sensors and a single fusion center. The local sensors gather the measurements y_n, make a local decision u_n per sensor, and transmit these decisions to the fusion center γ_0 through an *error-free* multiple access channel (MAC). The fusion center makes a global decision $\tilde{H}$ about the true state *H* based on the set of the local decisions obtained from all sensors. Here, we assume that the fusion center itself does not sense the measurements directly.

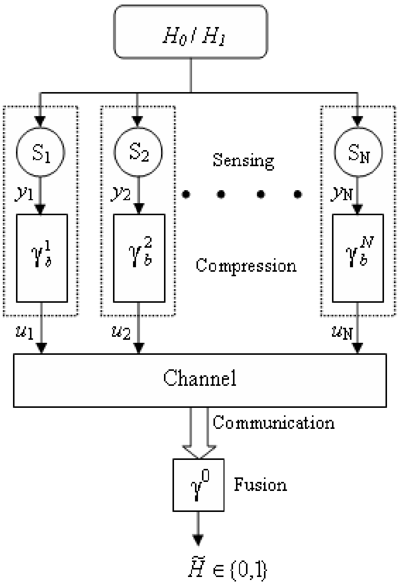

Fig. 4. Parallel fusion network

We consider the binary detection problem in this decentralized sensor network with hypotheses H_0 and H_1, with known prior probabilities π_0 and π_1, respectively. In this chapter, we assume that the observations y_n : $n = 1, 2, ..., N$ are independent and identically distributed when conditioned on $H_i, i \in \{0,1\}$. The case when the observations are not conditionally independent is considered in the next chapter.

The final output of the fusion center is binary, i.e., either H_0 or H_1. However, the local sensors are not restricted to binary outputs. Each local sensor classifies its observation y_n into one of $L = 2^b$ classes, where *b* is the number of transmitted bits per sensor. Thus, each sensor maps the observation space into a classification space that contains L classes, and, the fusion center maps the N local decisions into one of two classes, corresponding to the two hypotheses.

Each possible combination of local decisions are represented by a vector of N integers as follows

$$\mathbf{u} = (\ u_1 \quad u_2 \quad \cdots \quad u_N\ ),\ u_n \in \{0, 1, \cdots, L-1\}. \quad (1)$$

Assuming $L = 2^b$, **u** can be represented as a string of bN bits as follows

$$\mathbf{u} = (\ u_1^1 u_1^2 \cdots u_1^b\ u_2^1 u_2^2 \cdots u_2^b\ \cdots\ u_{\mathrm{N}}^1 u_{\mathrm{N}}^2 \cdots u_{\mathrm{N}}^b\ ), \quad u_{\mathrm{n}}^j \in \{0,1\} \quad (2)$$

Thus, the space of all possible local decisions is spanned by a single bN-bit integer *q*, whose value ranges from 0 to $2^{bN} - 1$. For a particular combination of the local decisions represented by *q*, the individual values of the local decisions $u_n, n = 1, 2, \cdots, N$, can be extracted by using a reverse mapping function $\Psi_n(q)$, which is defined as

$$\Psi_n(q) = \frac{q}{2^{b(N-n)}} \bmod L, \quad (3)$$

where mod is the modulo operation and all operations are carried out in integer mode.

We adopt the binary representation described in [14] to represent the fusion rules. This representation accounts for the output of the fusion rule under every possible combination of the local decisions. Since there are N sensors and each sensor classifies its measurement into L classes, each fusion rule should account for L^N local decision possibilities and, is therefore represented as a string of L^N bits as follows:

$$\begin{aligned} h = (\ h_0 \quad h_1 \quad \cdots \quad h_{L^N-2} \quad h_{L^N-1}\ ),& \\ h_q \in \{0,1\},\ q = 0, 1, \cdots, L^N - 1& \qquad (4) \end{aligned}$$

In order to optimize the global probability of error at the fusion center in this decentralized sensor network, the optimization/search has to be performed over all possible local classification rules and all possible fusion rules.

Before we attempt to jointly optimize the local and fusion rules, let us first take a closer look at the local decision rules. The conditional independence assumption on the observations, simplifies the problem since, in this case, the optimal local classifiers are likelihood ratio tests characterized by a finite number of thresholds [16], [17]. Further, if the likelihood ratio $f_1(y)/f_0(y)$ is monotonic in y [18], we can quantize the measurements themselves directly rather than their likelihood ratios. For the case where the observations are Gaussian, at most $L(L-1)/2$ quantization thresholds per local sensor are required to preserve the global optimality of the sensor network [4]. According to [14], numerical results conducted for $b = 2$ on the asymptotic regime show that optimizing a network with $L(L-1)/2$ thresholds per local sensor always converges to a simpler one having only $L - 1$ thresholds per local sensor. Hence, for the sake of simplicity, we assume that the local quantizers are characterized by $L-1$ thresholds as follows

$$u_n = \begin{cases} 0 & \text{if } y_n \le \lambda_{n,1} \\ 1 & \text{if } \lambda_{n,1} \le y_n \le \lambda_{n,2} \\ \vdots & \vdots \\ L-1 & \text{if } y_n > \lambda_{n,L-1} \end{cases} \qquad (5)$$

where, y_n is the local measurement at the nth sensor, u_n is the corresponding local decision, and $\lambda_{n,1}, \lambda_{n,2}, \cdots, \lambda_{n,L-1}$, are the $L-1$ quantization thresholds of that sensor.

Thus, one part of the problem, namely, finding the optimum local decision rules, boils down to finding the optimum set of $L-1$ thresholds. While the discussion above is restricted to the uncorrelated observation case, one can use the same approach for the correlated case as the authors have recently demonstrated in [19].

IV. Problem Definition

Problem Definition: Determine the optimum fusion rule h and the optimum local decision rules, where optimality is defined in the sense of minimizing the probability of error (P_e) at the fusion center.

Before formally discussing the optimization algorithms, let us first derive the probability of error as a function of the fusion rules and the local decision rules.

The average probability of error at the fusion center is given by the weighted sum of type-I and type-II errors, as follows

$$P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k P_k^0(\bar{k}, \lambda, h) \qquad (6)$$

where π_k is the prior probability of hypothesis H_k, $P_k^0(\bar{k}, \lambda, h) = Pr(u_0 = \bar{k}|H_k)$ is the probability of false alarm if $k = 0$ or the probability of miss if $k = 1$, and $\bar{k}$ is the binary NOT operation. Out of the L^N mutually exclusive possible local decision combinations, we sum over those that results in $u_0 = \bar{k}$ decision at the fusion center as follows

$$P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k \sum_{q=0, h_q=\bar{k}}^{L^N-1} P_k(u_1 = \Psi_1(q), .., u_N = \Psi_n(q)) \qquad (7)$$

where $P_k(u_1 = \Psi_1(q), \cdots, u_N = \Psi_n(q))$ is the joint probability of sensor 1 deciding $\Psi_1(q)$, sensor 2 deciding $\Psi_2(q)$, and so on conditioned upon H_k being true. Since, we have assumed that the local sensor observations are conditionally independent, we can write the joint probability as the product of individual probabilities as

$$P_k(u_1 = \Psi_1(q), \cdots, u_N = \Psi_n(q)) = \prod_{n=1}^{N} P_k^n(\Psi_n(q), \lambda). \qquad (8)$$

Here, $P_k^n(m, \lambda) = Pr(u_n = m|H_k)$ is the probability that the nth sensor decides m when H_k is true. Since, u_n is a quantized representation of the observation y_n, the probability $P_k^n(m, \lambda)$ is related to the $L-1$ quantization levels of the nth sensor. If we assume that the measurements are identically distributed, this probability is given by

$$\begin{aligned} P_k^n(m, \lambda) &= \int_{\lambda_{n,m}}^{\lambda_{n,m+1}} f_k(y)dy \\ &= F_k(\lambda_{n,m+1}) - F_k(\lambda_{n,m}) \qquad (9) \end{aligned}$$

where, $f_k(y)$ and is the probability density of the observation conditioned on hypothesis H_k; $F_k(x)$ is the corresponding conditional cumulative density function. Thus, eqns. 6-9 help in the evaluation of the average probability of error at the fusion center as a function of both the fusion rule and the local thresholds.

V. Proposed Approach

The problem of optimizing the decentralized sensor network over all possible fusion rules and local decision rules has been shown to be an NP-complete optimization problem when a discrete observation space is assumed, i.e., the solution cannot be determined in polynomial time [14]. The problem cannot be any easier if we consider a continuous observation space [2]. Thus, this problem has a computational complexity that increases exponentially with the number of users and, hence, it is impractical to implement an exhaustive search.

Evolutionary algorithms, such as genetic algorithms (GAs) are one among the many techniques that have been investigated to overcome this limitation. These GAs have been effective in finding approximate solutions for many NP-complete problems. A GA uses evolution and survival-of-the-fittest mechanisms to guide the search toward the fittest candidates [20].

Evolutionary algorithms have emerged as one of the most popular approaches for complex optimization problems in engineering. Evolutionary algorithms, such as genetic algorithms (GAs) have been effective in finding approximate solutions for many NP-complete problems. A GA uses evolution and survival-of-the-fittest mechanisms to guide the search toward the fittest candidates [20]. The algorithm draws upon Darwinian paradigms of evolution to search through the solution space (the set of all possible solutions). Each solution, is represented as a chromosome and comprises of smaller units called genes. Starting with a set (or population) of chromosomes, in each generation of the algorithm, new populations are created from older ones. The creation of new chromosomes is accomplished by means of two operators, recombination and mutation. Mutation is carried out by imparting a small, usually random perturbation to the chromosome. In a manner similar to the Darwinian paradigm of survival of the fittest, only "good" solutions are allowed to remain in a population with the degree of optimality ("goodness") being assessed through a measure called fitness. In our case, the fitness is measured with respect to the average probability of detection error at the fusion center.

In the following sub-sections, we present two different algorithms that can be used to solve the optimization problem at hand. The first method outlined here is the approach used in [14]. After that, we detail the algorithm that we are proposing as a computationally efficient alternative to the former.

A. GA-Stochastic Gradient (GA-SG) Approach

The GA-SG approach presented in [14] uses a GA to search for the optimal fusion rule and a gradient-based algorithm for optimizing the local thresholds. The overall algorithm is provided in Fig. 5.

Here, each chromosome in the population of the GA represents a candidate fusion rule *h*, which is represented as a string of L^N bits. A random initial population of such chromosomes is generated. The fitness of every chromosome is then calculated by optimizing the local thresholds for that particular fusion rule, and then evaluating the objective function $P_e(\lambda, h)$. A mating pool of parents are selected to undergo cross-over and obtain the offspring population. Once an offspring population is assembled, the fitness of each offspring is evaluated as before and the process is repeated till the search converges to the optimal solution. The optimization of the local thresholds for a particular fusion rule *h* is implemented by using a gradient-based approach as discussed in [14]. For each fusion rule, there is a set of $N(L-1)$ thresholds to be optimized with respect to the probability of error, which is a function of both the local thresholds λ, and the fusion rule *h*. This is a $[N(L-1)]$-dimensional nonlinear constrained optimization problem. But, instead of moving in the direction of the $N(L-1)$-dimensional gradient, each optimization step involves moving along the direction of the one-dimensional gradient with respect to one of the variables as long as the constraints are satisfied. The optimization is then carried out cyclically over all the variables.

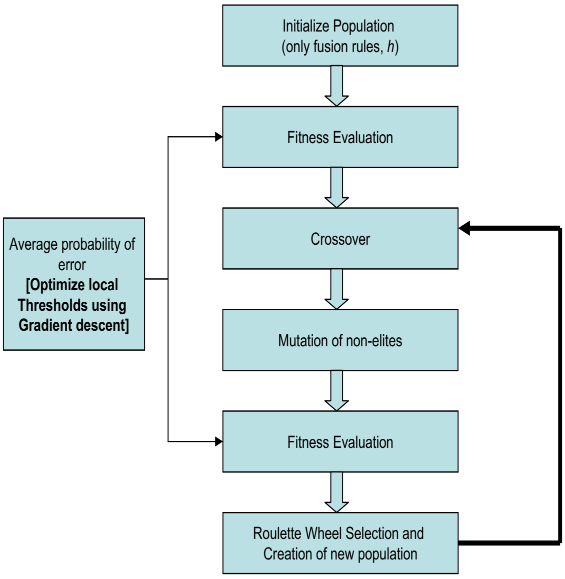
Initialize Population
(only fusion rules, *h*)
Fitness Evaluation
Average probability of error
[Optimize local Thresholds using Gradient descent]
Crossover
Mutation of non-elites
Fitness Evaluation
Roulette Wheel Selection and Creation of new population

Fig. 5. Optimization using GA-SG approach

Although the GA-SG performs well and converges to the optimal solution (see [14]), it is computationally expensive. This is due to the repeated evaluation of the gradient with respect to the $N(L-1)$ variables for each candidate fusion rule in each generation of the GA. The gradient with respect to one variable $\lambda_{\nu,\tau}$ is evaluated using the following expressions [14]

$$\frac{\partial}{\partial \lambda_{\nu,\tau}} P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k \frac{\partial}{\partial \lambda_{\nu,\tau}} P_{\mathrm{k}}^0(\bar{k}, \lambda, h) \tag{10}$$

$$\frac{\partial}{\partial \lambda_{\nu,\tau}} P_{\mathrm{k}}^0(\bar{k}, \lambda, h) = \tag{11}$$
$$\sum_{q=0, h_q=k}^{L^N-1} \frac{\partial P_k^\nu(\Psi_\nu(q), \lambda)}{\partial \lambda_{\nu,\tau}} \prod_{n=1, n\neq\nu}^{N} P_k^n(\Psi_n(q), \lambda)$$

$$\frac{\partial P_k^n(m, \lambda)}{\partial \lambda_{n,\tau}} = \begin{cases} -f_k(\lambda_{n,\tau}) & if\ m = \tau \\ f_k(\lambda_{n,\tau}) & if\ m = \tau - 1 \\ 0 & otherwise \end{cases} \tag{12}$$

where π_k is the prior probability of hypothesis H_k, $P_k^0(\bar{k}, \lambda, h) = Pr(u_0 = \bar{k}|H_k)$, $P_k^n(m, \lambda)$ is the probability that the *n*th sensor decided m when H_k is present and

$f_k(y)$ is the probability density function of y conditioned on H_k.

As can be seen in eqns. 10-12, the evaluation of the gradient is computationally expensive due to the need to iteratively calculate the summation and product terms. In the next subsection, we propose a Parallel GA approach that does not involve any such gradient evaluation.

B. Parallel GA Approach

The parallel GA (PGA) that we propose is essentially one in which we optimize both the fusion rule and the local thresholds simultaneously. Each chromosome in the GA is divided into two parts:

- the fusion rule, and,
- a set of local thresholds.

A random population is generated consisting of a group of such chromosomes. The fitness of each chromosome of the population is calculated as the probability of error $P_e(\lambda, h)$, which is evaluated using eqns. 6-9. After evaluating the fitness, both the fusion rule part and the thresholds part of the chromosomes, undergo cross-over and mutation *in parallel* to produce an offspring population. Elitism is also introduced to ensure that the best solutions in each generation are carried over without any change into the next generation. Once an offspring population is assembled, the fitness is again evaluated and the process continues till a desired termination criterion is reached. In this work, roulette wheel selection scheme is used for selecting parents for cross-over and cross-over of the non-binary thresholds part is performed as follows:

$$\lambda_{offspring} = x\lambda_{parent1} + (1-x)\lambda_{parent2} \qquad (13)$$

where x is a uniformly distributed random number between 0 and 1. It is important to note that while the crossover is performed based on eqn. 13 for both parts of the chromosome, the mutation operator is quite different for the two parts. The mutation for the fusion rule part involves a random flipping of bits in the fusion rule vector *h*. On the other hand, the mutation of the local thresholds is performed by perturbing a randomly selected threshold value by adding a Gaussian perturbation.

The overall PGA approach is illustrated in Fig. 6. From an initial look at the Figure, it appears that there is a single GA operating on the solution space. However, since the crossover and mutation operations are implemented independently and in parallel for the fusion rule part and the local threshold part of the chromosomes, one can perceive this approach as a parallel GA. It is this simultaneous/parallel search that significantly lowers the complexity of the approach as discussed next.

The advantage of this algorithm is that it greatly reduces the computational complexity, as the gradient calculations have been completely eliminated. Although, this means that it takes our parallel GA more number of generations to converge to the optimal solution, it must be kept in mind that each generation only involves the GA processes of cross-over and mutation unlike the GA-SG algorithm where, each generation also involves a complex gradient-based sub-process to optimize the local thresholds, in addition to the GA processes. Therefore, *per-generation computation of the parallel GA is much lower than that in the GA-SG algorithm.*

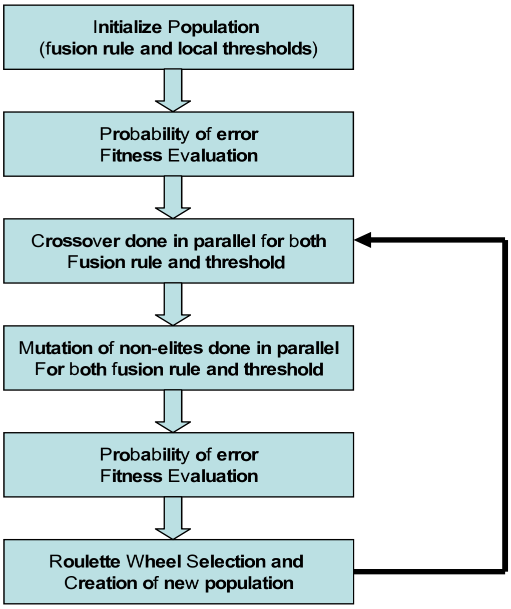
Initialize Population
(fusion rule and local thresholds)
Probability of error
Fitness Evaluation
Crossover done in parallel for both
Fusion rule and threshold
Mutation of non-elites done in parallel
For both fusion rule and threshold
Probability of error
Fitness Evaluation
Roulette Wheel Selection and
Creation of new population

Fig. 6. Optimization using PGA approach

C. Impact of Initialization

Another key point to be noted is that, for both the GA with gradient-based threshold optimization and the parallel GA, the initialization of the local thresholds as well as the fusion rule plays a crucial part in the convergence of the algorithms. This issue is not addressed in detail in [14].

The local thresholds have to be initialized close to the region of overlap between $f_0(y)$ and $f_1(y)$, which is intuitively reasonable since this is the region where it is hardest to discriminate between the two hypotheses. Additionally, the initialization of the fusion rule is also equally important. Both the algorithms are found to converge to the optimal solution sooner when the fusion rule is initialized such that the probability of getting a 1 in the rule (corresponding to deciding in favor of H_1) increases as we move from left to right along the fusion rule, i.e., the probability of the first bit (MSB) of the fusion rule being a 1 is 0 while the probability of the last bit (LSB) being a 1 is 1. This is because the first bit of the fusion rule corresponds to the case when all the sensors classify the observation as belonging to level L (highly in favor of H_1). Similarly, the last bit of the fusion rule corresponds to the case when all the sensors classify the observation as belonging to level 0 (highly in favor of H_0). Therefore, this initialization of the fusion rule is consistent with our intuition and yields the fastest convergence to the lowest probability of detection error as illustrated in the next section.

VI. Results

In this section, we present the results obtained by using both the GA-SG algorithm and our PGA approach. We consider a parallel decentralized sensor network with N sensors, each making a single measurement, quantizing it into b bits per measurement and transmitting these b bits to a central fusion center via an *error-free* multiple access channel. The fusion center then makes the global decision. We use an additive noise model $y = m_i + n$, where m_i is the signal mean under H_i and n is a zero-mean, unit variance Gaussian noise. The signal means m_0 and m_1 are assumed to be 0 and 1 under H_0 and H_1, respectively. We consider 2 cases: one where the sensors use identical quantizers (homogeneous sensors) and one where the sensors use non-identical quantizers (non-homogeneous sensors). The local observations are assumed to be conditionally independent and identically distributed. We consider sensor networks with 4 sensors, each transmitting 2 bits per measurement. Thus, each sensor quantizes its measurement into one of 4 classes using 3 thresholds. The 3 local thresholds for each sensor are initialized close to -0.5, 0.5 and 1.5, respectively. This corresponds to the region of overlap of the distributions under the 2 hypotheses. The initial population of fusion rules is initialized such that the probability of getting a 1 in the fusion rule increases from 0 to 1 as we move from the least significant bit (LSB) to the most significant bit (MSB). The prior probability π_0 is assumed to be 0.6 for all the cases.

Figure 7 shows the evolution of the global probability of error of the sensor network over 100 generations using the GA-SG algorithm. The sensors for this case are assumed to be heterogeneous, i.e., they have non-identical thresholds. As in [14], the population size is set at 1000 chromosomes while the crossover and mutation rates are 0.45 and 0.01, respectively. The algorithm for this case is found to converge after 45 generations.

Figure 8 shows the evolution of the probability of error for the same heterogeneous sensor case using the PGA approach over 4000 generations. The algorithm converges to a minimum after 2500 generations. *Although this is a much large number of generations as compared to that in the GA-SG algorithm, we must recall that the per generation computational complexity of the PGA algorithm is lower than that of the GA-SG by a large degree.* As explained in section V, this is due to the absence of the complex gradient-based sub-process in the PGA approach. The crossover and mutation rates for the PGA are set at 0.45 and 0.03 for the fusion center binary GA. For the local thresholds which are non-binary, the mutation rate is 0.05.

It should be noted that both the GA-SG algorithm and the PGA converge to the majority-like fusion rule described in [14], where the integer sum of all the local decisions is compared to a threshold given by

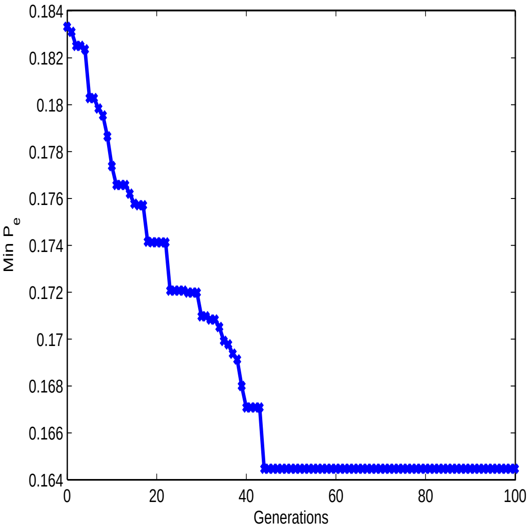

Fig. 7. Evolution of probability of error using GA-SG Algorithm for non-homogeneous sensors ($N = 4, L = 4, \pi_0 = 0.6$)

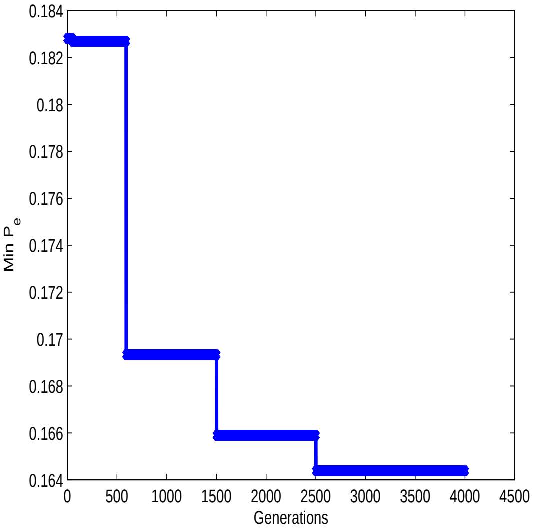

Fig. 8. Evolution of probability of error using PGA Algorithm for non-homogeneous sensors ($N = 4, L = 4, \pi_0 = 0.6$)

$$\lambda^0 \simeq \frac{1}{2}N(L-1) \tag{14}$$

where N is the number of sensors and L is the number of quantization levels per sensor. The fusion center decides in favor of hypothesis H_1 if the sum is greater than this threshold and in favor of H_0 otherwise.

Figure 9 shows the convergence of the probability of error for the homogeneous sensor case using the GA-SG algorithm over 500 generations. The algorithm is found to converge to the minimum after about 360 generations as opposed to the 50 generations in the case of heterogeneous sensors. Thus, it may seem that it is better to use heterogeneous sensors as the convergence is much faster in that case. But, we must bear in mind

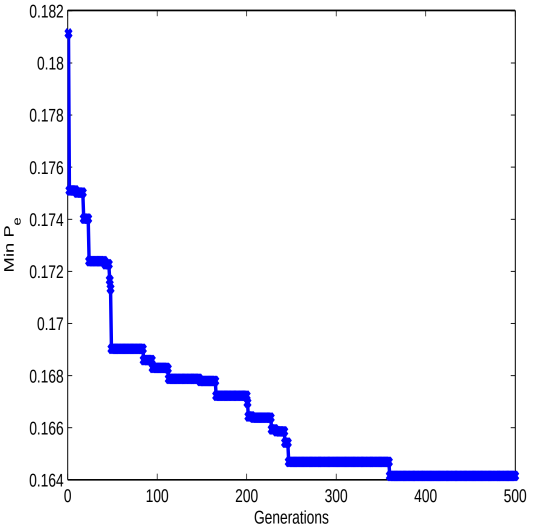

Fig. 9. Evolution of probability of error using GA-SG Algorithm for homogeneous sensors ($N = 4, L = 4, \pi_0 = 0.6$)

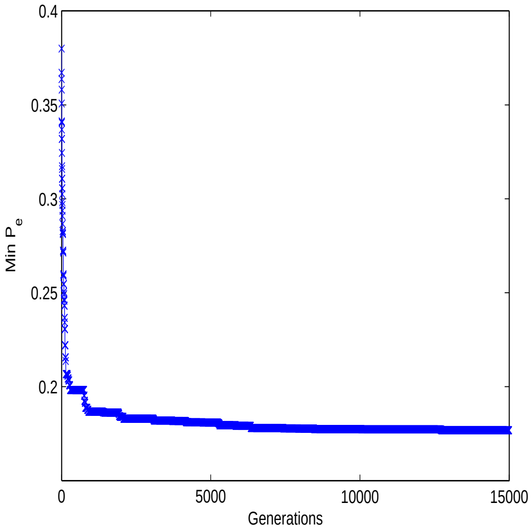

Fig. 11. Evolution of probability of error using GA-SG Algorithm for homogeneous sensors without proper initialization of fusion rule

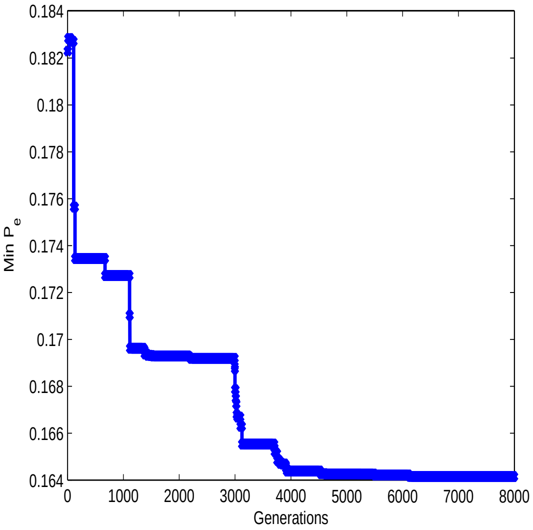

Fig. 10. Evolution of probability of error using PGA Algorithm for homogeneous sensors ($N = 4, L = 4, \pi_0 = 0.6$)

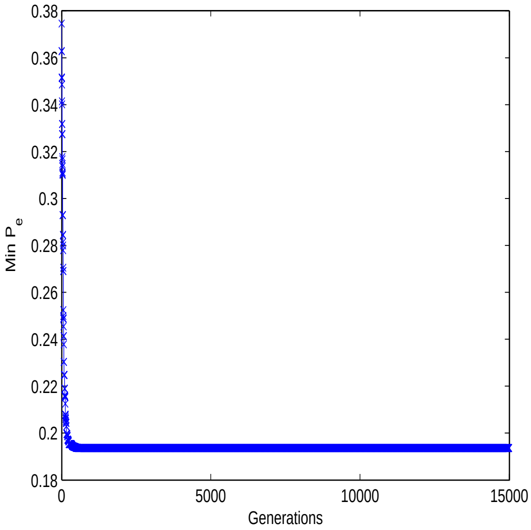

Fig. 12. Evolution of probability of error using PGA for homogeneous sensors without proper initialization of fusion rule

that the per generation computational complexity is much lower in the homogeneous sensor case as there are only L thresholds to optimize using the gradient-based sub-algorithm as opposed to the $N(L-1)$ thresholds in the heterogeneous sensor case.

Figure 10 shows the convergence plot of the probability of error for the same homogeneous case using the PGA approach over 8000 generations. Similar to the heterogeneous sensors case, both the PGA and GA-SG algorithms converge to the same majority-like fusion rule for the homogeneous case also. The PGA for this case is found to converge to a minimum after 6000 generations as compared to the 2500 generations in the case of optimizing heterogeneous sensors using the PGA. Even though the algorithm converges slower for homogeneous sensors as in the case of the GA-SG algorithm, the difference is not as pronounced in this case as in the GA-SG case. This is because, the per generation computational complexity is not affected much by the number of local thresholds in the case of the PGA as the thresholds are also optimized using the GA and not by a separate sub-process.

Figures 11 and 12, show the evolution of the probability of error plots for the GA-SG and PGA algorithms, respectively, when the fusion rule is not initialized properly, as explained earlier. For both cases, we considered homogeneous sensors and all other parameters are the same as before. The only difference from the previous simulations is that we used a completely random population for the fusion rule. As we can see, both the algorithms converge to an error probability of around 0.2 and do not reach the optimal minimum probability of error of 0.16, that was obtained in the previous simulations (with proper initial populations). Thus, choosing a good initial population for the GA plays an important role in the proper convergence of the GA.

VII. Conclusions

In this paper, we propose a Parallel Genetic Algorithm approach for optimizing both the fusion rule and local decision rules simultaneously in a probability of global detection error sense. We compare our results to those obtained using the gradient-based approach outlined in [14]. Our results show that our PGA approach converges to the same majority-like fusion rule and minimum probability of error as the gradient-based approach of [14] with greatly reduced computational cost. We optimize the fusion rule for the case of both heterogeneous and homogeneous sensors and show that our algorithm converges to the same optimal solution for both cases. We also analyze the effect of the quality of initial solution on the convergence of the GA. We conclude that the algorithm converges to the optimal solution if the initial population of the GA is selected appropriately. The local thresholds have to be initialized close to the region of overlap of the two hypotheses and the fusion rule has to be initialized such that the probability of deciding in favor of hypothesis H_1 increases as the number of sensors deciding in favor of H_1 increases and vice-versa.

References

[1] P. K. Varshney, *Distributed Detection and Data Fusion*. New York: Springer-Verlag, 1996.

[2] J. N. Tsitsiklis and M. Athans, "On the complexity of decentralized decision making and detection problems," *IEEE Transactions on Automatic Control*, vol. 30, pp. 440–446, 1985.

[3] E. Drakopolous and C. C. Lee, "Optimum multisensor fusion of correlated local decisions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, pp. 593–606, July 1991.

[4] W. W. Irving and J. N. Tsitsiklis, "Some properties of optimal thresholds in decentralized detection," *IEEE Transactions on Automatic Control*, vol. 39, pp. 835–838, 1994.

[5] J. F. Chamberland and V. V. Veeravalli, "Decentralized detection in sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, pp. 407 – 416, Feb. 2003.

[6] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part i - fundamentals," in *Proceedings of the IEEE*, vol. 85, no. 1, Jan. 1997, pp. 54 – 63.

[7] A. M. Aziz, M. Tummala, and R. Cristi, "Optimal data fusion strategies using multiple-sensor detection systems," in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, vol. 1, 2-5 Nov. 1997, pp. 941 – 945.

[8] M. Barkat and P. K. Varshney, "Decentralized cfar signal detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, no. 2, pp. 141 –149, Mar. 1989.

[9] W. Liu, Y. Lu, and J. Fu, "Data fusion of multiradar system by using genetic algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 601 – 612, April 2002.

[10] N. Mansouri and M. Fathi, "Simple counting rule for optimal data fusion," in *Proceedings of IEEE Conference on Control Applications*, vol. 1, June 2003, pp. 1186 – 1191.

[11] S. E. Aldosari and J. M. F. Moura, "Detection in decentralized sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2004, pp. 277–80.

[12] J. N. Tsitsiklis, "Decentralized detection by a large number of sensors," *Mathematics of Control, Signals, and Systems*, vol. 1, no. 2, pp. 167 – 182, 1988.

[13] Z. B. Tang, K. R. Pattipati, and D. L. Kleinman, "An algorithm for determining the decision thresholds in a distributed detection problem," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 231 – 237, Jan/Feb 1991.

[14] S. A. Aldosari and J. M. F. Moura, "Fusion in sensor networks with communication constraints," in *Proceedings IEEE/ACM Symposium on Information Processing in Sensor Networks (IPSN'04)*, April 2004, pp. 108–115.

[15] N. Gnanapandithan and B. Natarajan, "Parallel genetic algorithm based optimal fusion in sensor networks," in *Proceedings of IEEE Consumer Communications and Networking Conference Special session*, ser. Accepted for Publication, Jan. 2006.

[16] A. R. Reibman, "Performance and fault-tolerance of distributed detection networks," Ph.D. dissertation, Department of Electrical Engineering, Duke University, Durham, NC, 1987.

[17] S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal distributed decision fusion," *IEEE Transactions on Aerospace Electrical Systems*, vol. 25, pp. 761–765, Sept. 1989.

[18] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley, Reading, MA, 1991.

[19] N. Gnanapandithan and B. Natarajan, "Impact of local decision rules in distributed sensor networks," in *Proceedings of IEEE Vehicular Technology Conference*, ser. Submitted for Publication, May 2006.

[20] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann arbor University of Michigan Press, Ann Arbor, MI, 1975.

Nithya Gnanapandithan received her MS degree in Electrical Engineering from Kansas State University in Fall 2005 and her bachelor's degree in Electronics and Communication Engineering from Madras University, India in June 2003.

Nithya joined the WiCom group at K-State in August 2004 and she has been working in the distributed sensor networks/MIMO systems area since then. Her work involves the use of genetic algorithms to optimize distributed sensor networks. She is currently working as a design engineer at RELM Wireless Corporation, a company specializing in two-way radios.

Balasubramaniam Natarajan received his B.E and Ph.D. degrees, both in Electrical Engineering from Birla Institute of Technology and Science, Pilani, India and Colorado State University, Fort Collins, CO, in 1997 and 2002 respectively.

Since Fall 2002, he has been a faculty member in the Department of Electrical and Computer Engineering at Kansas State University, where he is currently the director of the WiCom research group. His research interests include spread spectrum communications, multi-carrier CDMA and OFDM, multi-user detection, statistical signal processing, distributed signal processing in sensor networks, antenna array processing and application of evolutionary algorithms in wireless communications. He has over 40 publications in peer-reviewed journals and conferences and has a pending patent in adaptive sequence design for spread spectrum systems.