The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)

PSO FOR CONSTRAINED OPTIMIZATION: OPTIMAL POWER SCHEDULING FOR CORRELATED DATA FUSION IN WIRELESS SENSOR NETWORKS

Thakshila Wimalajeewa
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM, USA

Sudharman K. Jayaweera
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM, USA

ABSTRACT

We consider the problem of optimal power scheduling for decentralized detection of a deterministic signal in a wireless sensor network with correlated observations. Each distributed sensor node independently performs amplify-and-forward (AF) processing of its observation. The fading coefficients of wireless links from distributed sensors to the fusion center (FC) are assumed to be available at transmitting nodes. When sensor observations are correlated it is difficult to derive a closed form solution for optimal power values to achieve a required fusion error performance. In this work, we develop an evolutionary computation technique based on Particle Swarm Optimization (PSO) to obtain the optimal power allocation under a required fusion error probability threshold constraint. It is shown that the optimal power allocation scheme turns off the nodes with poor channels and provides significant system power savings compared to that of uniform power allocation scheme especially when either the number of sensors in the system is large or the local observation quality is good.

I INTRODUCTION

Consider a Wireless Sensor Network (WSN) consisting of a fusion center (FC) and a large number of spatially separated sensors. The distributed sensor nodes collect observations, perform amplify-and-forward (AF) processing and transmit them to the FC. The node observations are assumed to be correlated. The wireless channel between sensor nodes and the FC is assumed to undergo fading. In this paper we consider the problem of optimal power allocation for fusion of a deterministic signal in such a sensor network with correlated observations to keep the fusion error probability under a required threshold.

We first derive the fusion error probability at the fusion center and then use Particle Swarm Optimization (PSO), which is an evolutionary computation technique based on the movement and intelligence of particles of a swarm, to numerically find the optimal power allocation scheme to keep the fusion error probability under a required threshold. Note that, he optimal power allocation scheme when observations are i.i.d. was previously derived in [1]. We show that the optimal power allocation scheme has considerably better performance over the uniform power allocation scheme specifically when the number of nodes in the network is large or the local SNR is high. It is also verified that the results obtained via PSO-based numerical method closely match with analytical results under the same network conditions.

The remainder of this paper is organized as follows: In Section II, the fusion problem is formulated and the optimal fusion performance is derived. Optimal power allocation scheme based on PSO is developed in Section III. Section IV presents the performance results and, finally, Section V gives concluding remarks.

II FUSION PROBLEM FORMULATION

We consider a binary hypothesis testing problem in an n-node distributed wireless sensor network. The k-th sensor observation under each hypothesis is given by,

$$\begin{aligned} H_0 : z_k &= v_k;\ k = 1, 2, ..., n \\ H_1 : z_k &= x_k + v_k;\ k = 1, 2, ..., n\ , \end{aligned} \quad (1)$$

where v_k is the zero-mean observation noise with variance σ_v^2 and x_k is the signal to be detected. In vector notation, (1) becomes $\mathbf{z} = \mathbf{x} + \mathbf{v}$ where $\mathbf{v}$ is a zero mean Gaussian n- vector of noise samples with covariance matrix $\mathbf{\Sigma_v}$. In general $\mathbf{\Sigma_v}$ is not a diagonal matrix unless the observation noise is i.i.d..

We consider the detection of a deterministic signal so that $x_k = m$ for all k. Let us define the local signal-to-noise ratio as $\gamma_0 = \frac{m^2}{\sigma_v^2}$. The prior probabilities of the two hypotheses H_1 and H_0 are denoted by $P(H_1) = \pi_1$ and $P(H_0) = \pi_0$, respectively.

Each node processes its own observation to produce a local decision $u_k(z_k)$ and sends it to the fusion sensor. In this paper we assume AF local processing, according to which each node retransmits an amplified version of its own observation to the fusion center. Note that this class of processing has been shown to perform well when the observations at the sensor nodes are corrupted by additive noise [2]. Hence the local decisions sent to the fusion center are, $u_k = g_k z_k;\ k = 1, 2, ...n$, where g_k is the amplifier gain at node k. The received signal r_k at the fusion center under each hypothesis is given by

$$\begin{aligned} H_0\ : r_k &= n_k;\ \ k = 1, 2, ..., n \\ H_1\ : r_k &= h_k g_k x_k + n_k;\ \ k = 1, 2, ..., n \end{aligned}$$

where $n_k = h_k g_k v_k + w_k$, h_k is the channel fading coefficient and w_k is the receiver noise that is assumed to be independent and identically distributed with mean zero and variance σ_w^2. It can be shown that the probability of error at the fusion center for a Bayesian optimal detector is given by

$$P(E) = Q\left(\frac{1}{2}\sqrt{m^2\mathbf{e^T A \Sigma_n^{-1} A e}}\right) . \quad (2)$$

where Q-function is defined by $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{\zeta^2}{2}} d\zeta$, $\mathbf{A} = \text{diag}(h_1 g_1, h_2 g_2, ..., h_n g_n)$, $\mathbf{\Sigma_n} = \mathbf{A\Sigma_v A} + \sigma_w^2\mathbf{I}$, $\mathbf{I}$ is the $n \times n$ identity matrix and $\mathbf{e}$ is the n-length vector of all ones. Note that, the prior probabilities are assumed to be equal in (2).

III PSO FOR OPTIMAL POWER ALLOCATION

The objective is to allocate transmit power to distributed nodes in order to maintain a required fusion error probability at the fusion center with minimum total power consumption by the whole network. Thus the optimal power allocation problem can be formulated as,

$$\begin{aligned} &\min_{g_i \geq 0} \sum_{i=1}^{n} g_i^2 \text{ such that} \\ &P(E) = Q\left(\frac{1}{2}\sqrt{m^2 \mathbf{e^T A \Sigma_n^{-1} A e}}\right) \leq \epsilon \\ \text{and} \quad &g_i \geq 0; \ \ i = 1, 2, \cdots, n \end{aligned} \tag{3}$$

where ϵ is the required fusion error probability threshold. Since it is not possible to find a closed form solution for g_i's in (3), (it is difficult to evaluate $\mathbf{\Sigma_n^{-1}}$ in closed form when $\mathbf{\Sigma_v}$ is not diagonal) in the following we propose a numerical method based on PSO to find the optimal power when local observations are correlated. Since the PSO is not directly applicable for constrained optimization problems, we first transform our constrained optimization problem in (3) into an unconstrained optimization problem using the exterior penalty function approach [3].

A *Penalty function approach for constrained optimization*

In the penalty function method, the constrained problem is transformed into a sequence of unconstrained minimization problems such that the constrained minimum can be obtained by solving the sequence of unconstrained minimization problems. There are basically two types of penalty function methods: the interior penalty function method and the exterior penalty function method [3]. In the interior penalty function method the initial solution should be in the feasible region, which may not be easily verifiable; whereas in exterior penalty function method the initial solution does not necessarily have to be in the feasible region. For this reason, the exterior penalty function method is widely used in evolutionary constrained optimization problems.

Let us re-write the the optimization problem (3) as,

$$\min \ \ f(\mathbf{g}) \ \text{ such that } \ h_j(\mathbf{g}) \leq 0; \ \ j = 1, \cdots, m. \tag{4}$$

where $f(\mathbf{g}) = \sum_{i=1}^{n} g_i^2$, $h_1(\mathbf{g}) = \beta^2 - \mathbf{e^T A \Sigma_n^{-1} A e}$ where $\beta = \frac{2}{m} Q^{-1}(\epsilon)$, $h_{i+1}(\mathbf{g}) = g_i$ for $i = 1, 2, \cdots, n$ and $\mathbf{g} = [g_1, \cdots, g_n]^T$. Then the exterior penalty function for the above minimization problem can be formulated as [3],

$$\begin{aligned} \phi(\mathbf{g}, r_k) = f(\mathbf{g}) + r_k\{&(\max[h_1(\mathbf{g}), 0])^q \\ &+ \sum_{j=2}^{m} (\max[h_j(\mathbf{g}), 0])^q\} \end{aligned} \tag{5}$$

where r_k is a positive penalty parameter and q is a non-negative constant. Usually, the value of q is chosen to be 2 in practice [3]. The exterior penalty function algorithm to find the optimal solution for the problem (4) can be stated as below. (Note that subscript of $\mathbf{g}$ denotes the index corresponding to a penalty parameter while the superscript of $\mathbf{g}$ denotes the iteration number of the minimization algorithm for a particular penalty parameter.)

step 1: Start from any initial solution $\mathbf{g}_1^1$ and a suitable value of r_1. Set $k = 1$.

step 2: Find the vector $\mathbf{g}_k^*$ that minimizes the function given in (5).

step 3: Test whether the point $\mathbf{g}_k^*$ satisfies all the constraints. If $\mathbf{g}_k^*$ is feasible, it is the desired optimum and hence terminate the procedure. Otherwise go to next step.

step 4: Choose the next value of the penalty parameter according to the relation $\frac{r_{k+1}}{r_k} = c$ where c is a constant greater than one and set $\mathbf{g}_{k+1}^1 = \mathbf{g}_k^*$ and $k = k + 1$. Go to step 2.

When $f(\mathbf{g})$ and $h_j(\mathbf{g})$, $j = 1, 2, \cdots, m$ are continuous, as in this case, the unconstrained minima $\mathbf{g}_k^*$ of $\phi(\mathbf{g}, r_k)$ converge to the optimal solution of the original problem $f(\mathbf{g})$ as $k \to \infty$ and $r_k \to \infty$, as long as an optimal solution exits for (4) [3]. When the observations are i.i.d, it can be shown that $\phi(\mathbf{g}, r_k)$ is a strictly convex function for $g_i \geq \frac{\sigma_w^2}{3h_i^2\sigma_v^2}$ for $i = 1, 2, \cdots, n$. Further, it can be seen that when h_i's are small enough the convexity of $\phi(\mathbf{g}, r_k)$ holds for $g_i \geq 0$, ensuring a global minimum for $\phi(\mathbf{g}, r_k)$. Thus, we may expect that $\phi(\mathbf{g}, r_k)$ has a global minimum for each r_k even when the observation noise is correlated under above conditions at least for sufficiently small correlations.

B *Particle swarm optimization*

To evaluate optimal $\mathbf{g}_k^*$ for each penalty parameter r_k as required in the *step 2* above, we use the particle swarm optimization technique [4–6]. Note that PSO is a stochastic evolutionary computation technique based on the movement and intelligence of swarms, that has been shown to outerperform other optimization methods such as genetic algorithms [6] in certain applications. A brief overview of the particle swarm terminology is given in Table 1. (For more details see [5]).

Each particle in the swarm is given an initial random location in the n-dimensional solution space of the problem being optimized. The solution space consists of a reasonable range for the parameter set in which to search for the optimal solution. This essentially specifies the minimum and maximum values that parameters can take in each dimension. In this solution space each particle acts individually and accelerates toward the best personal location (*pbest)* while checking the *fitness value* of its current position (fitness value of a position is obtained by evaluating the so-called *fitness function* at that location). If a particles's current location has a better fitness value than that of its current *pbest*, then the *pbest* is replaced by the current location. Each particle in the swarm has knowledge of the location with best fitness value of the entire swarm which is called the global best or *gbest*. At each point along their path, each particle also compares the fitness value of their *pbest* to that of *gbest*. If any particle has a *pbest* with better fitness value than that of current *gbest*, then the current *gbest* is replaced by that particle's *pbest*. The movement of particles is stopped once all particles reach sufficiently close to the position with best fitness value of the swarm.

Table 1: PSO Terminology

Particle/Agent	A single individual in the swarm
Location/Position	An agent's n-dimensional coordinates which represent a solution to the problem
Swarm	The entire collection of agents
Fitness	A single number representing the goodness of a given solution
$\mathbf{pbest}$	The location in parameter space of the best fitness returned for a specific agent
$\mathbf{gbest}$	The location in parameter space of the best fitness returned in the entire swarm
$\mathbf{V_{max}}$	The maximum allowed velocity in a given direction

In the following we give the algorithmic steps needed to implement the PSO for a given problem.

(I). Define the solution space and fitness function:

Pick the parameters that need to be optimized and give them a reasonable range in which to search for the optimal solution which is called the solution space. For the optimization problem (4), the parameter set to be optimized is $[g_1, \cdots, g_n]$, and the corresponding solution space for each g_i is $[0, \infty)$ for $i = 1, \cdots, n$. The fitness function should exhibit a functional dependance that is relative to the importance of each characteristic being optimized. For the optimization problem in (4) we select the penalty function (5) as the fitness function.

We denote the swarm size by M. For each k in (5), we perform PSO algorithm to find $\mathbf{g}_k^*$. For each k, let us define, $\mathbf{g}_{k,m}$ as the position vector of the m- th particle; $\mathbf{P}_{k,m}$ as the *pbest* of the m th particle; $\mathbf{P}_{k,gbest}$ as the *gbest* of the swarm; $\phi(\mathbf{g}_{k,m}, r_k)$ as the fitness value corresponding to the location $\mathbf{g}_{k,m}$ of the m-th particle; $\phi(\mathbf{P}_{k,m}, r_k)$ as the fitness value corresponding to the *pbest* $\mathbf{P}_{k,m}$ of the m-th particle; $\phi(\mathbf{P}_{k,gbest}, r_k)$ as the fitness value corresponding to the *gbest* of the swarm and $\mathbf{V}_{k,m}$ as velocity of the m-th particle. The maximum number of iterations of PSO for each k is set to S.

(II). Initialize swarm locations and velocities for each particle:

Initializing position: For $k = 1$, (i.e. for the penalty parameter r_1) and for each particle m, $m = 1, \cdots, M$, $\mathbf{g}_{k,m}^1$ is chosen randomly within the solution space. If $k > 1$, $\mathbf{g}_{k,m}^1$=$\mathbf{P}_{k-1,m}^S$ where $\mathbf{P}_{k-1,m}^S$ is the *pbest* for the m-th particle for $k = k - 1$ at the S-th iteration of PSO.

Initializing pbest: Since its initial position is the only location encountered by each particle at the run's start, this position becomes each particle's initial *pbest*. i.e. $\mathbf{P}_{k,m}^1 = \mathbf{g}_{k,m}^1$.

Initializing gbest: The first *gbest* is selected as the initial *pbest* which gives the best fitness value: i.e. $\mathbf{P}_{k,gbest}^1$=$\mathbf{P}_{k,m_1}^1$ where $m_1 = \arg\min_{1 \le m \le M}\{\phi(\mathbf{P}_{k,m}^1, r_k)\}$.

Initializing velocities: Initialize $\mathbf{V}_{k,m}^1$ as zeros for each particle m.

(III). Fly the particles through the solution space:

Each particle is then moved through the solution space by performing following steps on each particle individually.

(a). Evaluate the particle's fitness value and compare it with those of *pbest* and *gbest*.

In the s-th iteration of the PSO, for each particle m, if $\phi(\mathbf{g}_{k,m}^s, r_k) < \phi(\mathbf{P}_{k,m}^s, r_k)$ then set $\mathbf{P}_{k,m}^s = \mathbf{g}_{k,m}^s$. Set $\mathbf{P}_{k,gbest}^s = \mathbf{P}_{k,m_s}^s$ where $m_s = \arg\min_{1 \le m \le M}\{\phi(\mathbf{P}_{k,m}^s, r_k)\}$.

(b). Update the particle's velocity: The velocity of the particle is changed according to the relative locations of *pbest* and *gbest*. The particles are accelerated in the directions of the locations of best fitness value according to the following equation [5, 7]:

$$\mathbf{V}_{k,m}^{s+1} = \mathcal{X}(w\mathbf{V}_{k,m}^s + c_1 rand()(\mathbf{P}_{k,m}^s - \mathbf{g}_{k,m}^s) + c_2 rand()(\mathbf{P}_{k,gbest}^s - \mathbf{g}_{k,m}^s)), \quad (6)$$

where $\mathcal{X}$ is the constriction factor that controls and constricts velocities; w is the inertia weight that determines to what extent the particle remains along its original course unaffected by the pull of *pbest* and *gbest*, c_1 and c_2 are positive constants that determine the relative "pull" of *pbest* and *gbest* (In fact c_1 determines how much the particle is influenced by the memory of its best location and c_2 determines how much the particle is influenced by the rest of the swarm) and rand() is the random number generator that returns a number between 0 and 1.

(c). Move the particle: Once the velocity has been determined as in (6), move the particle to its next location as $\mathbf{g}_{k,m}^{s+1} = \mathbf{g}_{k,m}^s + \Delta t \mathbf{V}_{k,m}^{s+1}$. The velocity is applied for a given time step Δt.

(IV). Repetition:

After the velocity and the position are updated the process is repeated starting at step (III) until the termination criteria are met. The termination criteria can be a user-defined maximum iteration number or a target fitness termination condition. In the latter case, the PSO is run for the user-defined number of iterations, but at any time if a solution is found that is greater than or equal to the target fitness value, then PSO is stopped at that point. In our work we set the maximum iteration number (S) for PSO as defined before. Once the termination criteria are met, the optimal solution $\mathbf{g}_k^*$ for the unconstrained minimization problem in (5) for given k is $\mathbf{P}_{k,gbest}^S$.

C Selection of parameter values for constrained PSO

We chose the population size to be $M = 30$ as this has been shown to be sufficient for many engineering problems [8]. Various values for inertia weight w have been suggested in the literature. Since larger weights tend to encourage global exploration and, conversely smaller initial weights encourage local exploitations, Eberhart and Shi have suggested to vary w linearly from 0.9 to 0.4 over the course of the run [9]. We set both c_1 and c_2 to 2.0 [5,7], and the constriction factor $\mathcal{X}$ to 0.73 [7].

When the particle hits the boundary of the solution space, there are basically 3 ways to take this into account; Absorbing walls, reflecting walls and invisible walls [5]. It was shown in [5] that "invisible walls" technique has better performance over the other two in many optimization problems. In our problem, we used the invisible walls technique which allows the particles to fly without any restriction when they hit the boundary of the solution space. However, particles that roam outside the allowed solution space are not evaluated for fitness until they returned back to the solution space.

IV NUMERICAL RESULTS

In this section we illustrate performance gains possible with the derived optimal power allocation scheme via PSO-based constrained optimization. We assume that fading coefficients h_k's of the channel between sensors and the fusion center to be Rayleigh distributed with a unit mean. Note that, without loss of generality, the fading coefficients h_k's have been arranged in the descending order (i.e. $h_1 \geq h_2 \geq \cdots \geq h_n$) in obtaining Tables 2 and 3. Although the above PSO-based numerical method is applicable for any noise covariance matrix $\mathbf{\Sigma_v}$, in the simulations we assume a 1-D sensor network in which adjacent nodes are separated by distance d and correlation between nodes i and j is proportional to $\rho_0^{d|i-j|}$ where $|\rho_0| < 1$. Letting $\rho_0^d = \rho$, the noise covariance matrix $\mathbf{\Sigma_v}$ can be written as a Hermitian Toeplitz matrix with the first row, $\sigma_v^2(1, \rho, \cdots, \rho^{(n-1)})$.

A Convergence of constrained PSO based on exterior penalty function approach

When the observations are i.i.d., the solution for the optimal power allocation problem (3) can be found analytically [1]. It was sown in [1] that the optimal solution turns off the sensors with poor observation quality and bad channels so that they do not need to send the observations to the fusion center. Here, we observe that the results obtained via PSO-based method closely match with that of the analytical solution when the observations are i.i.d. The convergence of the optimal penalty function value $\phi(\mathbf{g}^*, r_k)$ to the optimal objective function $f(\mathbf{g}^*)$ as penalty parameter r_k is varied, is shown in table 2. The results in Table 2 corresponds to $n = 10$, $\gamma_0 = 10dB$, $\rho = 0$ and $\epsilon = 0.1$. As can be observed from Table 2, after seven iterations of r_k, the penalty function is very close to the objective function. The comparison of $\mathbf{g}^*$ obtained numerically as in Table 2 and analytically under the same network conditions are shown in first two rows of the Table 3 for 10 nodes. (analytical results are obtained from [1]). It can be seen that the numerical results closely match with that of analytical results after a few iterations over the penalty parameter. The third row of the Table 3 shows the optimal $\mathbf{g}^*$ obtained numerically when $\rho = 0.1$, $n = 10$, $\gamma_0 = 10dB$ and $\epsilon = 0.01$. It shows that, when the observations are correlated the optimal solution for (3) turns off the sensors with poor channels similar to that in i.i.d observation case. But it is also seen that the sensors need to spend slightly more power when the observations are correlated (under the same n, γ_0 and ϵ).

Table 2: Numerical results: $n = 10$, $\gamma_0 = 10dB$, $\rho = 0$ and $\epsilon = 0.01$

Value of r_k	$\phi(\mathbf{g}^*, r_k)$	$f(\mathbf{g}^*)$	error
1	156.1372	6.5386	149.5986
20	15.0887	15.0784	0.0103
40	15.0934	15.0864	0.0069
60	15.0955	15.091	0.0045
80	15.0965	15.093	0.0035
100	15.0972	15.0945	0.0028
120	15.0977	15.0954	0.0023

Table 3: Comparison of Analytical and Numerical Results

$\mathbf{g}^*$: Analytical ($\rho = 0$)	[1.6172, 1.5888, 1.5555, 1.4666, 1.4616, 1.4107, 1.1231, 0, 0, 0]
$\mathbf{g}^*$: Numerical ($\rho = 0$)	[1.6163, 1.5696, 1.5548, 1.5014, 1.4501, 1.4099, 1.1212, 0.0013, 0.0066, 0.0008]
$\mathbf{g}^*$: Numerical ($\rho = 0.1$)	[1.6717, 1.5867, 1.6112, 1.5034, 1.5285, 1.4758, 1.3381, 0.3366, 0.0062, 0.0005]

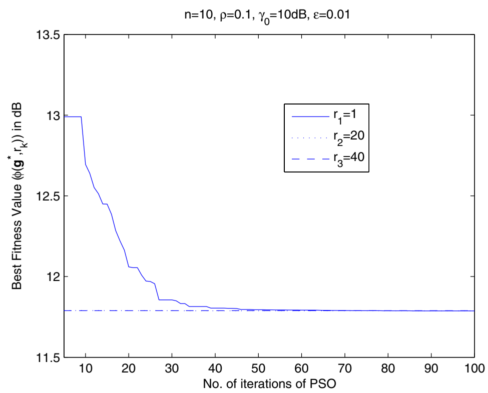
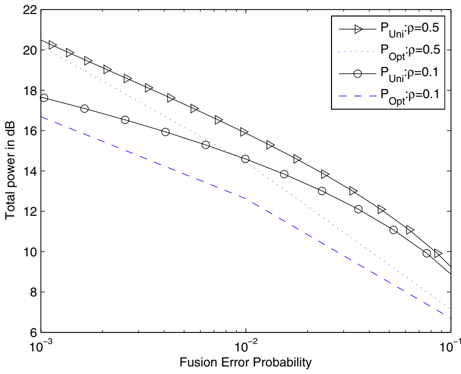
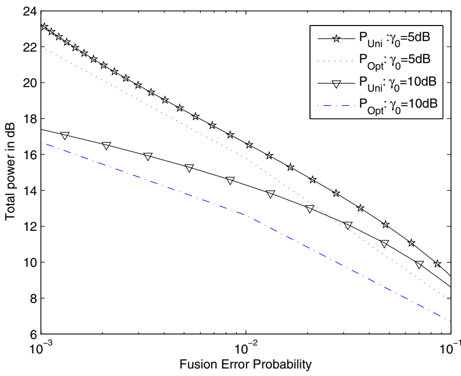


Figure 1: Best fitness returned for PSO iterations for a given penalty parameter: P(e)=0.1

Note that we deployed the PSO algorithm for each penalty parameter r_k of the unconstrained optimization problem (5) to find optimal $\phi(\mathbf{g}^*, r_k)$. Figure. 1 shows how fast the best fitness value of PSO converges for a given penalty parameter r_k; The penalty parameter r_k was set to $1, 20, 40, \cdots$ for $k = 1, 2, 3, \cdots$. As can be seen, when the penalty parameter is r_1, the penalty function $(\phi(\mathbf{g}, r_1))$ converges to its optimal value within less than 50 iterations of PSO. For the rest of the penalty parameters r_k where $k > 1$, the penalty function $(\phi(\mathbf{g}, r_k))$ converges to its optimal value in 1-2 iterations of PSO.



(a)



(b)

Figure 2: PSO: Total Power Vs. Fusion Error Probability when observation noise is correlated. (a). $n = 20, \gamma_0 = 10dB$ (b). $n = 20, \rho = 0.1$

B Optimal power allocation via constrained PSO

In the following we show the performance of the optimal power allocation scheme, obtained through numerical method based on constrained-PSO, over the uniform power allocation scheme. The performance measure is the total network power expenditure defined as $P_{total} = \sum_{i=1}^{n} g_i^2$. The dependance of the total power on the required fusion error probability is shown in Fig. 2, parameterized by n, ρ and γ_0.

Figure 2a shows that the network needs more power when the correlation coefficient of the observation noise is high since then the new information added by each additional sensor decreases degrading the fusion performance. On the other hand, when local SNR is high it is enough to turn on only the nodes having channels with high fading coefficients so that the total power spent by the network decreases (Fig. 2b). It was observed that when the number of sensors in the network is large the required total power with optimal power allocation is less than that with uniform power allocation, since all sensors do not need to operate at their maximum power level when there is a large number of nodes.

V CONCLUSION

In this paper we developed a numerical method for obtaining the optimal power scheduling scheme for data fusion in a wireless sensor network when the observations are correlated. The proposed method is an evolutionary computation technique based on extending the PSO to constrained optimization problems. We saw that the constrained PSO via exterior penalty function approach has a fast convergence to the required optimal solution. From numerical results we observed that the optimal power allocation scheme provides significant total energy savings over that of the uniform power allocation scheme specifically when number of nodes in the system is large, or the local observation quality is good.

REFERENCES

[1] T. Wimalajeewa and S. K. Jayaweera, "Optimal Power Scheduling for Data Fusion in Inhomogeneous Wireless Sensor Networks", *Proc. IEEE Int. conf. on Advanced Video and Signal based Surveillance (AVSS'06)*, Nov. 2006, Sydney, Australia

[2] J. F. Chamberland and V. V. Veeravalli, "Decentralized Detection in wireless sensor Systems with Dependent Observations", *Proc. Int. Conf. Comput., Commun. and Contr. Technol. (CCCT)*, Aug. 2004, Austin,TX

[3] S. S. Rao, *Optimization: Theory and Applications*. Wiley Eastern Ltd, New Delhi, India, 1995

[4] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", *Proc. IEEE Conf. Neural Networks IV*, 1995, Piscataway, NJ

[5] J. Robinson and Y. R. Samii, "Particle Swarm Optimization in Electromagnetism", *IEEE Trans. Antennas Propagat.*, IOS press, 2002, pp. 397–407 vol. 52, no. 2, Feb. 2004.

[6] J. Kennedy and W. M. Spears, "Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on multi modal problem generator", *Proc. IEEE Int. Conf. Evolutionary Computation*, 1998

[7] K. E. Parsopoulos and M. N. Vrahatis, "Particle Swarm Optimization Method for Constrained Optimization Problems", *in Intelligent Technologies: New Trends in Intellegent Technologies*, IOS press, 2002, pp. 214–220

[8] A. Carlisle and G. Doizier, "An off-the-shelf PSO", *Proc. 2001 Workshop on Particle Swarm Optimization*, 2001, Indianapolis, IN

[9] R. C. Eberhart and Y. Shi, "Evolving Artificial Neural Networks", *Proc. 1998 Int. Conf. Neural Networks and Brain*, 1998, Beijing, P.R.C