Meta-Heuristics for Symbol Detection in a Spatial Multiplexing System

Adnan Ahmed Khan*, Muhammad Naeem, Syed Ismail Shah*****
*Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan
adnankhan@mcs.edu.pk
**Simnon Fraser University Burnaby, BC Canada
***Iqra University, Islamabad, Pakistan

Abstract

An application of Swarm Intelligence (SI) based Meta-heuristics for a NP-hard problem in the area of wireless communications is explored. The specific problem is of detecting symbols in a Multi-Input Multi-Output (MIMO) communications system. This approach is particularly attractive as SI is well suited for physically realizable, real-time applications, where low complexity and fast convergence is of absolute importance. Application of Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) algorithms is discussed. While an optimal Maximum Likelihood (ML) detection using an exhaustive search method is prohibitively complex, we show that the Swarm Intelligence optimized MIMO detection algorithms gives near-optimal Bit Error Rate (BER) performance in fewer iterations, thereby reducing the ML computational complexity significantly. The simulation results suggest that the proposed detector gives an acceptable performance complexity trade-off in comparison with optimal ML and non-linear Vertical Bell labs Layered Space Time (VBLAST) detectors. The proposed techniques result in as high as 14-dB enhanced BER performance with acceptable increase in computational complexity in comparison with VBLAST. The reported algorithms reduce the computer time requirement significantly over exhaustive search method with a reasonable BER performance.

Keywords: Spatial Multiplexing, Swarm Intelligence, PSO, ACO, Symbol Detection, Multi-Input Multi-Output System (MIMO).

Introduction

Real life optimization problems are often so complex that finding the best solution becomes computationally infeasible. Therefore, an intelligent approach is to search for a good approximate solution consuming lesser computational resources. Several engineering problems contain multiple objectives that need to be addressed simultaneously. Many techniques have been proposed that imitate nature's own ingenious ways to explore optimal solutions for both single and multi-objective optimization problems. Earliest of the nature inspired techniques are genetic and other evolutionary heuristics that evoke Darwinian evolution principles.

Computational Swarm Intelligence [1] is one such innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking phenomena in vertebrates and the cooperative forging strategy of real ants.

PSO meta-heuristic is a population-based Swarm Intelligence (SI) technique inspired by the coordinated movements of birds flocking introduced by Kennedy and Eberhart in 1995 [2],[3]. Standard PSO uses a real-valued multidimensional solution space [2], whereas in binary PSO particle positions are discrete rather than real valued [4].

ACO meta-heuristic is another SI algorithm based on the cooperative forging strategy of real ants [5],[6]. In this approach, several artificial ants perform a sequence of operations iteratively. Ants are guided by a greedy heuristic algorithm which is problem dependent, that aid their search for better solutions iteratively. Ants seek solutions using information gathered previously to perform their search in the vicinity of good solutions. Its binary version known as binary ant system (BAS) is well suited for constrained optimization problems with binary solution structure [7], [8].

Simple mathematical model, resistance to being trapped in local minima and convergence to near optimal solution in fewer iterations make these techniques a suitable candidate for real-time NP-hard communication problems [20],[21], in addition to other wide range of applications like traveling salesman problem [3], [6].

The relevant information–theoretic analysis reveals that significant performance gains are achievable in wireless communication systems using a MIMO architecture employing multiple antennas [9]. This architecture is suitable for higher data rate multimedia communications [10]. One of the challenges in building wide band MIMO systems is the tremendous processing power required at the receiver side. While coded MIMO schemes offer better performance than separate channel coding and modulation scheme by fully exploring the tradeoff between multiplexing and diversity [11], its hardware complexity can be significant, especially for wide band system with more than four antennas both at the transmitter and the receiver sides. On the other hand, it is easier to implement traditional channel coding schemes such as Convolution code and Turbo code for data rates of hundreds of Mbps. For this

Corresponding Author: Adnan Ahmed Khan
NUST Publishing, © (2008), ISSN: 2070-9900

reason we discuss uncoded MIMO system also called spatial multiplexing as shown in Fig 1.

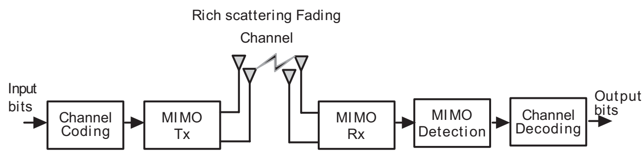

Fig. 1. Spatial multiplexing system

One of the challenges in designing a MIMO system is tremendous processing requirements at the receiver. MIMO symbol detection involves detecting symbol from a complex signal at the receiver. This detection process is considerably complex as compared to single antenna system. Several MIMO detection techniques have been proposed [12]. These detection techniques can be broadly divided into linear and non-linear detection methods. Linear methods offer low complexity with degraded BER performance as compared to non-linear methods. This paper focuses on non-linear detectors and makes an effort to improve BER performance at the cost of complexity and vice versa. ML and V-BLAST detectors [13],[14] are well known non-linear MIMO detection methods. ML outperforms VBLAST in BER performance, while VBLAST is lesser complex than ML. In [15],[16] a performance complexity trade off between the two methods have been reported.

Being NP-hard [12] computational complexity of optimum ML technique is generically exponential. Therefore, in order to solve these problems for any non-trivial problem size, exact, approximate or un-conventional techniques such as meta-heuristics based optimization approach can be used. The exact method exploits the structure of the lattice and generally obtains the solution faster than a straightforward exhaustive search [12]. Approximation algorithm provides approximate but easy to implement low-complexity solutions to the integer least-squares problem. Whereas, meta-heuristics based algorithm works reasonably well on many cases, but there is no proof that it always converges fast like SI techniques.

An application of SI based meta-heuristics for symbol detection problem in wireless communication is presented in [21], [22]. The problem is to detect symbols from a composite signal, received at multiple receivers, transmitted from multiple transmitters. This MIMO detection problem is one of the most important issues faced in wireless communications area. A performance analysis of binary particle swarm and binary ant system based symbol detection algorithm for spatial multiplexing systems is reported in this paper. An acceptable and interesting performance complexity trade off is observed from the reported results.

The rest of the paper is organized as follows. Section 2 provides the system model. In section 3 MIMO symbol detection problem for flat fading channel is described. A brief overview of the existing MIMO detectors is given in section 4. Section 5 provides the details of the proposed SI based MIMO detection techniques. Performance of the proposed detector is reported in section 6, while section 7 concludes the paper.

MIMO detection for flat-fading channel

MIMO channel model

Consider a MIMO system where N_t different signals are transmitted and arrive at an array of N_r $(N_t \leq N_r)$ receivers via a rich-scattering flat-fading environment. Grouping all the transmitted and received signals into vectors, the system can be viewed as transmitting an N_t x 1 vector signal **x** through an N_t x N_r matrix channel **H,** with N_r x 1 Gaussian noise vector **v** added at the input of the receiver. The received signal as an N_r x 1 vector can be written as:

$$\mathbf{y} = \mathbf{Hx} + \mathbf{v} \tag{1}$$

Where **y** is the received The $(n_r, n_t)^{th}$ element of **H,** $h_{n_r n_t}$, is the complex channel response from the n_t^{th} transmit antenna to the n_r^{th} receive antenna. The transmitted symbol **x** is zero mean and has covariance matrix $\mathbf{R_x} = \mathrm{E}\{\mathbf{xx^*}\} = \sigma_x^2\mathbf{I}$. The vector **v** is also zero-mean and $\mathbf{R_v} = \mathrm{E}\{\mathbf{vv^*}\} = \sigma_v^2\mathbf{I}$. The entries of the channel matrix **H** are assumed to be known to the receiver but not to the transmitter. This assumption is reasonable if training or pilot signals are sent to estimate the channel, which is constant for some coherent interval.

Problem formulation

The task is that of detecting N_t transmitted symbols from a set of N_r observed symbols that have passed a non-ideal communication channel, typically modeled as a linear system followed by an additive noise vector as shown in Fig 2:

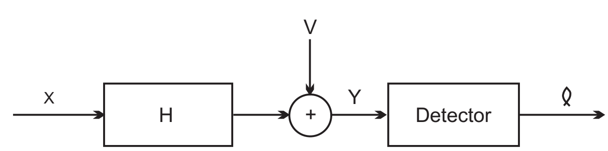

Fig. 2. A simplified linear MIMO communication system showing the following discrete signals: transmitted symbol vector $\boldsymbol{x} \in \chi^{N_t}$, channel matrix $\mathbf{H} \in \boldsymbol{R}^{N_t x N_r}$, additive noise vector $\mathbf{v} \in \boldsymbol{R}^{N_t}$, receive vector $\mathbf{y} \in \boldsymbol{R}^{N_t}$, and detected symbol vector $\hat{\boldsymbol{x}} \in \boldsymbol{R}^{N_r}$.

Transmitted symbols from a known finite alphabet $\boldsymbol{\chi} = \{x_1,\ldots,x_M\}$ of size M are passed to the channel. The detector chooses one of the M^{Nt} possible transmitted symbol vectors from the available data. Assuming that the symbol vectors $\boldsymbol{x} \in \chi^{N_t}$ are equiprobable, the *Maximum Likelihood (ML)* detector always returns an optimal solution according to the following:

$$\boldsymbol{x}_* = \underset{x \in \chi^{N_t}}{\arg\max}\, P(\boldsymbol{y}\, is\, observed \,|\, \boldsymbol{x}\, was\, sent) \quad (2)$$

Assuming the additive noise **v** to be white and Gaussian, the ML detection problem of Figure 1 can be can be expressed as the minimization of the squared Euclidean distance to a target vector **y** over N_t-dimensional finite discrete search set:

$$\boldsymbol{x}_* = \underset{x \in \chi^{N_t}}{\arg\min} \left\| \boldsymbol{y} - H\boldsymbol{x} \right\|^2 \quad (3)$$

Optimal ML detection scheme needs to examine all M^{Nt} or 2^{bNt} symbol combinations (*b* is the number of bits per symbol). The problem can be solved by enumerating over all possible **x** and finding the one that causes the minimum value as in (3). Therefore, the computational complexity increases exponentially with constellation size *M* and number of transmitters N_t.

We present Swarm Intelligence algorithms assisted spatial multiplexing system symbol detectors that view the MIMO symbol detection issue as a combinatorial optimization problem and try to approximate the near optimal solution iteratively.

Some existing MIMO detectors

Linear MIMO detectors

A straightforward approach to recover x from y is to use an N_t x N_r weight matrix **W** to linearly combine the elements of **y** to estimate **x**, i.e. $\hat{x} = \mathbf{W}\mathbf{y}$.

Zero-Forcing(ZF)

The ZF algorithm attempts to null out the interference introduced from the matrix channel by directly inverting the channel with the weight matrix [12].

Minimum Mean Squared Error (MMSE)

A drawback of ZF is that nulling out the interference without considering the noise can boost up the noise power significantly, which in turn results in performance degradation. To solve this, MMSE minimizes the mean squared-error, i.e. $J(\mathbf{W}) = \mathrm{E}\{(x-\hat{x})^*(x-\hat{x})\}$, with respect to **W** [17], [18].

Non-Linear MIMO Detectors

VBLAST

A popular nonlinear combining approach is the vertical Bell labs layered space time algorithm (VBLAST)[13]. This detection method is also called Ordered Successive Interference Cancellation (OSIC). It uses the detect-and-cancel strategy similar to that of decision-feedback equalizer. Either ZF or MMSE can be used for detecting the strongest signal component used for interference cancellation. The performance of this procedure is generally better than ZF and MMSE. VBLAST provides a suboptimal solution with lower computational complexity than ML. However, the performance of VBLAST is degraded due to error propagation.

ML Detector

Maximum Likelihood detector is optimal but computationally very expansive. ML detection is not practical in large MIMO systems.

Pso for spatial multiplexing system

Particle Swarm Optimization (PSO)

Particle Swarm Optimization argues that intelligent cognition derived from interactions of individuals in a social world and this socio-cognitive approach can be effectively applied to computationally intelligent systems [3]. A swarm consists of a number of particles (possible solutions) that move (fly) through the feasible solution space to explore the optimal solution that can be coded as binary strings or real-valued vectors. The particles are capable of interacting with each other in a given neighborhood, and traverse a search space where a quality measure, fitness can be evaluated. The particles are evolved through cooperation and competition among themselves over iterations. The coordinates of each particle represent a possible solution associated with two vectors, the position ($\boldsymbol{X}_i$) and velocity ($\boldsymbol{V}_i$). In d-dimensional search space, the i^{th} particle can be represented by d-dimensional position vector $\boldsymbol{X}_i = (x_{i1}, x_{i2}, ..., x_{id})$ and another d-dimensional velocity vector $\boldsymbol{V}_i = (v_{i1}, v_{i2}, ..., v_{id})$. Each particle experiences an iterative procedure of adaptation to two types of major information i.e. individual learning and cultural transmission, which means the procedure, accelerates particles at each time step, towards personal best (personal best for each particle) and the position of the most recent global best (best position returned form the swarm) point, with the relative acceleration towards each determined stochastically. A key attractive feature of the PSO approach is its simple mathematical model involving two model equations [3].

In binary PSO [4], velocity loses its physical meaning. It is used to determine a probability by squashing velocities to the range (0,1) by using sigmoid function.

PSO based MIMO Detection

Here we exploit parsimonious binary choice PSO algorithm's potential to optimize MIMO symbol detection [22]. An important step to implement PSO is to define a *fitness function*; this is the link between the optimization algorithm and the real world problem. Fitness function is unique for each optimization problem. The fitness function using the coordinates of the particle returns a fitness value to be assigned to the current location. If the value is greater than the value at respective personal best (*pbest*) for each particle, or global best (*gbest*) of the swarm, then previous locations are updated with the present locations. The velocity of the particle is changed according to the relative locations of *pbest* and *gbest* as shown in Fig. 3.

Once the velocity of the particle is determined, it simply moves to the next position. After this process is applied on each particle in the swarm, it is repeated till the maximum number of iterations is reached. PSO algorithms flow diagram is shown in Fig. 4. This exploratory-exploitive optimization approach can be extended to MIMO detection optimization problem discussed below.

The major challenge in designing Binary PSO based MIMO detector was selection of BPSO parameters that fit the symbol detection optimization problem. In addition, the selection of effective fitness function is vital as well as problem dependent. Fitness function perhaps is the only link between the real world problem and the optimization algorithm. The basic fitness function used by the optimization algorithm to converge to the near optimal solution is (3) that is minimum Euclidian distance. Choice of initial solution plays an important role in the fast convergence of the optimization algorithm to a suitable solution Initial guess is essential for these algorithms to perform. Therefore, our detector takes the output of ZF or ZF-VBLAST as its initial solution guess. This educated guess enables the algorithm to reach more refined solution iteratively by ensuring fast convergence. Assuming random initialization does not guarantee convergence to reasonable solution in lesser iterations.

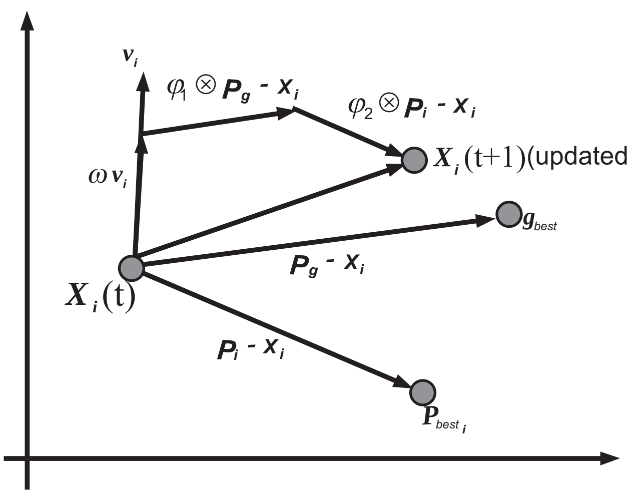

Fig. 3. PSO Visualization

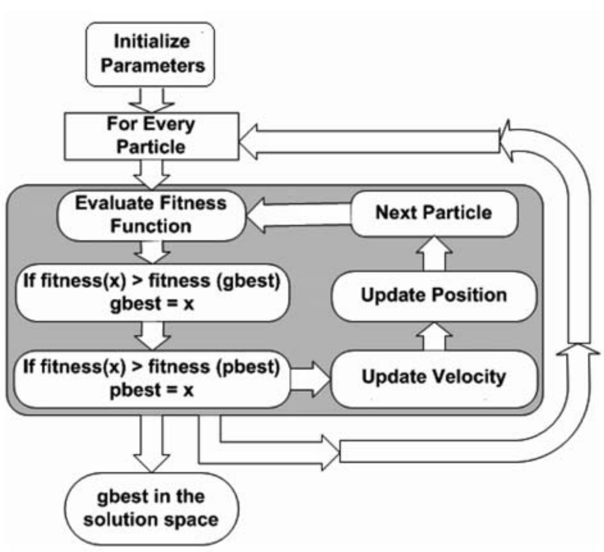

Fig. 4. PSO flow diagram

The proposed detection algorithm is detailed below:

1. Take the output of ZF or ZF-VBLAST such as $\mathbf{x_i} \in \{0,1\}$ as initial particles (initial solution bit string) instead of selecting randomly from the solution space.
2. The algorithm parameters are initialized. 'v_{id}' is initialized to zero (equal probability for binary decision), '$pbest_{id}$' and '$gbest_d$' are initialized to maximum Euclidean distance depending upon the QAM size.
3. Evaluate the fitness of each particle (bit):

$$f = \left\| \mathbf{y} - \mathbf{Hx} \right\|^2 \tag{4}$$

Minimum Euclidean distance for each symbol represents the fitness of solution. Effect on the Euclidean distance due to search space bits is measured. Find the global best performance '$gbest_d$' in the population that represents the least Euclidean distance found so far. Record the personal best '$pbest_{id}$' for each bit along its previous values.

4. For each search space bit at d[th] side of the bit string of particle $\mathbf{x_i}$, compute bits velocity using following PSO velocity update equation:

$$\begin{aligned} v_{id}(k) &= v_{id}(k-1) + \varphi_1 \text{rand}_1 [pbest_{id} - x_{id}(k-1)] \\ &+ \varphi_2 \text{rand}_2 [gbest_d - x_{id}(k-1)] \end{aligned} \tag{5}$$

with $v_{id} \in \{-v_{max}, v_{max}\}$.

5. The particle position is updated depending on the following binary decision rule:

If $\text{rand}_3 < S(v_{id}(k))$, then $x_{id}(k) = 1$, else $x_{id}(k) = 0$. (6)

6. Goto step 3 until maximum number of iterations is reached. The number of iterations is system and requirement dependent (usually kept less than 25 to avoid large complexity). Solution gets refined iteratively.

Here 'k' is the number of iterations, *rand* is a random number generated uniformly in [0,1] and '*S*' is sigmoid transformation function.

$$S(v_{id}(k)) = \frac{1}{1 + \exp(-v_{id}(k))} \tag{7}$$

The parameter 'v_i' is the particles predisposition to make 1 or 0, it determines the probability threshold to make this choice. The individual is more likely to choose 1 for higher $v_{id}(k)$, whereas its lower values will result in the choice of 0. Such a threshold needs to stay in the range of [0,1]. The sigmoid logistic transformation function maps the value of $v_{id}(k)$ to a range of [0,1]. The terms φ_1 and φ_2 are positive acceleration constants used to scale the contribution of cognitive and social components such that $\varphi_1 + \varphi_2 < 4$ [3]. These are used to stochastically vary the relative pull of *pbest* and *gbest*. v_{max} sets a limit to further exploration after the particles have converged. Its values are problem dependent, usually set in the range of [-4,+4] [3].

ACO for spatial multiplexing system

ACO is another attractive Swarm Intelligence technique that is very effective in solving optimization problems that have discrete and finite search space. Since the optimal MIMO detection problem involves a search process across the finite number of possible solutions, ACO is another suitable candidate to solve this problem [23].

Ant colony optimization (ACO)

ACO is based on the behavior of a colony of ants searching for food. In this approach, several artificial ants perform a sequence of operations iteratively. Within each iteration, several ants search in parallel for good solutions in the solution space. One or more ants are allowed to execute a move iteratively, leaving behind a pheromone trail for others to follow. An ant traces out a single path, probabilistically selecting only one element at a time, until an entire solution vector is obtained. In the following iterations, the traversal of ants is guided by the pheromone trails, i.e., the stronger the pheromone concentration along any path, the more likely an ant is to include that path in defining a solution. The quality of produced solution is estimated via a cost function in each iteration. This estimate of a solution quality is essential in determining whether or not to deposit pheromone on the traversed path.

As the search progresses, deposited pheromone dominates ants' selectivity, reducing the randomness of the algorithm. Therefore, ACO is an exploitive algorithm that seeks solutions using information gathered previously, and performs its search in the vicinity of good solutions. However, since the ant's movements are stochastic, ACO is also an exploratory algorithm that samples a wide range of solutions in the solution space.

Binary ant system (BAS)

1) Solution construction: In BAS, artificial ants construct solutions by traversing the mapping graph as shown in Fig 5 below.

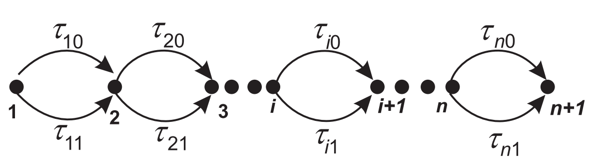

Fig. 5. Routing Diagram for Ants in BAS

A number of n_a ants cooperate together to search in the binary solution domain per iteration. Each ant constructs its solution by walking sequentially from node 1 to node n+1 on the routing graph shown above. At each node *i*, ant either selects upper path i_0 or the lower path i_1 to walk to the next node *i+1*. Selecting i_0 means x_i=0 and selecting i_1 means x_i=1. The selecting probability is dependent on the pheromone distribution on the paths:

$$p_{is} = \tau_{is}(k), i = 1,...,n, s \in \{0,1\} \quad (8)$$

here 'k' is the number of iterations.

2) Pheromone Update: The algorithm sets all the pheromone values as $\tau_{is}(0) = 0.5$, initially but uses a following pheromone update rule:

$$\tau_{is}(k+1) \leftarrow (1-\rho)\tau_{is}(k) + \rho \sum_{x \in S_{upd} | is \in x} w_x \quad (9)$$

Where S_{upd} is the set of solutions to be intensified; w_x are explicit weights for each solution $x \in S_{upd}$, which satisfying $0 \leq w_x \leq 1$ and $\sum_{x \in Supd | is \in x} w_x = 1$. The evaporation parameter ρ is initially as ρ_0, but decreases as $\rho \leftarrow 0.9\rho$ every time the pheromone re-initialization is performed. S_{upd} consists of three components: the global best solution S^{gb}, the iteration best solution S^{ib}, and the restart best solution $S^{rb.}$ w_x combinations are implemented according to the convergence status of the algorithm which is monitored by convergence factor *cf*, given by:

$$cf = \sum_{i=1}^{n} \left| \tau_{i0} - \tau_{i1} \right| \Big/ n \quad (10)$$

The pheromone update strategy in different values of *cf*, are given in table-1, here w_{ib}, w_{rb} and w_{gb} are the weight parameters for S^{ib}, S^{rb} and S^{gb} respectively, cf_i, $i=1,...,5$ are threshold parameters in the range of [0,1]. When $cf > cf_5$, the pheromone re-initialization is preformed according to S^{gb}.

BA-MIMO detection algorithm

The fitness function used by this optimization algorithm to converge to the optimal solution is (4) similar to PS-MIMO algorithm discussed earlier. ZF or ZF-VBLAST output was assumed as initial solution guess to ensure fast convergence. The proposed detection algorithm is described as follows:

1. Take the output of ZF or VBLAST as initial input to algorithm instead of keeping random values, such that $\boldsymbol{x}_i \in \{0,1\}$. Number of nodes *n* visited by n_a ants is bxN_t i.e ML search space size ($\boldsymbol{x}_i$). Here $\boldsymbol{x}_i$ represents the bit strings of the detected symbols at the receiver and *i= 1 to n*.
2. The probability of selecting $\boldsymbol{x}_i$=0 or 1 depends upon the pheromone deposited according to (9). Where $\tau_{is}(0) = 0.5$ for equal initial probability. Evaluate the fitness of solution based on (4). Minimum Euclidean distance for each symbol represents the fitness of solution. Effect on the Euclidean distance due to $\boldsymbol{x}_i$ measured.
3. Pheromone update based on (9) is performed. S_{upd} that consists of S^{gb}, S^{ib}, and S^{rb} is calculated with weights w_x based on *cf* (10) and Table-1.
4. Goto step-2 until maximum number of iterations is reached. The solution gets refined iteratively.

As $cf \rightarrow 0$, the algorithm gets into convergence, once $cf > cf_5$, the pheromone re-initialization procedure is done according to S^{gb}.

$$\tau_{is} = \tau_H \text{ if } is \in S^{gb}$$
$$\tau_{is} = \tau_L \quad otherwise \quad (11)$$

where τ_H and τ_L are the two parameters satisfying $0<\tau_L<\tau_H<1$ and $\tau_L+\tau_H=1$.
The algorithm parameters are set as : $x_i= bN_t$, τ_0=.5, τ_H=.65 and ρ_0=0.3.

Table 1. Pheromone update strategy[7] for BA-MIMO system

	$cf<cf_1$	$cf<(cf_1, cf_2)$	$cf<(cf_2, cf_3)$	$cf<(cf_3, cf_4)$	$cf<(cf_4, cf_5)$
w_{ib}	1	2/3	1/3	0	0
w_{rb}	0	1/3	2/3	1	0
w_{gb}	0	0	0	0	1

Simulation and numerical results

This section provides some simulation and numerical results to prove the performance of the reported MIMO-SI detectors.

BER versus SNR performance

We evaluate these detectors performance for a 4x4, 6x6 and 8x8 N_txN_r MIMO system using 4-QAM scheme and 4x4 MIMO system with 16-QAM constellations. The *SNR* (E_b/N_o) is the average Signal to noise ratio per antenna (P/σ_v^2) where P is the average power per antenna and σ_v^2 is the noise variance. The simulation environment assumes Rayleigh fading channel. The channel is assumed to be quasi-static for each symbol, but independent among different symbols. Perfect sampling and carrier frequency offset synchronization are assumed. Particle size in PSO and Colony Size in case of ACO depends upon N_t and QAM constellation alphabet size. It will be similar to 'solution bit string length'. Therefore, $N_p = b$xN_t where 'b' is bits per symbol. For 4x4, 4-QAM system, N_p or x_i equals 8 and it grows to 16 for 8x8, 4-QAM system. Similarly, number of algorithm iterations (N_{itr} earlier referred as k) depends upon N_t and QAM sizes. 'N_{itr}' is kept in the range of 5 to 20 in our simulations. Iterations can be tuned like other algorithm parameters according to the system requirements. Iterations are according to the system requirements. Larger N_{itr} can result in better BER at the cost of complexity. However, the algorithm reaches saturation after a certain number of iterations and therefore N_{itr} needs to be tuned carefully. Larger N_{itr} can result in achieving better BER performance at the cost of complexity. Optimum N_{itr} value is taken from the algorithms convergence pattern shown in Fig. 14. For PSO, $\varphi_1 = \varphi_2$=1 and v_{max}=±4 are assumed. Similarly, for ACO τ_0=.5, τ_H=.65 and ρ_0=0.3 are assumed in the simulations. An average of no less than 30,000 simulations is taken to report statistically relevant results.

Fig. 6 presents the BER versus E_b/N_o performance of proposed PSMIMO detector compared with ML and VBLAST detectors for 4x4 MIMO system keeping N_{itr} at 10. Initial solution guess is taken from VBLAST for better convergence. At 10^{-4} BER, the PSMIMO results in 5-dB degraded performance in comparison with ML. Whereas BAMIMO algorithm's BER performance shown in Fig. 7 is further deteriorated by 1-dB in comparison with PSMIMO detection technique. However, in comparison with VBLAST, both proposed detection algorithms show 14-dB and 13-dB enhanced BER performance, respectively.

Fig. 8 and Fig.9 show a 6x6 MIMO system. BER gain for both PSMIMO and BAMIMO detectors in comparison with VBLAST is 10-dB and 7-dB, respectively. The BER performance of these detectors is much less than optimal ML however at a significant complexity reduction (discussed next). The algorithm iterations are kept at 15 for these results.

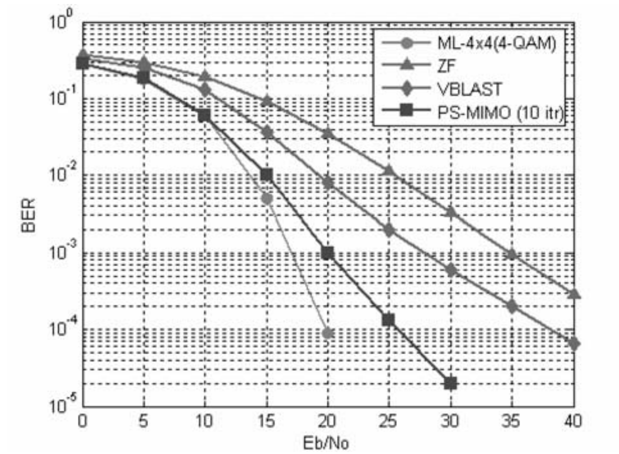

Fig. 6. BER versus E_b/N_o for 4-QAM 4x4 PS-MIMO system.

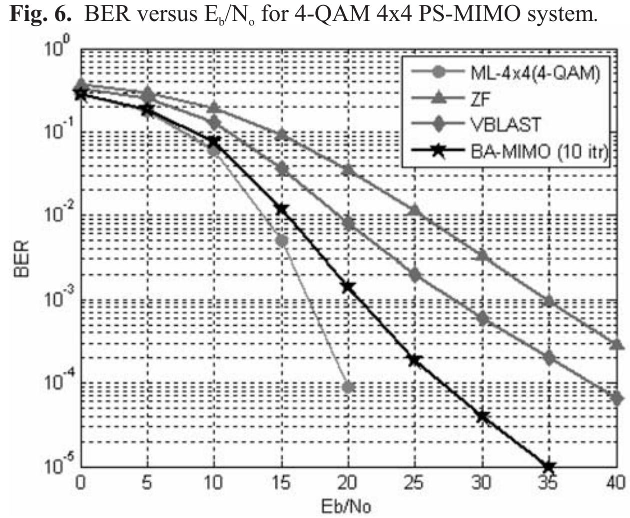

Fig. 7. BER versus E_b/N_o for 4-QAM 4x4 BA-MIMO system.

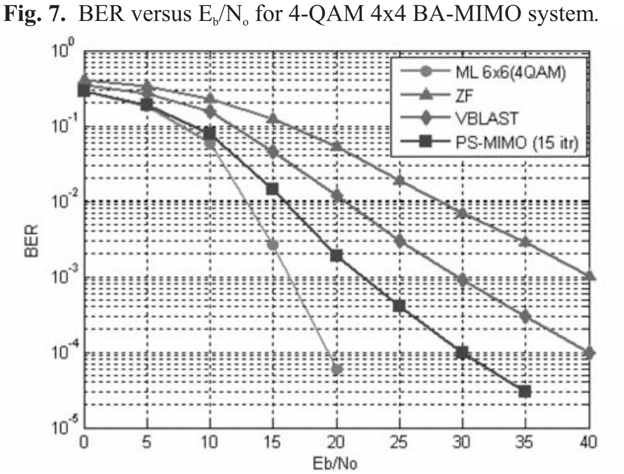

Fig. 8. BER versus E_b/N_o for 6x6 4-QAM PS-MIMO system

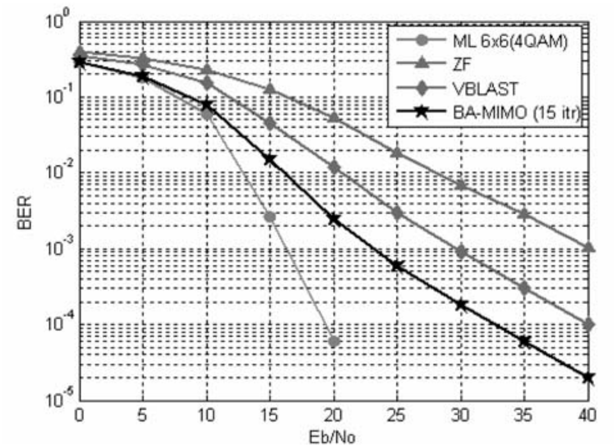

Fig. 9. BER versus E_b/N_o for 6x6 4-QAM BA-MIMO system.

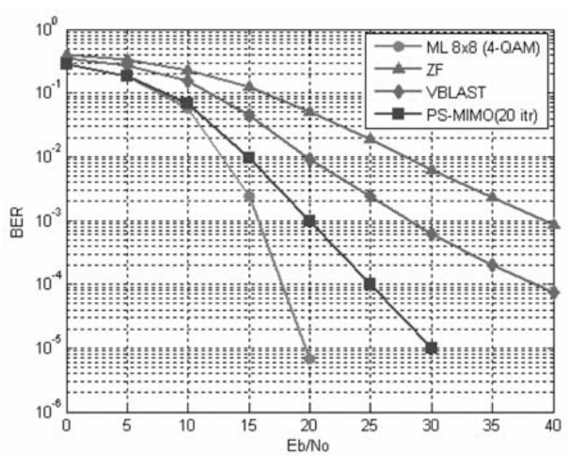

Fig. 10. BER versus E_b/N_o for 8x8 4-QAM PS-MIMO system

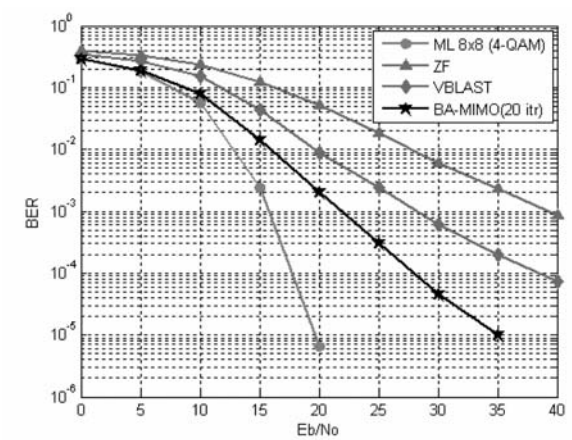

Fig. 11. BER versus E_b/N_o for 8x8 4-QAM BA-MIMO system

Similarly, for a 8x8 ($N_{itr}=20$) system in Fig. 10 and Fig. 11, the BER improvement in comparison to VBLAST for both the proposed detectors is significant. However, ML has superior BER performance but its complexity is also considerable.

Increase in system size (N_txN_r), results in exponential increase of search space, therefore more algorithm iterations are required to converge to near-optimal solution. These heuristics tend to saturate after few iterations, therefore an optimum number of iterations are selected for efficient performance. A trade off between systems BER performance and iterations has to be maintained according to the system requirement and priority.

Computational complexity comparison

Here we examine the computational complexity of the reported detectors and compare it with ML and VBLAST detectors. As the hardware cost of each algorithm is implementation-specific, we try to provide a rough estimate of complexity in terms of number of complex multiplications. The computational complexity is computed in terms of the N_t, N_r and the constellation size M.

For ML detector as seen from (3) $M^{Nt}(N_rN_t)$ multiplications are required for matrix multiplication operation and additional $M^{Nt}N_r$ multiplications are needed for square operation. Therefore, ML complexity becomes:

$$\gamma_{ML} = N_r(N_t+1)M^{N_t} \tag{12}$$

In case of ZF, the pseudo-inverse of matrix $(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$ takes $4N_t^3+2N_t^2N_r$ multiplications [24]. Therefore, ZF complexity becomes:

$$\gamma_{ZF} = 4N_t^3+2N_t^2N_r \tag{13}$$

For VBLAST the pseudo-inverse matrix is calculated N_t times with decreasing dimension. In addition, the complexity of ordering and interference canceling is $\sum_{i=0}^{N_t-1}[N_t(N_t-i)+2N_t]$. Therefore, total complexity of VBLAST (γ_{VBLAST}) results in

$$\gamma_{VBLAST} = \sum_{i=0}^{N_t}(4i^3+2N_ri^2)+\sum_{i=0}^{N_t-1}[N_t(N_t-i)+2N_t] \tag{14}$$

$$= N_t^4+(5/2+2/3N_r)N_t^3+(7/2+N_r)N_t^2+1/3N_tN_r \tag{15}$$

For the proposed detector, first fitness using (3) is calculated. Here $N_p = \boldsymbol{x}_i$. Multiplication complexity (γ_{SI}) becomes:

$$\gamma_{SI} = N_p(N_tN_r) \tag{16}$$

Velocity update in PSO and pheromone update in case of ACO require μ_{vel} additional multiplications per iteration from (5) and (9). To reduce some complexity w=1 and $\varphi_1+\varphi_2=2$ is assumed in PSO. Therefore μ_{si}, becomes 2, the complexity becomes:

$$\gamma_{SI} = N_p(N_tN_r+\mu_{si}) \tag{17}$$

This procedure is repeated N_{itr} times to converge to the near-optimal BER performance. Therefore,

$$\gamma_{SI} = N_p(N_tN_r+\mu_{si})N_{itr} \tag{18}$$

Table 2. Complexity Comparison (Complex Multiplications) N_txN_r MIMO ystem

Method	4x4(4-QAM)	6x6(4-QAM)	8x8(4-QAM)	4x4(16-QAM)
ML (γ_{ML})	5120	30720	4.7 M	1.3M
VBLAST (γ_{VBLAST})	712	3054	8864	712
Proposed Detectors	*(N_p=8, N_{itr}=10, μ_{vel}=2)*	*(N_p=10, N_{itr}=15, μ_{vel}=2)*	*(N_p=16, N_{itr}=20, μ_{vel}=2)*	*(N_p=16, N_{itr}=10, μ_{vel}=2)*
PS-MIMO	2152	9894	30624	3592
BA-MIMO	2152	9894	30624	3592

These SI-MIMO detectors take initial solution guess as ZF or VBLAST output therefore, it is added into get the resultant complexity $\gamma_{SI\text{-}Tot}$.

$$\gamma_{SI-Tot} = \gamma_{SI} + (\gamma_{VBLAST} \ or \ \gamma_{ZF}) \quad (19)$$

From (12) it is observed that the complexity of ML is exponential with N_t and M. ML complexity for a 4-QAM 4x4 system is 5120 and it grows to 4.7 M for 8x8 system. This increase is even significant with higher order modulation schemes.

Computational complexity of VBLAST for 4-QAM 4x4, 6x6 and 8x8 systems computed from (15) is 712, 3054 and 8864 respectively. The complexity of proposed detectors with VBLAST initialization for 4x4, 6x6 and 8x8 configurations is comes out to be 2152, 9894 and 30624 respectively. A detailed complexity comparison is shown in Table 2. However, this complexity estimate is only meaningful in the order of magnitude sense since it is based on the number of complex multiplications only.

Performance-complexity trade-off

Table 3, suggests that a reasonable performance-complexity trade-off exists when a comparison of the proposed detectors is drawn with ML and VBLAST detectors. Compared with ML the complexity reduction of the proposed detector is significant for larger MIMO systems where ML is not practical to use. However, this complexity gain is at the cost of degraded BER performance. For a 4x4, 4-QAM system, at 10^{-4} BER the performance of proposed PS and BA detectors is degraded by 5-dB and 6-dB, respectively with 58% complexity reduction. Similarly, in 8x8, 4-QAM system, the proposed algorithm achieves 10^{-4} BER at 6-dB more SNR than ML, however, the ML complexity reduction is as high as 99%. When compared with VBLAST the proposed detector complexity increase is approximately 70 % with a BER gain up to 14-dB for larger MIMO system.

Performance in Higher Order Modulation Schemes

The performance of PSMIMO and BAMIMO detectors in a 16-QAM MIMO system is shown in Fig. 12 and Fig. 13, respectively. We observe a consistent BER performance of the proposed techniques even at larger QAM systems. BER degradation in comparison to ML is approximately 6-dB. However, the complexity reduction calculated in Table-4 with this BER performance trade-off is convincingly high. VBLAST detector has reduced complexity as compared to the proposed detectors but its BER performance is also significantly less.

Effects of change in algorithm parameters and increase in iterations

These SIMIMO detectors converge to near optimal performance with increase in the algorithm iterations, however these algorithm also experience saturation after reaching a particular threshold BER. Therefore a perfect algorithm tuning at an optimum N_{itr} for efficient performance is required. Fig-14 shows the convergence of PSMIMO algorithm with an increase in iterations for ZF and VBLAST initial inputs. The algorithm gets saturated

Table 3. Performance Complexity Trade-Off N_txN_r MIMO system

Performance complexity comparison of the proposed detectors with VBLAST and ML		Proposed Detectors (at 10^{-4} BER)	4x4 (4-QAM)	6x6 (4-QAM)	8x8 (4-QAM)	4x4 (16QAM)
ML and proposed detectors	PS-MIMO Detector	Complexity reduction	58%	68%	99%	99%
		Performance degradation	5-dB	10-dB	8-dB	6-dB
	BA-MIMO Detector	Complexity reduction	58%	68%	99%	99%
		Performance degradation	6-dB	13-dB	10-dB	8-dB
VBLAST and proposed detectors	PS-MIMO Detector	Complexity increase	67%	69%	71%	80%
		Performance improvement	14-dB	10-dB	14-dB	10-dB
	BA-MIMO Detector	Complexity increase	67%	69%	71%	80%
		Performance improvement	12-dB	7-dB	11-dB	8-dB

at around 10 iterations with VBLAST input and 15 iterations for ZF initialization case. Therefore, N_{itr} is kept at 10 for a 4x4 MIMO system. Similarly, iterations tuning for the other systems is performed to find the optimum N_{itr}. Choice of good initial guess has an effect on the detectors convergence as can be seen from Fig 12. Here VBLAST output as initial guess results in faster convergence than ZF initial input.

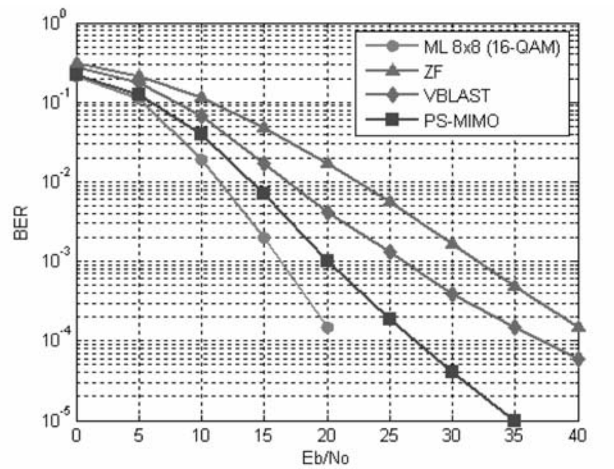

Fig. 12. BER versus E_b/N_o for 16-QAM 4x4 PS-MIMO system

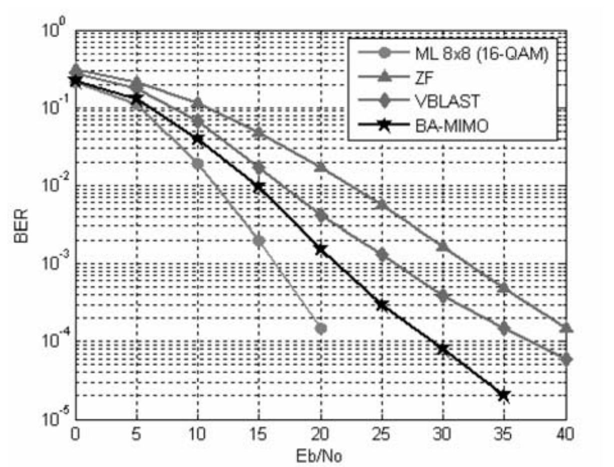

Fig. 13. BER versus E_b/N_o for 16-QAM 4x4 BA-MIMO system

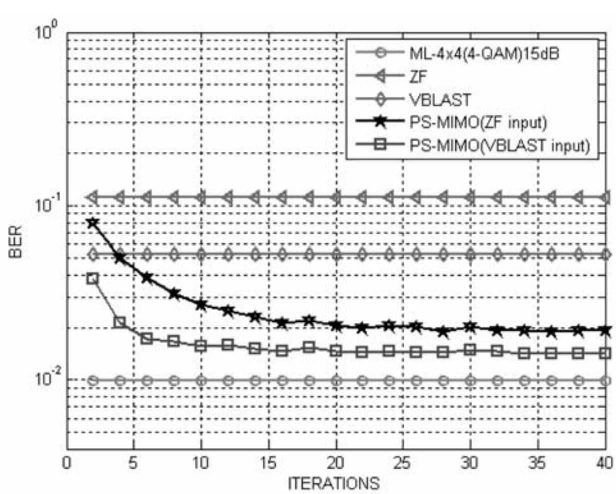

Fig. 14. Convergence with iterations at 15-dB

Fig.15 presents the effect of changing the algorithm parameters on the detectors performance. Values of the cognitive component (c_1) and social component (c_2) are changed. Results in Fig. 15 assume c_1+c_2= .5, 1 and 2. Larger values of social and cognitive components results in an improvement in BER performance. A possible reason for this can be that more fly over and coming back to better solution is achieved with higher c_1 and c_2 values.

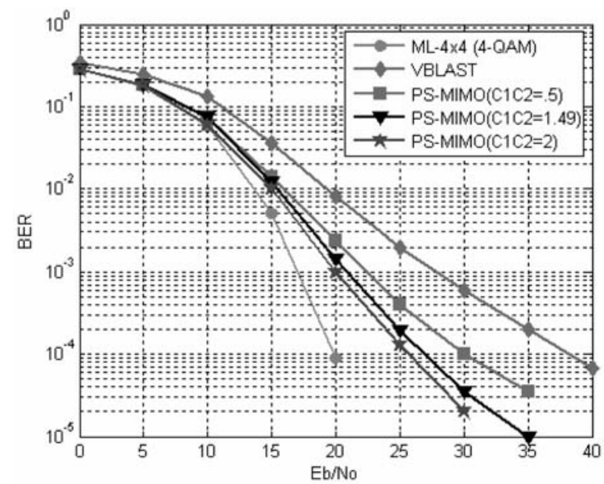

Fig. 15. Effect on BER of PS-MIMO algorithm with C1 and C2

Analysis of Swarm Intelligence assisted detection algorithms

Swarm Intelligence assisted detection approach show promising results. Their simple mathematical model, lesser implementation complexity, resistance to being trapped in local minima and convergence to reasonable solution in lesser iterations make these nature inspired techniques a suitable candidate for real-time symbol detection in Spatial Multiplexing System. SI algorithms imitate nature's own ingenious ways to explore the search space to find out optimal solution. PSO uses the intelligence derived from the coordinated movements of birds, whereas ACO is inspired form cooperative forging strategy of ants. The efficiency of these algorithms also lies in a simple computer code in the central algorithm with few parameters to tune. In this particular MIMO detection application, PSO has outperformed ACO in terms BER performance, whereas the computational complexity is same.

Conclusion

In this paper an application of Computational Swarm Intelligence for symbol detection in spatial multiplexing system is presented. These heuristics prove to be powerful function optimizers. Their simple model with lesser implementation complexity makes these suitable for this NP-hard wireless communications problem. The algorithms show promising results when compared with the optimal ML and traditional VBLAST detectors. SI optimized MIMO symbol detection methods approach near-optimal performance with much reduced computational complexity, especially for complex systems with multiple transmitting antennas, where conventional ML detector is computationally expensive and impractical to deploy.

Although VBLAST detector has a reduced complexity, its BER performance is inferior to the proposed detector. The simulation results suggest that the proposed detector improves ML complexity by as high as 99% with 6-dB to 8-dB BER performance degradation for 8x8 MIMO system.

REFERENCES

1. A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley and Sons, 2005.
2. J. Kennedy and R. C. Eberhart, “Particle Swarm Optimization,” *Proc. IEEE International Conference on Neural Networks IV*, Piscataway, NJ, 1995.
3. J. Kennedy and R. C. Eberhart, *Swarm intelligence*, Morgan Kaufman, 2001.
4. J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm”, *IEEE International Conference on Systems, Man, and Cybernetics*, 1997, pp. 4104.
5. M. Dorigo and T. Stutzle, *Ant Colony Optimization*, Cambridge, MA: MIT press, 2004.
6. M. Dorigo and L. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, Vol. 1, Apr. 1997, pp. 53-66.
7. C. Blum, M. Dorigo, “The hyper-cube framework for ant colony optimization”, *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, Vol. 34, No. 2, 2004, pp. 1161-1172.
8. M. Kong and P. Tian, “Introducing a Binary Ant colony Optimization,” *5th International Workshop on Ant Colony Optimization and Swarm Intelligence*, September 2006.
9. A. Goldsmith, S. A. Jafar, N.Jindal, and S. Vishwanath, “Capacity limits of MIMO channels”, *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 5, June 2003.
10. Syed Aon Mujtaba, *TGn Sync Proposal Technical Specification*, Agere Systems Inc., U.S.A., May 2005, http://www.tgnsync.org/techdocs/11-04-0889-06-000n-tgnsyncproposal-technical-specification.doc.
11. L. Zheng and D. Tse, “Diversity and multiplexing: a fundamental tradeoff in multiple antenna channels,” *IEEE Tran. Inform. Th.*, Vol. 49, May 2003, pp. 1073-96.
12. H. Bolcskei, D. Gesbert, C. and Papadias , *Space-Time Wireless Systems: From Array Processing to MIMO Communications*, Cambridge University Press, 2005.
13. G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multiple antennas,” *Bell Labs Technical Journal*, Vol. 1, August 1996, pp. 41-59.
14. G. J. Foschini and M. J. Gans, “On limits of wireless communications in a fading environment when using multiple antennas,” *Wireless Personal Communications*, Vol. 6, No. 3, 1998, pp. 311-335.
15. W. J. Choi, R. Negi and J. M. Cioffi, “Combined ML and DFE decoding for the V-BLAST system”, *Proc. IEEE International Conference on Communications 2000*, New Orleans, LA, Jun. 2000, pp. 1243-1248.
16. W. J. Choi, K. W. Cheong and J. M. Cioffi, “Iterative soft interference cancellation for multiple antenna systems,” in *Proc. IEEE Wireless Communications and Networking Conference 2000*, Chicago, IL, Sep. 2000, pp. 304-309.
17. S. Haykin, *Adaptive Filter Theory*, 3rd ed., Prentice-Hall, 1996.
18. A. Sayed, *Fundamentals of Adaptive Filtering*, Wiley-IEEE Press, 2003.
19. H. H. El-Mora, A.U. Sheikh and A. Zerguine, “Application of Particle Swarm Optimization Algorithm to Multiuser Detection in CDMA”, *16th IEEE International Symposium on Personal, Indoor and Mobile Radio communications*, 2005.
20. Samer L. Hijazi, “Multiuser Detection for Multicarrier Communication Systems” PhD dissertation, Kansas State University, 2006.
21. A. A. Khan, M. Naeem and S. I. Shah, “A Particle Swarm Algorithm for Symbols Detection in Wideband Spatial Multiplexing Systems”, *Genetic and Evolutionary Computation Conference*, GECCO- 2007, pp. 63-69.
22. A. A. Khan, S. Basher, M. Naeem, S. I. Shah and A.U. Sheikh, “Binary Ant Colony Algorithm for Symbol Detection in a Spatial Multiplexing System”, *Unconventional Computation*, UC 2007, LNCS 4618, pp. 115-126.
23. G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, 1996.