13

Computationally Efficient Symbol Detection Schemes in Multi-Device STBC-MIMO Systems

Daniel C. Lee and Muhammad Naeem
Simon Fraser University, School of Engineering Science, Burnaby, BC, Canada

1. Introduction

The multi input multi output (MIMO) communication system has significantly higher channel capacity than the Single-Input-Single-Output (SISO) system for the same total transmission power and bandwidth (Foschini et al., 1998 & Telatar, 1999). It is known that the use of Space Time Block Code (STBC) can realize the increased capacity of MIMO systems and thus improve data throughput and spectral efficiency (Tarokh et al., 1998). In this work, we focus on the system that comprises one receiving station and multiple transmitting devices (e.g., uplinks in cellular systems). The receiver's front end has multiple antennas, and each transmitting device has multiple transmit antennas. The system discussed in this chapter is illustrated in Fig. 1. A Space Time Block Code (STBC) is used in each transmitting device, and joint signal detection is performed at the receiver. We refer to such a system as Multi-Device (MD) STBC-MIMO system. Generally in a MD-STBC-MIMO system, the number of receive antennas is typically smaller than the number of all transmit antennas used by all transmitting devices in the system. An example of MD-STBC-MIMO would be the uplink multiple access communication system, where the number of receive antennas at the base station or the access point is smaller than the total number of transmit antennas at the mobile devices.

In this work, we address symbol detection in Multi-Device (MD) STBC-MIMO systems. As will be discussed in section 3, the maximum a posteriori (MAP) detection, which reduced to the Maximum Likelihood (ML) detection in the case of a priori equally likely symbol blocks, minimizes the probability of detection error, and thus is optimal. However, a computationally efficient algorithm for achieving MAP or ML detection is not known. Some studies with sphere decoding (SD) algorithms exhibit that their expected computational complexity grows polynomially with the problem size, say m, up to some value of m for the cases of small constellation sizes (Vikalo et al., 2005), but it grows exponentially for the cases of large constellation sizes. Also, for some sphere decoding algorithms, operation at a low SNR requires inordinately high computation, although operation at a high SNR is efficient. In any case, an algorithm with polynomial growth of expected complexity for all values of the problem size, m, has not yet been found. In fact, Jalden et al. (2005) shows that even the expected computational complexity of the sphere decoding grows exponentially with the problem size in MIMO communication systems.

In this work, we present two evolutionary optimization methods, Biogeography-Based Optimization (BBO) and Estimation of Distribution Algorithm (EDA) to solve the problem

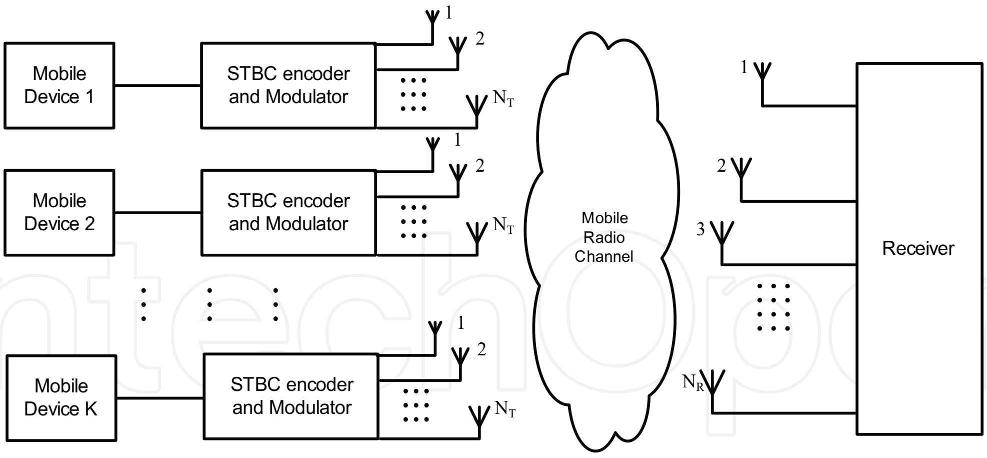

Fig. 1. A block diagram of a MD-STBC-MIMO system

of detection in a MD-STBC-MIMO system. Our simulation results show that BBO and EDA can meet the best known detector (i.e., sphere decoder) with less complexity and has better performance than other methods such as Minimum Mean Square Error (MMSE), Vertical Bell Laboratories Layered Space-Time (V-BLAST) (Wolniansky et al. ,1998), Semi-Definite Relaxation (SDR) (Nekuii et al. 2008), and genetic algorithm (GA).

2. System model

Fig. 1 shows an MD-STBC-MIMO system with one receiver and multiple transmit devices. Each of the K mobile devices (information senders) has N_T transmit antennas that apply STBC, and the receiver's front end has N_R receive antennas. The multiple devices in the system can cause co-channel interference. An IQ-modulation scheme (e.g. MPSK, M-QAM, etc.) maps source information into complex numbers. Even if transmit devices each employ an orthogonal space-time code, orthogonality among their signals cannot be guaranteed due to the absence of coding across different mobile devices.

First, let us consider the case of single mobile device; i.e. K=1. The mobile device transmits its signals through N_T transmit antennas, and the receiver has N_R antennas. We denote by T the number of time slots in the space-time code block. We assume that the channel is quasi-static; i.e., the channel gain remains constant during each time block of data. We also assume that the channel gain at each time block is known to the receiver. This assumption is often used in the literature and is reasonable if training or pilot signals are used. A complex $N_T \times N_R$ matrix H represents the MIMO channel and another complex $T \times N_T$ matrix S represents the input signal in the space-time code block. The relationship between the input and output signal is

$$\tilde{Y} = SH + \tilde{Z} \tag{1}$$

where $\tilde{Y}$ is the $T \times N_R$ complex output matrix, and $\tilde{Z}$ represents the additive white noise matrix.

In analyzing the system with linear dispersion space-time coding, the relation between the input and output of the channel is often expressed in another form (Hassibi et al., 2002) than (1). We now briefly describe this alternative form. An input signal, denoted by matrix S, in the space time block code (STBC) can be expressed as:

$$S=\sum_{q=1}^{Q}\left[\left(\alpha_q+j\beta_q\right)C_q+\left(\alpha_q-j\beta_q\right)D_q\right] \tag{2}$$

where Q is the number of symbols conveyed in a space time code block, and $\alpha_q + j\beta_q$, $q=1,\ldots,Q$ are complex numbers that represent the Q symbols. (Note that α_q and β_q denote the real and imaginary parts of a symbol.) Then, the Q symbols can be represented as a $2Q$-dimensional real-valued row vector χ, where components of χ are constituted by α_q and β_q, $q=1,\ldots, Q$. The real and imaginary parts of matrix $\tilde{Y}$'s components can be arranged as a $2TN_R$ -dimensional real-valued row vector y. In this alternative form, χ and y are arranged in such a way that their relation is expressed as:

$$\mathrm{y}=\chi\,\Omega\;+\mathrm{Z} \tag{3}$$

where $2Q\times 2TN_R$ real-valued matrix Ω is derived from the component of matrices H, C_q, D_q, $q=1,\ldots,Q$, and Z is the $2TN_R$ -dimensional real-valued vector representing noise.
In the case of multiple mobile devices, equation (1) is naturally generalized to

$$\tilde{Y}=\sum_{k=1}^{K}S_kH_k+\tilde{Z} \tag{4}$$

where the $T\times N_T$- complex matrix S_k is the input signal from mobile device k, and the $N_T\times N_R$-complex matrix H_k represents the channel from the kth device to the receiver. Correspondingly, (3) is naturally generalized to

$$y=\begin{bmatrix}\chi_1 & \chi_2 & \cdots & \chi_K\end{bmatrix}\begin{bmatrix}\Omega_1\\ \Omega_2\\ \vdots\\ \Omega_K\end{bmatrix}+Z \tag{5}$$

where χ_k is a $2Q_k$-dimensional real-valued row vector that represents the Q_k complex symbols sent from mobile device k in a space time code block. Note that (5) can model the case in which different mobile devices use different code rates Q_k/T and different space time codes. We denote by $N_S=\sum_{k=1}^{K}Q_k$ the total number of symbols (from all mobile devices) transmitted in a space-time coded block through all of their transmit antennas.

3. Signal detection

The detector at the receiver has to choose from M^{N_S} possible sequences of symbols transmitted in a space-time code block, where M is the size of the symbol constellation associated with the modulation scheme. ML detection is known to yield the lowest symbol error probability in the case of a priori equally likely symbols. ML detection chooses transmitted symbols $\left[\chi_1,\chi_2,\cdots,\chi_K\right]$ that maximize $P\left(y\middle|\chi_1,\chi_2,\cdots,\chi_K\right)$. In the case of additive white Gaussian noise, Z, the ML detection is reduced to choosing the vector $\left[\chi_1,\chi_2,\cdots,\chi_K\right]$, from M^{N_S} possibilities, that has the shortest Euclidean distance:

$$\left\| y - \sum_{k=1}^{K} \chi_k \Omega_k \right\| \tag{6}$$

Defining $b = \log_2 M$, and M as the size of the symbol constellation, the ML detection scheme can be implemented by searching through all $M^{N_S} = 2^{bN_S}$ possible symbol sequences. Performing such an exhaustive search to find the minimum of (6) is computationally inefficient, especially for large N_S. Computational complexity increases exponentially with $N_S = \sum_{k=1}^{K} Q_k$, the number of bits per symbol, transmit antennas per device, and the number of mobile devices (K). High-speed communication requirements demand a low-complexity detection scheme. For low-complexity near-optimal detection, in this chapter we apply two population-based evolutionary algorithms, biogeography-based optimization (BBO) and estimation-of-distribution algorithm (EDA), to this MD-STBC-MIMO detection problem. The MD-STBC-MIMO detection problem is converted into a discrete optimization problem that searches the space of $M^{N_S} = 2^{bN_S}$ symbol combinations. Namely, the receiver's MD-STBC-MIMO detection problem is to find the value of $\left[\chi_1, \chi_2, \cdots, \chi_K\right]$ that minimizes cost function $\left\| y - \sum_{k=1}^{K} \chi_k \Omega_k \right\|$ for a received signal y and the known channel condition $\Omega_1, \Omega_2, \cdots, \Omega_K$. In section 4, we describe how we can apply BBO and EDA to the MD-STBC-MIMO signal detection problem.

4. BBO and EDA algorithms

Population-based evolutionary algorithms (EAs) in general have been often used to solve difficult optimization problems. Candidate solutions to an optimization problem are represented as individuals in the population. Most of the evolutionary algorithms are inspired by the theory of biological evolution (e.g., selection, crossover, mutation, recombination, and reproduction). In EAs the objective function value of a candidate solution can be considered as the fitness of the individual in the concept of natural selection. For the MD-STBC-MIMO detection problem, expression (6) can be used as the fitness function, where the smaller value of (6) means the better fitness. If each candidate solution being represented as a binary string, the search space is $I_s = \{0,1\}^n$ where $n = N_S \log_2 M$. The MD-STBC-MIMO detection problem can be converted into a binary optimization problem by associating M^{N_S} symbols with 2^{bN_S} bit strings. Each of the solutions has length, $n = N_S \log_2 M$ bits. In this section, we present a MD-STBC-MIMO detector that utilizes BBO-based and EDA-based evolutionary algorithms.

4.1 BBO

Biogeography-based optimization (Simon, 2008) is a population-based, stochastic global optimization EA, which is based on the mathematics of biogeography theory. Biogeography is the study of the geographical distributions of biological organisms. Mathematical models of biogeography describe how species migrate from one island to another, how new species arise, and how species become extinct.

Consider an optimization problem:

$$\begin{aligned} &\max_x && F(x) \\ &\text{subject to} && x \in X \end{aligned} \tag{7}$$

where $x \equiv (x_1, x_2, \ldots x_m)^T$ is a vector and X is a constraint set. In the original BBO, each candidate solution is represented by a vector variable of the optimization problem. In the context of evolutionary algorithms, a candidate solution is also referred to as an "individual," and a group of candidate solutions is referred to as a "population" of individuals. In BBO, each individual (candidate solution to an optimization problem) is analogically considered as an island (habitat) in Biogeography. The fitness value, $F(x)$, of each individual x corresponds to the Habitat Suitability Index (HSI) of an island in Biogeography. In Biogeography, features that affect HSI include vegetation, rainfall, topographic diversity, temperature, etc., and these features are characterized by variables that are called Suitability Index Variables (SIVs). As mentioned earlier, a candidate solution $x \equiv (x_1, x_2, \ldots x_m)^T$ in optimization problem (7) analogically corresponds to an island (a habitat) in Biogeography. Then, components $x_1, x_2, \ldots x_m$, of x correspond to its SIVs, and F(x) correspond to the HSI of island x. We will often use these terminologies to refer to a candidate solution $x \equiv (x_1, x_2, \ldots x_m)^T$, its components, $x_1, x_2, \ldots x_m$, , and the fitness value of a candidate solution x.

A good solution indicates an island with a high HSI, which are well suited as habitats for biological species. An island with a high HSI tends to have a large number of species, while an island with a low HSI tends to have a small number of species. An island with a high HSI tends to have a low immigration rate because it is already saturated with species (Simon, 2008). Also, many species emigrate from high-HSI islands to nearby habitats, as animals ride flotsam, fly or swim to neighboring islands. These habitats are said to have high emigration rates. Suppose there are several candidate solutions to a problem. A good solution is analogous to an island with a high HSI, while a poor solution is analogous to an island with a low HSI. High-HSI solutions are more likely to share their features (SIVs) with other habitats, which is analogous to emigration. Low HSI habitats tend to accept features of other solutions, which is analogous to immigration. Through this kind of probabilistic evolution, biogeography-based optimization searches for a good solution to an optimization problem.

We denote by λ and μ the immigration rate and emigration rate, respectively. Immigration rate λ and emigration rate μ are functions of the island's HSI (or equivalently, the number of species), S, on the island. An island has the maximum possible immigration rate I when there is no species in the island. As the number of species increases and the island becomes more crowded, the immigration rate decreases because fewer species can successfully survive immigration. The immigration rate is zero if the island has the largest possible number of species that the island can support. Similarly, when there is no species in the island, no species can emigrate from it, so its emigration rate is zero. The maximum emigration rate E occurs when the island contains the largest possible number of species. In short λ is a non-increasing function of HSI S, and is a non-decreasing function of S. Simple examples would be the linear functions illustrated in Figs. 2a and 2b, which can be mathematically expressed as

$$\lambda(S) = I\left(1 - \frac{S}{S_{\max}}\right), \text{ for } 0 \le S \le S_{\max} \tag{8}$$

$$\mu(S) = E\frac{S}{S_{\max}}, \text{ for } 0 \le S \le S_{\max} \tag{9}$$

Parameters and notations used in BBO are summarized in table 1, and the step-by-step algorithm of BBO used for the purpose of this chapter is presented in Fig. 3. We use the notation p_i to denote the ith population member (island) and $p_i(s)$ to denote the sth SIV of the ith population member.

λ_i	Immigration rate into the i^{th} island in the population
μ_i	Emigration rate from the i^{th} island in the population
$S_{\max}$	The maximum number of species that a habitat can support
p	The population size (the number of islands in the population)
p_i	The ith island in the population
$p_i(s)$	The sth SIV of the i^{th} population member (island)
q	The probability of mutation
I	The maximum possible immigration rate
E	The maximum possible emigration rate
g	The number of iterations used as termination condition

Table 1. Parameters and notation of BBO

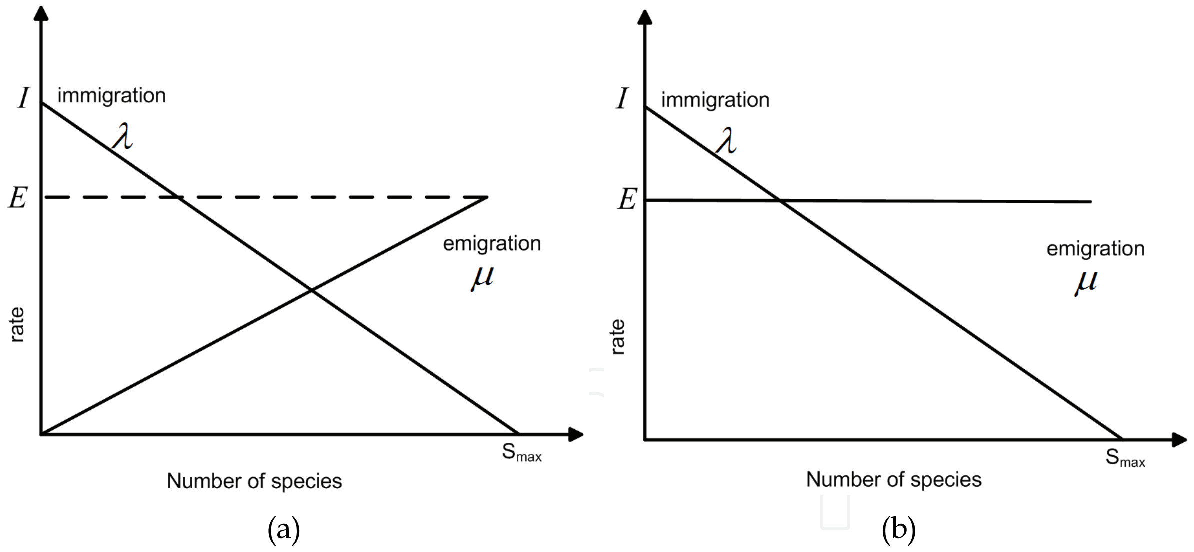

Fig. 2. (a) Typical migration Model used for BBO (b) Low- complexity migration Model for BBO

Various algorithms have been developed that use different migration schemes (Simon, 2008). The migration algorithm we use is basically a simplified version of the partial immigration-based BBO. We used a linear, decreasing λ (immigration rate) curve with a maximum of I and a constant μ (emigration rate) equal to E, as illustrated in Fig. 2b, in order to reduce computation. This constant emigration rate reduces the process of selecting the island that will emigrate to each island that is decided to accept immigration. Our preliminary results indicated that this computational simplification resulted in BER comparable to the implementation of (Simon, 2008).

```
For each iteration g
  For each island $p_i$
    For each SIV s
      With probability $\lambda_i$ , $p_i(s)$ is decided to accept immigration;
      If $p_i(s)$ is decided to accept immigration, then
        Select one island, $p_j$, that emigrates to $p_i$ from the rest of the population with
          equally likely probability;
          Assign $p_i(s) = p_j(s)$ ($p_j(s)$ emigrates into $p_i(s)$);
      End If
    Next SIV
  NEXT Island

  For each island $p_i$
    For each SIV s
      With probability q,   $p_i(s)$ is decided to mutate;
      If $p_i(s)$ is decide to mutate, then
        Replace $p_i(s)$ with a randomly generated SIV;
      End If
    Next SIV
  NEXT island

  For each island $p_i$
    Calculate HIS
  NEXT island

Sort population
NEXT generation
```

Fig. 3. One generation of BBO pseudo code

BBO takes the advantage of mutation. In each island in each generation, each SIV mutates with a particular probability. Simon (2008) introduces a procedure of assigning different probabilities of mutation. His mutation approach tends to increase diversity among the population. The aim of this scheme is to make an island with low HSI more likely to mutate its SIVs. However, we use a less complex approach of equal probability of mutation, *q* for all islands and all SIVs. Our results indicated that this equal mutation probability did not much influence the BER performance in our symbol detection problem.

4.2 EDA

Unlike other population-based evolutionary algorithms such as the genetic algorithm, in EDA the individuals are generated without the crossover and mutation operators. Instead, in EDA, a new population is generated based on a probability distribution, which is estimated from the best-selected individuals of the previous iterations (Larrañaga et al., 2001).

In general, conventional EDAs can be characterized and described by parameters and notations $(I_s, F, \Delta_l, \eta_l, p_s, \Gamma, I_{Ter})$, where

1. I_s denotes the space of all potential solutions (entire search space of individuals).
2. F denotes the fitness function.
3. Δ_l denotes the population (the set of individuals) at the l_{th} iteration.
4. η_l denotes the set of best candidate solutions selected from set Δ_l at the l_{th} iteration.
5. p_s is the selection probability. The EDA selects $p_s|\Delta_l|$ individuals from set Δ_l to make up the set η_l.
6. We denote by Γ_l the distribution estimated from η_l (the set of selected candidate solutions) at each iteration.
7. I_{Ter} is the maximum number of iterations.

In conventional EDAs each individual can be designated by a binary string of length n (n-dimensional binary vector). We denote by a binary row vector $X = (x_1, x_2, \cdots, x_n)$, $x_i \in \{0,1\}$ as an individual. In each iteration, an EDA maintains a population of individuals. We denote by $|\Delta_l|$ the number of individuals in population Δ_l. Population Δ_l can be specified by the following matrix

$$\Lambda = \begin{pmatrix} X^1 \\ X^2 \\ \vdots \\ X^{|\Delta_l|} \end{pmatrix} = \begin{pmatrix} x_1^1 & x_2^1 & \vdots & x_n^1 \\ x_1^2 & x_2^2 & \vdots & x_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ x_1^{|\Delta_l|} & x_2^{|\Delta_l|} & \vdots & x_n^{|\Delta_l|} \end{pmatrix} \tag{10}$$

where superscript j in the row vector $X^j = \left(x_1^j, x_2^j, x_3^j, \ldots, x_n^j\right)$ indexes an individual in the population. A typical EDA applied to the MD-STBC-MIMO detection problem can be described in the following steps:

Step 0: *Generate an initial population Δ_0. The initial population ($|\Delta_0|$ individuals) is typically obtained by sampling according the uniform (equally likely) distribution (Larrañaga et al., 2001):*

$$p(\theta_1, \theta_2, \cdots, \theta_n) = \prod_{i=1}^{n} p_i(\theta_i),$$
$$p_i(\theta_i = 1) = p_i(\theta_i = 0) = 0.5,\ i = 1,2,\ldots,n. \tag{11}$$

(*In accordance with Eqn. (11), in a typical EDA the joint probability distribution from which the individuals are sampled is factorized as a product of n univariate marginal probability distributions, each following a Bernoulli distribution with parameter value equal to 0.5.*

For iterations l = 1,2, ..., follow Step 1 through Step 6:

Step 1: *Evaluate the individuals in the current population Δ_{l-1} according to the fitness function F. Sort the candidate solutions (individuals in the current population) according to their fitness orders.*

Step 2: *If the best candidate solution satisfies the convergence criterion*[1] *or if the number of iterations exceeds its limit I_{Ter}, then terminate; otherwise go to step 3.*

Step 3: *Select the best $p_s|\Delta_{l-1}| = |\eta_{l-1}|$ candidate solutions (individuals) from the current population Δ_{l-1}. This selection is accomplished according to the sorted candidate solutions.*

---

[1] A simple example of the convergence criterion would be to terminate the algorithm if there is no improvement of the fitness value in the iteration.

Step 4: *Estimate the probability distribution* $p(\theta_1,\theta_2,\cdots,\theta_n)$ *on the basis of* $|\eta_{l-1}|$ *best candidate solutions. We denote this estimation by*

$$\Gamma_{l-1} \equiv P(\theta_1,\theta_2,\cdots,\theta_n \mid \eta_{l-1}) \tag{12}$$

Step 5: *Generate new* $|\Delta_l|-|\eta_{l-1}|$ *individuals on the basis of this new estimated probability distribution* Γ_{l-1} *. Combine these newly generated* $|\Delta_l|-|\eta_{l-1}|$ *individuals with members of* η_{l-1} *to form a new population* Δ_l *.*
Step 6*: Go to step 1 and repeat the steps.*
We followed the steps of the above pseudo code for our EDA implementation for the MD-STBC-MIMO detecton problem. In our experimentation, for estimation (12), we used the simple scheme of estimating the marginal distributions separately and using the product form

$$\begin{aligned}\Gamma_{l-1} = p(\theta_1,\theta_2,\cdots,\theta_n \mid \eta_{l-1}) &= \prod_{i=1}^{n} p_i(\theta_i \mid \eta_{l-1}) \\ &= \prod_{i=1}^{n}\left(\frac{\sum_{j=1}^{|\eta_{l-1}|}\delta\left(x_i^j = \theta_i \mid \eta_{l-1}\right)}{|\eta_{l-1}|}\right)\end{aligned} \tag{13}$$

in order to generate the samples in the next iteration (generation), where δ is an indicator function for the individual indexed by j in the set η_{l-1}.

$$\delta\left(x_i^j = \theta \mid \eta_{l-1}\right) = \begin{cases} 1 & if\ x_i^j = \theta \\ 0 & otherwise \end{cases} \tag{14}$$

The use of a product-form distribution as in (13) is a part of our heuristic presented in this chapter. In fact, the statistics of the candidate solutions in η_{l-1} may show correlations among the variables $x_1, x_2, ..., x_n$. From these statistics, we could construct an empirical distribution (12) that captures correlations among variables, but that procedure would increase computational complexity. In the future, we will study to find how much performance improvement can be made by using such expensive procedures. Product-form distributions like (13) in EDA should not be discredited a priori because the benefit of searching variable correlations could, under particular circumstances, remain unclear (Platel et al.,2005).
A typical EDA can get stuck in a local optimum due to premature convergence of the probability distributions, or can be slowed down due to non-convergence of the probability distributions. We present a method of avoiding these two problems. Our approach is to apply a threshold on estimated parameters of the distributions. During the execution of a typical EDA, some of the estimated probabilities $P(\theta_i = 1 \mid \eta_{l-1}), ..., i = 1,2,...,$ or n, may become 0 or become very close to 0 at an early stage of the execution (at a small value of iteration count l). In that case, the algorithm is not likely to explore the candidate solutions with $x_i = 1$ during the rest of the execution. In order to thwart such premature convergence, we present an idea of adjusting the estimated probabilities $P(\theta_i = 1 \mid \eta_{l-1})$, i = 1, 2, .., n after

estimating these at each iteration. The adjustment in general can be regarded as a mapping from set of *n*-tuples

$$\Pi \equiv \left\{ \left(P(\theta_1 = 1 | \eta_{l-1}), P(\theta_2 = 1 | \eta_{l-1}), \dots, P(\theta_n = 1 | \eta_{l-1}) \right) \;\middle|\; 0 \le P(\theta_i = 1 | \eta_{l-1}) \le 1, i = 1, \dots, n \right\}$$

to set Π itself. First, we address the problem that a marginal probability value, in the estimated distribution, prematurely converges to 1. To avoid this, we set some thresholds $0.5 < \gamma_1, \gamma_2, \dots, \gamma_n < 1$. At any iteration, if the probability value $P(\theta_i = 1 | \eta_{l-1})$, *i*= 1, 2, …, n , is greater than γ_i, we set that value to γ_i −i.e., we set $P(\theta_i = 1 | \eta_{l-1}) = \gamma_i$. This way, some degree of randomness remains in the algorithm until the termination criterion is satisfied. A simpler application of this idea is to set the same threshold $\gamma = \gamma_1 = \gamma_2 = \dots = \gamma_n$. We can similarly address the problem that a probability value prematurely converges to 0. We define thresholds $0 < \tilde{\gamma}_1, \tilde{\gamma}_2, \dots, \tilde{\gamma}_n < 0.5$. At any iteration, if the estimated probability value $P(\theta_i = 1 | \eta_{l-1})$, i = 1, 2, .., n, is less than $\tilde{\gamma}_i$, we set $P(\theta_i = 1 | \eta_{l-1}) = \tilde{\gamma}_i$, so that some degree of randomness remains in the algorithm until the termination criterion is satisfied. A simpler application of this idea is to set the same threshold $\tilde{\gamma} = \tilde{\gamma}_1 = \tilde{\gamma}_2 = \dots = \tilde{\gamma}_n$.

5. Computational complexity

A motivation for applying the proposed near-optimal algorithms to a MD-STBC-MIMO problem is their low computational complexity. In this section, the computational complexity of BBO and EDA for MD-STBC-MIMO symbol detection is compared with that of MMSE, SD, SDR, V-BLAST, GA and the exhaustive search. The computational complexity of exhaustive search (an implementation of the ML detector) is $O\left(M^{N_S}\right)$ or $O\left(2^n\right)$, where $N_S = \sum_{k=1}^{K} Q_k$, $n \equiv N_S \log_2 M$, so an exhaustive search is usually impractical for real-time operations of symbol detection. A number of suboptimal detection schemes with better computational complexity have been presented in the literature.

The worst-case complexity of SD is exponential, and its expected complexity depends on the problem size and the SNR (Hassibi & Vikalo, 2005). The expected complexity of SD is roughly $O\left(N_S^3\right)$ at high SNRs (Hassibi & Vikalo, 2005) and $O\left(N_S^6\right)$ at low SNRs (Damen et al., 2000). MMSE is one of the sub-optimal detectors that involve inverting a matrix, and its computational complexity is $O\left(N_S^3\right)$ (Comaniciu et al.,2005). In V-BLAST, each iteration requires either a Zero-Forcing (ZF) or MMSE operation, and the number of iterations is equal to the total number of transmitted antennas. If the total number of transmit antennas in the system is equal to the number of receive antennas, then the complexity is $O\left(N_S^3\right)$. If the total number of transmit antennas in the system is greater than the number of receive antennas, then the complexity is $O\left(N_S^4\right)$ (Hassibi, 2000). The computational complexity of SDR (Kisialiou et al., 2005) is $O\left(N_S^{3.5}\right)$.

Typically, the time complexities of population-based algorithms are analyzed in terms of the number of function evaluations. The number of function evaluations in GA grows on the order of $O(gp)$, where g is the number of iterations (generations) and p is the population size. The number of function evaluations both in BBO and EDA also grows on the order of $O(gp)$. Now, let us consider the number of operations required for each function evaluation. Each function evaluation computes expression (6). Recall that the dimensions of Ω_k , χ_k , and y are $2Q_k \times 2TN_R$, $1 \times 2Q_k$, and $1 \times 2TN_R$, respectively. Computing the term $\sum_{k=1}^{K} \chi_k \Omega_k$ requires about $4TN_R\left(\sum_{k=1}^{K} Q_k\right) \approx 4N_s TN_R$ operations. Computing the norm square of the difference between two $2TN_R$-dimensional vectors as in $y - \sum_{k=1}^{K} \chi_k \Omega_k$ requires about $6TN_R$ arithmetic operations. Therefore, the total number of arithmetic operations to evaluate (6) is on the order of $4N_s TN_R + 6TN_R$. Combining the complexities of these procedures, we can say that the evaluation of functon (6) requires $O\left(N_s TN_R p\right)$ operations in each generation (iteration), where p is the population size. In each generation of GA, crossover and mutation procedures are performed, and these procedures take on the average $O\left(pN_s \log_2 M\right) = O\left(pN_s\right)$ computer operations, where $N_S \log_2 M$ is the length of the binary string that represents a chromosome[2]. As can be deduced from the pseudo code in Fig. 3, the average number of operations for migration and mutation in each generation (iteration) of BBO is also on the order of $O\left(pN_s \log_2 M\right)$. The average number of operations to estimate the probability distribution in each iteration of EDA is on the order of $O\left(N_s p \log_2 M\right)$. The sorting of p individuals (chromosomes) in GA, EDA and BBO can be performed in $O\left(p \log p\right)$. Therefore, the number of operations in each generation of GA, EDA and BBO can be expressed as $O\left(N_s TN_R p + N_s p + p \log p\right) \approx O\left(N_s TN_R p + p \log p\right)$.

6. Simulation results

In this section, we present the simulation results of the proposed BBO-based and EDA-based detection schemes and their comparison with other detection techniques applied to the MD-STBC-MIMO system. The system model used in our simulations is depicted in Fig. 1. The channels are assumed to be quasi-static, and different channels in the MD-STBC-MIMO system are assumed to be statistically independent. In all our simulations, we used the 4−QAM modulation. Each point in the plots of Figs. 4−10 is a value averaged over multiple independent simulation runs. In each simulation run, the set of transmitted signals $\left[\chi_1 \quad \chi_2 \quad \cdots \quad \chi_K\right]$, channel matrices $\left[\Omega_1 \quad \Omega_2 \quad \cdots \quad \Omega_K\right]^T$ and noise Z are generated randomly and independently. The main objective is to find the vector $\left[\chi_1 \quad \chi_2 \quad \cdots \quad \chi_K\right]$ that minimizes (6). This experimental setup enables us to compare different algorithms in terms of the performance averaged over different channel and noise realizations. Some BBO and EDA parameters are kept constant through all simulations, such as $I = 1$, $q = 0.1$, $p_s = 0.5$. As mentioned in section 4.1, for BBO we assumed constant emigration rate over all islands. In each simulation experiment, we set the BBO, EDA, and GA to have the same population size and number of iterations for fair comparison.

---

[2] In this discussion, we fix M, the constellation size of the modulation.

The simulation results in Figs. 4 through 7 show the BER performance comparison between MMSE, V-BLAST, SDR (Luo, 2010), SD (Hassibi & Vikalo, 2005), GA, EDA and BBO detectors. The MD-STBC-MIMO system configuration, (K,N_T,N_R,M), is set to (4,2,6,4), (5,2,8,4), (6,2,10,4) and (3,4,4,4,) for Figs. 4, 5, 6 and 7, respectively. The Figs. 4, 5, and 6 use orthogonal space time coding scheme (Alamouti, 1998). Fig. 7 uses a non-orthogonal space time coding scheme (Boariu et al., 2003). BBO, EDA and GA parameters, (g, p), which denote the number of iterations and the population size are set to (60,100), (100,100), (100,120) and (100,150) for Figs. 4, 5, 6 and 7, respectively. For these Figs., the total number, N_S, of symbols transmitted from all users are set 8, 10, 12, and 12, respectively. Thus, our BBO, EDA and GA experiments have search spaces of 4^8, 4^{10}, 4^{12} possible solutions. Figs. 4, 5, 6 and 7 indicate that in the MD-STBC-MIMO system, EDA and BBO have significantly better BER performance than MMSE, SDR, V-BLAST, and GA. BBO and EDA outperform other suboptimal detection methods in all four Figs. and can achieve performance close to the sphere decoding[3] detector. While BBO and EDA have performances close to that of the sphere decoding, they require much less computation. Table 2 shows the average number of operations required by SD, EDA and BBO for Figs. 4 to 7. From Table 2 we can observe that the average number of operations required by EDA or BBO is much smaller than the sphere decoder.

	Average number of operations			
	Fig. 4	Fig. 5	Fig. 6	Fig. 7
SD	16,777,216	64,000,000	19,110,2976	19,110,2976
EDA or BBO	20,4000	42,0000	60,0950	752640
$\frac{\text{Ave.Number of EDA or BBO operations}}{\text{Ave.Number of SD operations}}$	0.0120	0.0065	0.0031	0.0039

Table 2. Computation complexity comparison

Intuitively, for any population-based evolutionary algorithm, the larger population size and the larger number of iterations will produce the better results. However, the larger population size and larger number of iterations both result in the more computational load. Fig. 8 shows how the BER performance of BBO, EDA, and GA imporves as the number of iterations increases. The BER results of other schemes (SD, V-BLAST, SDR, ZF, MMSE) and the number of iterations (generations) in BBO, EDA, and GA are irrelevant, so we indicated the BER results of SD, V-BLAST, SDR, ZF, and MMSE as horizontal lines in Fig. 8. For this experimentation, we used MD-STBC-MIMO system configuration (K,N_T,N_R,M) = (4,2,6,4) and an Alamouti space time block code. We assumed quasi-static channels and fixed the SNR to 8 dB. Fig. 8 shows that the BER perfromance of BBO and EDA with the population size fixed to 100 approaches the BER perfromance of the sphere decoder in less than 100

---

[3] Due to heavy computational load of performing the Maximum Likelihood (ML) detection, which is optimal, it is difficult to compare performance with the ML detector. We used sphere decoding as the benchmark.

iterations while the GA performance is nowhere close to that of the sphere decoder even with 100 iterations.

Fig. 9 shows how the BER performance of BBO depends on the population size (p) and the number (g) of iterations. Thus, Fig. 9 exhibits the tradeoff between the population size and the number of iterations required to achieve a desired BER in BBO. The MD-STBC-MIMO system configuration was set $(K,N_T,N_R,M) = (4,2,4,4)$, and we used an Alamouti code. We assumed quasi-static channels and fixed the SNR to 8 dB. This tradeoff result is useful from the system design point of view. Increase in the population size and increase in the number of iterations each improves performance. While the increase in the population requires more memory module of the hardware, the increase in the number of iterations require fast processing in order to finish computation within a specified time. The tradeoff results such as Fig. 9 can guide how to set the population size and the number of iterations on the basis of hardware to be used. For example, if the hardware has high processing capabilities and small memory space, then we can set the population size low and run more iterations of BBO. Vice versa, if the hardware has large memory space and a slow processor, we can set the population size large and run a smaller number of iterations to get the same BER performance. We observe results similar to Fig. 9 for EDA.

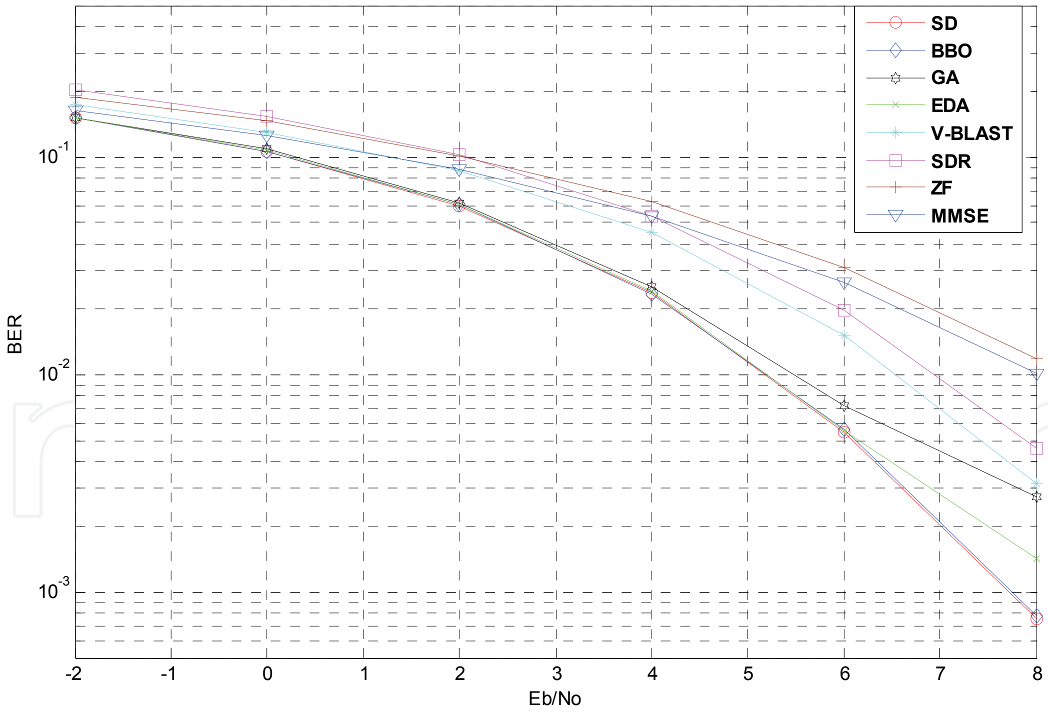

Fig. 4. BER performance comparison for $(K,N_T,N_R,M) = (4,2,6,4)$ over quasi-static fading channel. The (g,p) for GA, BBO and EDA is set to (60,100).

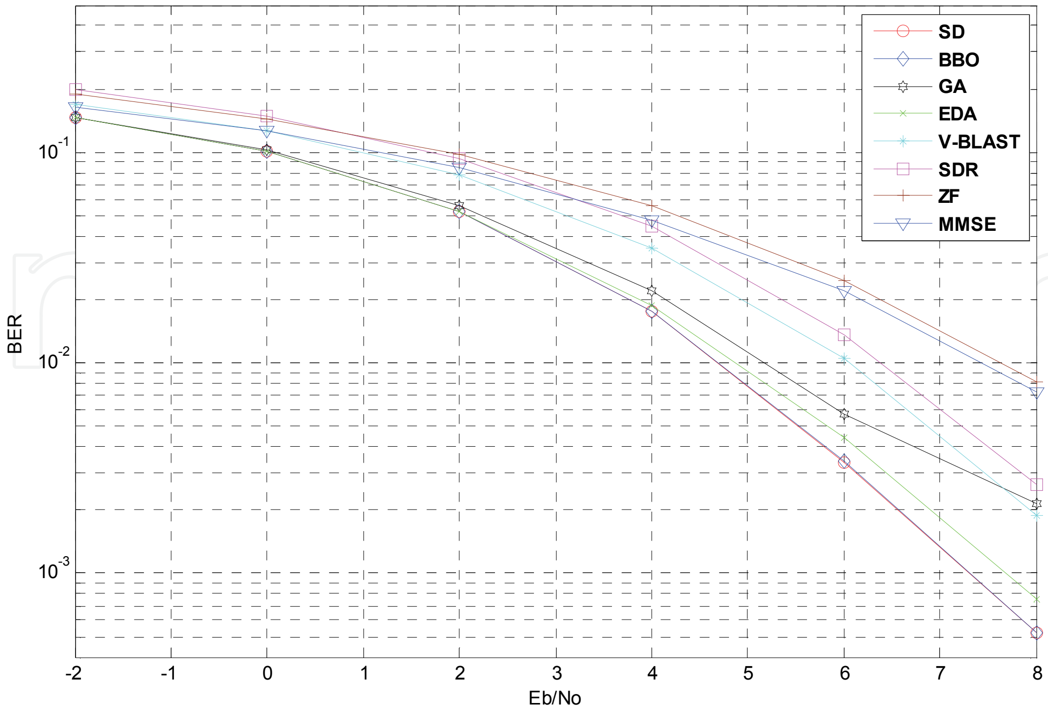

Fig. 5. BER performance comparison for (K,N_T,N_R,M) = (5,2,8,4) over quasi-static fading channel. The (g,p) for GA, BBO and EDA is set to (100,100).

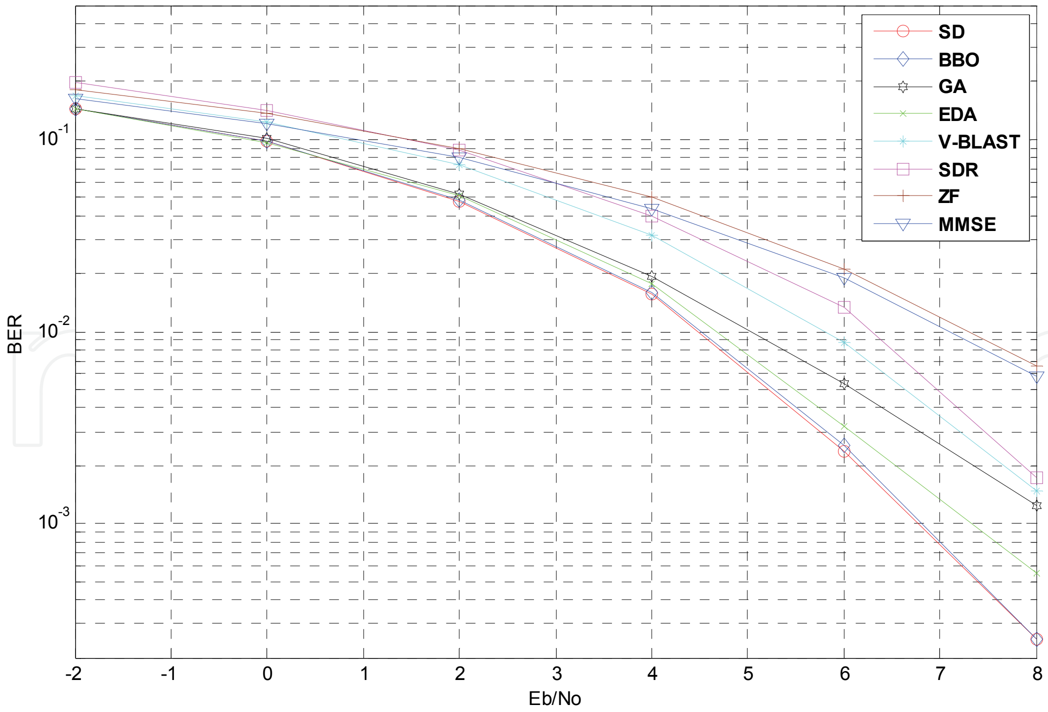

Fig. 6. BER performance comparison for (K,N_T,N_R,M) = (6,2,10,4) over quasi-static fading channel. The (g,p) for GA, BBO and EDA is set to (100,120).

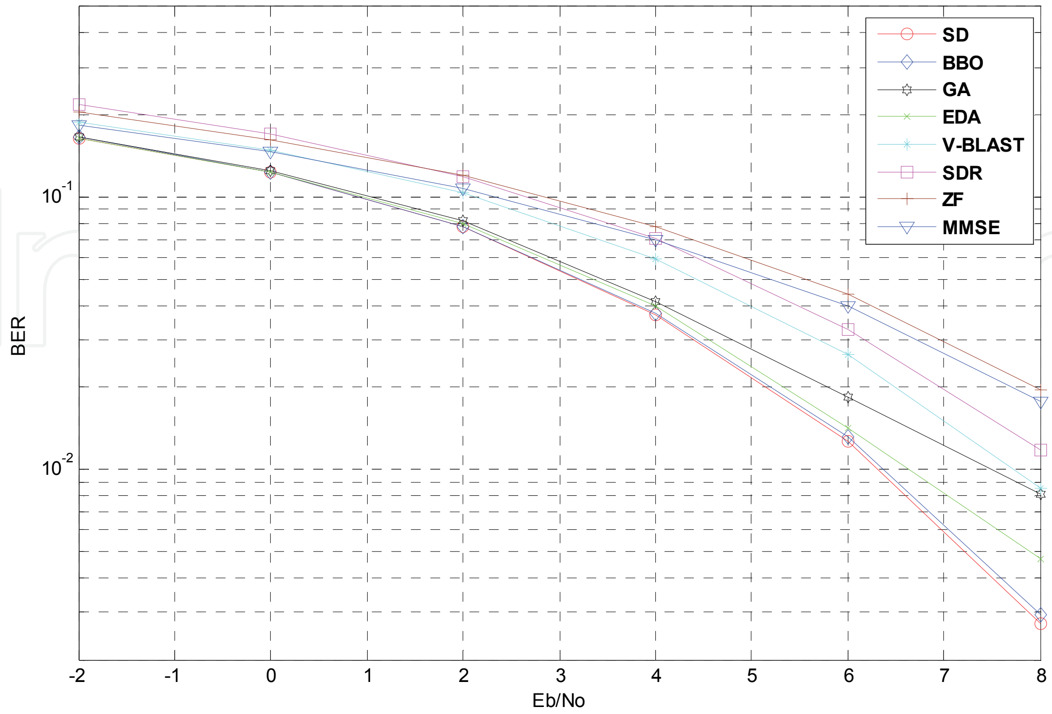

Fig. 7. BER performance comparison for (K,N_T,N_R,M) = (3,4,4,4) over quasi-static fading channel. The (g,p) for GA, BBO and EDA is set to (100,150).

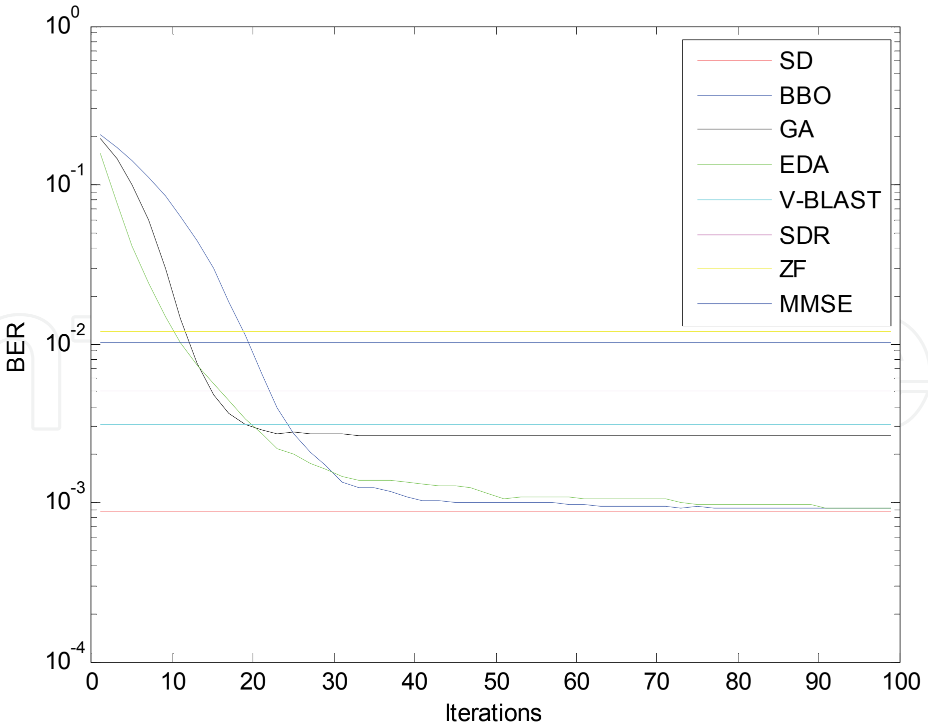

Fig. 8. How BER improves with the number of iterations in BBO, EDA, and GA. (K,N_T,N_R,M) = (4,2,6,4). The population size is fixed to 100.

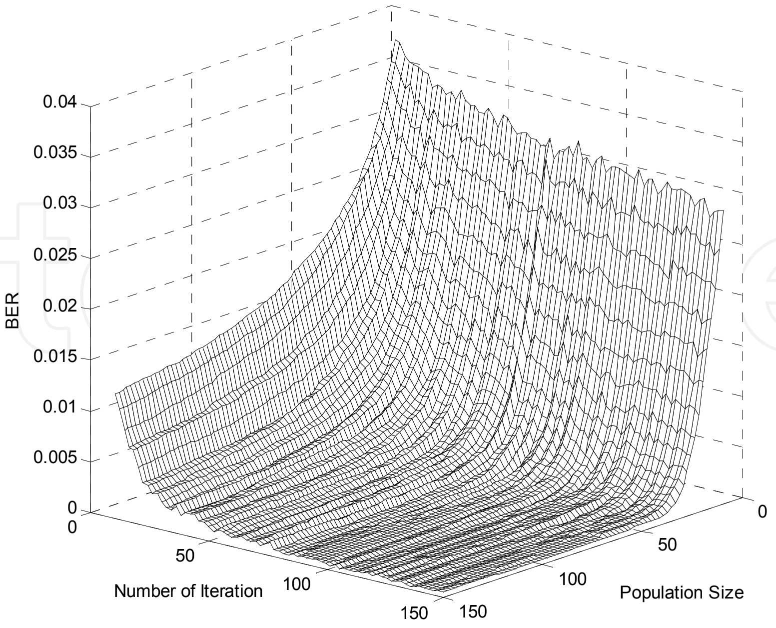

Fig. 9. BER dependency on the population size and the number of iterations in BBO. $(K, N_T, N_R, M) = (4,2,4,4)$

7. Conclusions

In this chapter, we proposed two population-based evolutionary algortihms, BBO and EDA, for symbol detection in the Multi-Device (MD) Space-Time Block Coded (STBC) Multi Input Multi Output (MIMO) Communication System. The proposed BBO and EDA algortihms have low complexity as compared with the sphere decoding algorithm, which is the best known algorithm for STBC decoding. Thus, the proposed BBO and EDA algorithms are suitable for high-speed real-time communications. In addition, compared to other Evolutionary Algorithms like GA, BBO and EDA algorithms show significantly better performance for the MD-STBC-MIMO detection. The proposed algorithms also show consistently better BER performance-complexity trade-offs in comparison to existing algorithms. Moreover, we believe that BBO and EDA for MD-STBC MIMO symbol detection still have room to improve further in terms of performance-complexity trade-offs. For example, we believe that BBO can be further improved by adjusting migration and mutation procedures.

8. References

Alamouti, S. M. (1998). A Simple Transmit Diversity Technique for Wireless Communications. *IEEE journal on select areas in communications*, Vol. 16, No. 8, pp. 1451-1458.

Boariu, A. and Ionescu, D.M. (2003) A class of nonorthogonal rate one space-time block codes with controlled interference. *IEEE Trans. Wireless Communication*, Vol. 2, No. 2 pp. 270-276.

Comaniciu, C.; Mandayam, N.B. & Poor, H.V. (2005) *Wireless Networks: Multiuser Detection in Cross-Layer Design*, Springer, NY.

Damen, O.; Chkeif, A. & Belfiore, J. C. (2000). Lattice code decoder for space-time codes. *IEEE Communications Lett*. Vol. 4, No. 5, pp. 161–163.

Foschini, G. J. & Gans, M. J. (1998). On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, Vol. 6, pp. 311-335.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.

Hassibi. B. (2000) An efficient square-root algorithm for BLAST. *Proceedings of IEEE ICASSP* pp. 737-740.

Hassibi, B. & Hochwald, B.M. (2002). High-rate codes that are linear in space and time. *IEEE Trans. Information Theory*, Vol.48, No.7, pp. 1804−1824.

Hassibi . B. & Vikalo, H. (2005). On the sphere decoding algorithm: Part I, the expected complexity. *IEEE Trans. Signal Processing*, Vol. 53, No. 8, pp. 2806-2818.

Jalden, J. & Ottersten, B. (2005). On the complexity of sphere decoding in digital Communications. *IEEE Trans. on Signal Processing*, Vol. 53, No 4, pp. 1474-14844.

Kisialiou, M. & Luo, Z.-Q. (2005). Performance analysis of quasi-maximum-likelihood detector based on semi-definite programming. *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*., Vol. 3, pp. 433-436.

Larrañaga, P. & A Lozano, J. (2001) *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers.

Luo, Z. -Q. (2010) http://www.ece.umn.edu/~luozq/software/sw_about.html.

Nekuii, M.; Kisialiou, M.; Davidson, T.N. & Luo, Z. Q. (2008). Efficient Soft Demodulation of MIMO QPSK via Semidefinite Relaxation. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2665-2668.

Platel, M.D.; Schliebs, S. & Kasabov, N. (2009) Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA. *IEEE Transactions on Evolutionary Computation,* Vol. 13, No. 6, pp. 1218−1232.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation* pp. 702-713.

Tarokh, V.; Seshadri N., & Calderbank A. R.(1998). Space-time codes for high data rate wireless communications: performance criterion and code construction. *IEEE Trans. Information Theory*, Vol. 44, pp. 744-765.

Telatar, E. (1999). Capacity of Multi-antenna Gaussian Channels. *European Trans. on Telecomm*. vol. 10, pp. 569-709, Nov. 1999.

Verdú, S. (1986). Minimum Probability of Error for Asynchronous Gaussian Multiple Access Channels. *IEEE Trans. Information Theory*, Vol. 32, pp. 85-96.

Verdú S. (1989). Computational Complexity of Optimum Multiuser Detection," Algorithmica, Vol.4, No.3, pp. 303-312.

Vikalo, H. & Hassibi, B.(2005).On the sphere decoding algorithm: Part II, generalizations, second-order statistics and applications to communications. *IEEE Trans. on Signal Processing*, Nol 53, No. 8, pp. 2819-2834.

Wolniansky, P.W.; Foschini, G.J.; Golden, G.D. & R.A. Valenzuela (1998). V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel. *Proc. ISSSE-98*, Pisa, Italy. pp. 87-91.

Wei, H. & Hanzo, L. (2004). Reduced-complexity near-optimum genetic algorithm assisted multiuser detection for synchronous multicarrier cdma. *Proceedings of IEEE 59th Vehicular Technology Conference*, Vol. 3, pp. 1717 – 1721.

MIMO Systems, Theory and Applications
Edited by Dr. Hossein Khaleghi Bizaki

ISBN 978-953-307-245-6
Hard cover, 488 pages
Publisher InTech

In recent years, it was realized that the MIMO communication systems seems to be inevitable in accelerated evolution of high data rates applications due to their potential to dramatically increase the spectral efficiency and simultaneously sending individual information to the corresponding users in wireless systems. This book, intends to provide highlights of the current research topics in the field of MIMO system, to offer a snapshot of the recent advances and major issues faced today by the researchers in the MIMO related areas. The book is written by specialists working in universities and research centers all over the world to cover the fundamental principles and main advanced topics on high data rates wireless communications systems over MIMO channels. Moreover, the book has the advantage of providing a collection of applications that are completely independent and self-contained; thus, the interested reader can choose any chapter and skip to another without losing continuity.

Daniel C. Lee and Muhammad Naeem (2011). Computationally Efficient Symbol Detection Schemes in Multi-Device STBC-MIMO Systems, MIMO Systems, Theory and Applications, Dr. Hossein Khaleghi Bizaki (Ed.), ISBN: 978-953-307-245-6, InTech, Available from: http://www.intechopen.com/books/mimo-systems-theory-and-applications/computationally-efficient-symbol-detection-schemes-in-multi-device-stbc-mimo-systems

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821