Hindawi Publishing Corporation
Advances in Artificial Intelligence
Volume 2013, Article ID 578710, 14 pages
http://dx.doi.org/10.1155/2013/578710

Research Article

Discrete Artificial Bee Colony for Computationally Efficient Symbol Detection in Multidevice STBC MIMO Systems

Saeed Ashrafinia, Muhammad Naeem, and Daniel Lee

School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6

Correspondence should be addressed to Saeed Ashrafinia; saa62@sfu.ca

Received 1 June 2012; Accepted 31 October 2012

Academic Editor: Jun He

A Discrete Artificial Bee Colony (DABC) is presented for joint symbol detection at the receiver in a multidevice Space-Time Block Code (STBC) Mutli-Input Multi-Output (MIMO) communication system. Exhaustive search (maximum likelihood detection) for finding an optimal detection has a computational complexity that increases exponentially with the number of mobile devices, transmit antennas per mobile device, and the number of bits per symbol. ABC is a new population-based, swarm-based Evolutionary Algorithms (EA) presented for multivariable numerical functions and has shown good performance compared to other mainstream EAs for problems in continuous domain. This algorithm simulates the intelligent foraging behavior of honeybee swarms. An enhanced discrete version of the ABC algorithm is presented and applied to the joint symbol detection problem to find a nearly optimal solution in real time. The results of multiple independent simulation runs indicate the effectiveness of DABC with other well-known algorithms previously proposed for joint symbol detection such as the near-optimal sphere decoding, minimum mean square error, zero forcing, and semidefinite relaxation, along with other EAs such as genetic algorithm, estimation of distributions algorithm, and the more novel biogeography-based optimization algorithm.

1. Introduction

Multi-Input Multi-Output (MIMO) communication systems can offer spatial diversity gains in the fading channels and have significantly higher channel capacity than the Single-Input Single-Output (SISO) systems for the same total transmission power and bandwidth [1, 2]. The system proposed in this paper comprises of one receiving station and multiple transmitting devices. The receiver's front end has multiple antennas, and each transmitting device has multiple transmit antennas. Employing the Space Time Block-Code (STBC) is realized to increase the capacity of MIMO systems and consequently improves data throughput and spectral efficiency [3]. Multiantenna systems are widely used because of their ability of dramatically increasing the channel capacity in fading channels [4]. Each transmit device uses an STBC; the receiver side performs the joint signal detection. Such a system is referred to as a multidevice (MD) STBC-MIMO system. Generally in an MD-STBC-MIMO system, the number of receive antennas is typically smaller than the cumulative number of transmit antennas used by all transmitting devices in the system. An example of MD-STBC-MIMO, with a smaller number of antennas at the base station or access point, would be the uplink multiple access communication in cellular systems.

This paper addresses the symbol detection in MD-STBC-MIMO systems. 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 and is further explained in Section 3. 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 M up to some value of M for the cases of small constellation sizes [5], but it grows exponentially for the cases of large constellation sizes. Also, for some SD algorithms, operation at a low SNR requires inordinately high computation; yet operation at a high SNR is efficient. In fact, [6] shows that even the expected computational complexity of the SD grows exponentially with the problem size in MIMO communication systems. In any case, an algorithm with polynomial growth of expected complexity for all values of the problem size M has not yet been found.

Due to the combinatorial nature of the problem, the ML detection is a choice to obtain the optimal solution; yet it has a high computational complexity. Taking advantage of heuristic algorithms, evolutionary algorithms (EAs) more specifically, and their ability to solve optimization problems efficiently facilitates finding an optimal solution with relatively low computational cost. evolutionary algorithms (EAs) are a subset of heuristic algorithms, which is inspired by biological evolution and mutation.

The Artificial Bee Colony (ABC) algorithm is one of the novel EAs that has been introduced by Karaboga in [7]. He presented the ABC algorithm for real (continuous) parameter optimization in unconstrained optimization problems in [8]. ABC is a population-based, stochastic global optimizer Evolutionary Algorithm. It is based on the theory of foraging bees searching for food sources for their nectar and sharing the information of food sources' locations to other bees in the hive. This algorithm demonstrates good accuracy and efficiency, compared with other mainstream EAs. In [8], Karaboga and Akay showed that ABC algorithm outperforms other EAs such as Differential Evolution (DE) [9], Particle Swarm Optimization (PSO) [10], and Genetic Algorithm (GA) [11] for numeric problems with multi-dimensions. In his recent paper [12], Karaboga compared the ABC programming with various genetic programming techniques and crossover methods available in [13] and demonstrated the superiority of ABC over these schemes through simulations. After introducing the ABC algorithm in 2007, some papers are published on the applications of ABC to different optimization problems [14–16], in addition to various real-world applications including filter design [17], image processing [18], control engineering [19], computer science [20, 21], neural networks [22], and even biology [23]. In these studies, ABC outperforms other EAs and is turned into a popular global optimization solver to the continuous optimization problems and applications.

The ABC algorithm discussed in [24] is primarily presented for continuous functions; yet the MD-STBC-MIMO detection problem is in the discrete domain. The impressive results of ABC implementation for continuous problems bring up the idea that a discretized versions of ABC can be a potential high-efficient low-complex solver for discrete or numerical optimization problems. In this paper we introduce a discrete ABC algorithm, in addition to some new features to enhance the overall algorithm's performance. Although there are a few discretized version of ABC in the literature, we will discuss in Section 4 that their efficiency is the result of their high complexity; thus they cannot be utilized for a real-time symbol detection problem such as MD-STBC-MIMO.

Our simulation results show that discrete ABC can meet the best known semi-optimal detector (i.e., SD) with less complexity and has better performance than other methods such as Minimum Mean Square Error (MMSE), Zero Forcing (ZF), and Semi-Definite Relaxation (SDR) [25], while it outperforms other EAs such as GA, Estimation of Distributions Algorithm (EDA) [26], and the recently proposed Biogeography-Based Optimization (BBO) [27].

In the rest of this paper, the system model is presented in Section 2. The application of existing symbol detection algorithms is discussed in Section 3. The discrete ABC algorithm and its application to the symbol detection problem are presented in Section 4. The computational complexity comparison of EAs and other solvers is discussed in Section 5. The simulation results are presented in Section 6, and Section 7 contains the conclusion and the future work.

2. System Model

The system consists of K mobile devices transmitting signals and one receiver. This system can model the uplink communication of the cellular system. Each mobile device has N_T transmit antennas that apply STBC, whereas the receiver front end has N_R receive antennas. The multiple mobile devices in the proposed systems can cause cochannel interference. An IQ-modulation scheme (e.g., M-QAM, M-PSK) maps source information into complex numbers. Even if each transmit device employs an orthogonal space-time code, the absence of coding across different wireless devices cannot guarantee the orthogonality among their signals. In the case of a single mobile device $K = 1$, the wireless device transmits using N_T transmit antennas and communicates with a receiver that has N_R antennas. The number of time slots in the space-time block code is denoted by T. The channel is assumed to be quasistatic; that is, the channel gain remains constant during each time block of data. It is also assumed that the channel gain at each time block is known to the receiver. This assumption is often used in the literature and reasonable if training or pilot signals are used. A complex $N_T \times N_R$ dimensional matrix H represents the MIMO channel, and another complex $T \times N_T$ dimensional matrix S represents the input signal in a space-time block code. The relationship between the input and output signals can be expressed as

$$\widetilde{Y} = SH + \widetilde{Z}, \tag{1}$$

where $\widetilde{Y}$ is a $T \times N_R$ dimensional complex output matrix and $\widetilde{Z}$ denotes the additive white noise matrix.

Equation (1) describes the relation between the input transmitted signals and the output received signals in terms of complex-valued matrix equation. The relation between input and output of the channel in a system with linear dispersion space-time coding can be equivalently expressed in terms of a real-valued matrix equation. We now briefly discuss the real-valued matrix equation. The input signal in (1) in the case of the linear dispersion code [28] is denoted by a complex-valued matrix S that takes the following form:

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

Here, $\alpha_q + j\beta_q$ ($q = 1, \ldots, Q$) is the complex number that represents the qth symbol, where α_q and β_q correspond to the real and imaginary parts of the symbol, respectively, and Q indicates the number of symbols conveyed in a space-time code block. In the IQ constellation diagram, α_q and β_q are discrete-valued variables, such that $\alpha_q + j\beta_q$ corresponds to a symbol in the constellation diagram. In 4-QAM, for example, each of these two variables can take values of ±1, and thus $\alpha + j\beta$ determines one of the four possible symbols arranged in the square grid of $(1, i)$, $(-1, i)$, $(-1, -i)$, and $(1, -i)$. These Q symbols $\alpha_q + j\beta_q$ can be represented as a $2Q$-dimensional real-valued row vector $\mathscr{X}$, whose components are constituted by $\alpha_q, \beta_q, q = 1, \ldots, Q$. The real and imaginary parts of matrix $\widetilde{Y}$'s components can be arranged as a $2TN_R$-dimensional real-valued row vector y. The relation between $\mathscr{X}$ and y in this new alternative form can be expressed as

$$y = \mathscr{X}\Gamma + Z, \tag{3}$$

where Z is a $2TN_R$-dimensional real-valued vector representing noise, and Γ is a $2Q \times 2TN_R$ real-valued matrix derived from the component of matrices $C_q, D_q, q = 1, \ldots, Q$, and H. Equation (1) in the case of multiple wireless devices can be expressed as

$$\widetilde{Y} = \sum_{k=1}^{K} S_k H_k + \widetilde{Z}, \tag{4}$$

where S_k is a $T \times N_T$-dimensional complex matrix of the input signal from wireless device k and the $N_T \times N_R$-dimensional complex matrix H_k represents the channel from the kth device to the receiver. As a result, (3) can be written as

$$y = [\mathscr{X}_1 \mathscr{X}_2 \cdots \mathscr{X}_K] \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_K \end{bmatrix} + Z, \tag{5}$$

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

3. Signal Detection

The ML detection is known to yield the lowest symbol error probability in the case of a priori equally likely symbols. In the case of our problem, 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 chooses transmitted symbols $[\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_K]$ that maximize $P(y \mid \mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_K)$. In the case of additive white Gaussian noise Z, the ML detection is reduced to choosing the vector $[\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_K]$ from M^{N_S} possibilities that has the shortest Euclidean distance $\hat{l}$, which is expressed as

$$\hat{l} = \left\| y - \sum_{k=1}^{K} \mathscr{X}_k \Gamma_k \right\|. \tag{6}$$

The ML detection scheme can be implemented by searching through all $M^{N_S} = 2^{bN_S}$ possible symbol sequences, where $b = \log_2 M$ and M is the size of the symbol constellation. Performing such an exhaustive search to find the minimum Euclidean distance in (6) is computationally inefficient, especially for large N_S. Computational complexity increases exponentially with N_S, the number of bits per symbol, transmit antennas per device, and the number of wireless devices K. High-speed communication requirements demand a low-complexity detection scheme. For low-complexity near-optimal detection, in this paper the ABC algorithm is applied 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. Section 4.2 describes how discrete ABC is applied to the signal detection. In the following sections, the performance of the discrete ABC-based detector is compared with other low-complexity suboptimal algorithms such as MMSE, ZF, SDR, and SD.

4. Discrete Artificial Bee Colony

This section presents the discrete Artificial Bee Colony (ABC) algorithm. A general description of ABC is given in the next subsection, followed by the modified discrete ABC algorithm in the subsequent section.

4.1. The Artificial Bee Colony Algorithm.

evolutionary algorithms (EAs) have been often used to solve difficult optimization problems. Most of the EAs are inspired by the theory of biological evolution (e.g., selection, crossover, mutation, recombination, and reproduction). The ABC algorithm has been recently presented by Karaboga for real (continuous) parameter optimization in unconstraint optimization problems [24], which is based on a particular intelligent behavior of the honey bee swarms. This algorithm demonstrates good accuracy and efficiency compared to other EAs such as differential evolution (DE) [29], ant colony optimization (ACO) [30], PSO, and GA, for numeric problems with multidimensions [8].

Consider an optimization problem

$$\begin{gathered} \max_x F(x) \\ \text{subject to:} \quad x \in C, \end{gathered} \tag{7}$$

where $x \equiv (x_1, x_2, \ldots, x_D)^t$ is a vector and C is a constraint set. In the original ABC, 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 ABC, each individual (candidate solution to an optimization problem) is analogically considered as a food source position. The fitness value, $F(x)$, of each individual (food source) x corresponds to the nectar quality of the food source.

This algorithm imitates the behaviors of the real bees on finding food source locations and sharing the information of food sources to the other bees in the nest. In this algorithm colony bees are classified into three types with certain responsibilities: employed bees, onlooker bees, and scout bees. Employed bees are the bees that have already been assigned to a food source. Each of them saves the food source position and selects another food source in her neighbor and chooses out of two the one that has a better nectar. Then they return to the hive and start to dance based on the quality of the nectar of their associated food source. An onlooker bee watches the dance of employed bees at the hive and selects an employed bee based on the dances observed so that the probability of choosing an employee bee is proportional to the nectar quality of that employee bee. Then the onlooker bee receives the information of the chosen employed bee associated with food source (the food source position and its nectar quality) from her and becomes an employed bee associated with that food source. Since then, the new employed bee (former onlooker) performs the same act as the employed bee in the previous phase; that is, she searches for a new food source in the neighbor of her associated food source for higher nectar quality and saves the best food source and its nectar to her memory. Finally, scout bees are free bees responsible for finding new food sources and evaluating their nectar. As soon as a scout bee finds a food source, she turns into an employed bee.

The algorithm assumes that there is only one employed bee for every food source; thus the number of employed bees is equal to the number of individuals in the population N. If there is no improvement in the nectar quality of a food source after certain number of trials, the food source will be abandoned and the employed bee assigned to that food source will become a scout bee that looks for a new food source. A pseudocode of the ABC algorithm is given in Pseudocode 1.

At the first step, ABC generates randomly distributed initial food source positions of the size N, whereas each individual solution x^i, $i \in (1, 2, \ldots, N)$ is a D-dimensional vector of numbers. In this step each scout bee that finds a food source location saves the current location in her memory and becomes an employed bee. In the employed bees phase, each employed bee finds a new food source position v^i in the neighborhood of her current associated food source x^i, and if the new food source has a better nectar, she saves the new position to her memory. In the original ABC algorithm, an employed bee locates the new food source positions using the following expression [24]:

$$v^i_j = x^i_j + \phi_{ij}\left(x^i_j - x^l_j\right). \qquad (8)$$

In this equation x^i_j is the jth component of the ith individual of the population, $l \in \{1, 2, \ldots, N\}$ is a randomly selected food source location (different from i), and $j \in \{1, 2, \ldots, D\}$ is a randomly chosen index. ϕ_{ij} is a random real number between $[-1, 1]$ that controls the production of a neighbor food source around x^i_j.

At the beginning of the onlooker bees phase, employed bees share their information about the quality of food sources with onlooker bees. An onlooker bee chooses an employed bee to take the food source information based on the following probability:

$$p_i = \frac{F\left(x^i\right)}{\sum_{i=1}^{N} F\left(x^i\right)}, \qquad (9)$$

where $F(x^i)$ is the fitness value of the ith solution in the population x^i. After an onlooker bee selects a food source, she becomes an employed bee and locates a new food source in the neighborhood using (8), then she compares its nectar quality with the current food source, and saves the food source position that has a better nectar quality to her memory and returns back to the hive to share this information.

If the number of trials t^i of a food source x^i is not improved through a predetermined number of trials t, it will be removed from the population, and the employed bee assigned to that bee becomes a scout that searches for a new food source. Each component of the new food source is randomly selected from $[x_{\min}, x_{\max}]$, where $x_{\min}$ and $x_{\max}$ are the minimum and maximum of the allowable values in the problem domain. The previous steps are repeated until the termination condition is satisfied, which here is a preset maximum number of generations G.

4.2. Discrete ABC Algorithm for Joint Symbol Detection. This subsection introduces a discrete version of the ABC algorithm. The ABC algorithm discussed in the previous subsection is for optimization problems in the continuous domain. However, some of the previous steps have to be modified because the decision variables, which in this case are the transmitted symbols of the MD-STBC-MIMO problem, are a set of nonnegative integer numbers. In the discrete ABC (DABC) algorithm, we define a new expression to search in the neighborhood of the current food source position as a replacement to expression (8):

$$v^i_j = \text{randint}\left(x^l_j, 2x^i_j - x^l_j\right), \qquad (10)$$

where "randint (a, b)" returns a random integer number between a and b (more specifically a random integer number starting from $\min\{a, b\}$ to $\max\{a, b\}$). Note that if the result of (10) falls beyond the problem's integer domain, that number is replaced with the closest integer defined within the boundaries. This expression is used in both employed bees and onlooker bees phases. Moreover, the onlooker bees select employed bees with the following probability:

$$p_i = \sqrt{\frac{F\left(x^i\right)}{\max_{i=1,\ldots,N} F\left(x^i\right)}}, \qquad (11)$$

which has been observed to increase the efficiency of the algorithm more than other selection methods and the one in (9). The evaluation of (10) is explained in the appendix.

We have applied more enhancements to the algorithm in order to reduce its computational complexity. The most

```
Send scouts (generate initial population)
Repeat
    Employed bees phase
    Onlooker bees phase
    Scout bee phase
    Memorize the best food source found so far
Until termination condition satisfied
```

Pseudocode 1: Pseudocode of the general ABC algorithm.

```
Initialize the population of solutions $x^i$, $i = 1, 2, \ldots, N$,
Evaluate $F(x^i)$, $\forall i$,
repeat
    Run the DABC employed bee phase, (Pseudocode 3)
    Run the DABC onlooker bee phase, (Pseudocode 4)
    Run the DABC scout bee phase, (Pseudocode 5)
    Save the best results,
until termination condition satisfied,
```

Pseudocode 2: The DABC algorithm pseudocode.

complex section of ABC, and most other EAs, is the fitness function evaluation. Therefore, if the number of these evaluation decreases, the algorithm runs faster. In this version of the discrete ABC, during the employed bees and onlookers bees phases, the algorithm is set to only evaluate those individuals that are modified during the greedy selection process. When the algorithm uses (10) to select a neighbor food source, it may not always return a new food source position due to the stochastic nature of (10). The DABC is set to check whether a food source has been modified prior to proceeding to the fitness function evaluation. In this case, it eliminates a number of fitness evaluations for some individuals that were already evaluated during the previous generations.

The last phase of the DABC is similar to the scout bee phase of the original ABC. In the scout bee phase of DABC, the algorithm selects only one food source x^i that exceeds maximum allowable number of trials t^i to abandon and sends one scout bee to explore new food source positions. The scout bee randomly selects one food source, evaluates its nectar quality, and saves it in her memory. This procedure helps the algorithm to explore the search space more effectively, which is an advanced version of the mutation process in GA and some other EAs that they randomly mutate any individual of the population. A detailed pseudocode for the discrete ABC algorithm is given in Pseudocodes 2, 3, 4, and 5.

We found a few articles in the literature that employ ABC for numerical optimization problem. In [31], Tsai et al. presented an enhanced ABC, which is applied to numerical optimization problems, and called it the interactive ABC (IABC). They have modified the way in which onlooker bees choose a neighboring food source position. IABC incorporates the concept of universal gravitation into the consideration of the affection between employed bees and onlooker bees, and their simulation results demonstrate the high performance of IABC compared with ABC and PSO. Note that the higher performance of their new method comes with the price of a significant increase in the complexity of the algorithm—there is heavy computational load for calculating the gravitation between n employed bees in (10) and (11) in [31] for every food source in every algorithm iteration. As mentioned before, an important aspect of EAs' implementation for wireless communication problems, specifically the MD-STBC-MIMO detection, is to have high performance results while maintaining low complexity.

There are other articles that apply ABC to optimization problems with a binary domain. Wang et al. have presented a binary selection method to the ABC algorithm in [32]. Their binary encoding method employs a sigmoid function of velocity as a logical choice for binary selection. However, because they are incorporating (8) with a sigmoid function, their method has a higher complexity than (10) presented in the present paper. Salim et al. introduce a discrete bee algorithm for numerical optimization. However, their algorithm includes binary operators ((8) and (11)–(13) in [33]). They show that their algorithm has better performance than ABC; but if this algorithm is applied to a numerical optimization problem, integer-to-binary and binary-to-integer built-in functions are required to convert all of the integer individuals of each population to binary for food source exploration, and they all have to be converted back into integer for fitness function evaluation after population modification. This procedure has to be done at least two times (during employed bee and onlooker bee phases) for all individuals in all generations, which dramatically increases the algorithm complexity. The algorithm in the present paper has two advantages over the aforementioned papers: it can be implemented for numerical optimization problems, including problems with a binary domain, and it clearly has less complexity

```
$t^i = 0, \forall i$
for each food source $x^i$, $i = 1, 2, \ldots, N$,
    Select a random food source $l$, $l \neq i \in \{1, 2, \ldots, N\}$,
    Select a random component $j$, $j \in \{1, 2, \ldots, D\}$,
    $v^i_j = \text{randint}\left(x^l_j,\ 2x^i_j - x^l_j\right)$,
    $t^i = t^i + 1$,
    if $v^i \neq x^i$ then,
        Evaluate $F\left(v^i\right)$,
        $x^i \leftarrow v^i$,
        $t^i = 0$,
    end if,
end for,
```

Pseudocode 3: The DABC employed bee phase pseudocode.

```
Calculate probability values $p_i$ for $x^i, \forall i$ using (11),
$w = 1; i = 1;$
for $w = 1, \ldots, N$, %($w$ corresponds to the $w$th onlooker bee)%
    if rand > $p_i$ then, %(select the $i$th employed bee to follow)%
        Select a random food source $l$, $l \neq i \in \{1, 2, \ldots, N\}$,
        Select a random component $j$, $j \in \{1, 2, \ldots, D\}$,
        $v^i_j = \text{randint}\left(x^l_j, 2x^i_j - x^l_j\right)$,
        $t^i = t^i + 1$,
        if $v^i \neq x^i$ then,
            Evaluate $F\left(v^i\right)$,
            $x^i \leftarrow v^i$,
            $t^i = 0$,
        end if,
        $w = w + 1$,
    end if,
    $i = i + 1$;
    if $i > N$ then $i = 1$; %(reset $i$)%
end for,
```

Pseudocode 4: The DABC onlooker bee phase pseudocode.

compared with algorithms presented in other papers. DABC's superior performance compared with other mainstream EAs is demonstrated through simulation results in Section 6.

4.3. Application of DABC to MD-STBC-MIMO Joint Symbol Detection. The aim of applying DABC to the MD-STBC-MIMO symbol detection problem is to minimize the Euclidian distance defined in (6). Therefore, the Euclidian distance in (6) represents the fitness function or nectar quality, and shorter Euclidian distance means better fitness. An individual of the discrete ABC algorithm corresponds to a possible solution to the joint symbol detection problem, that is, a set of conveyed symbols from the K transmit devices.

In the MD-STBC-MIMO system discussed in this paper, transmitted symbols are chosen from an IQ modulation such as M-QAM or M-PSK constellation diagram. We represent each of the M possible points in the constellation by a unique integer in the set $\{0, \ldots, M-1\}$. The system comprises K transmit devices, each device indexed by k transmitting Q_k M-QAM symbols in a space-time code block. Therefore, a DABC individual x (a food source location) can be defined as a $N_S = \sum_{k=1}^{K} Q_k$-dimensional ($D = N_S$) integer row vector $x \triangleq [x_1, x_2, \ldots, x_{N_S}]$ where $x_v \in \{0, \ldots, M-1\}$, $v \in \{1, \ldots, N_S\}$. For DABC, the integer vector x represents the vector $[\mathcal{X}_1 \mathcal{X}_2 \cdots \mathcal{X}_K]$ in expression (6), and the fitness function is translated accordingly. Consequently, a scout bee generates a random vector of D integer numbers in $\{0, \ldots, M-1\}$ as a new individual (new food source location) in line 1 of Pseudocode 2 and line 3 of Pseudocode 5; that is, $x_{\min} = 0$, $x_{\max} = M - 1$.

5. Computational Complexity

A motivation for applying the proposed near-optimal algorithms to an MD-STBC-MIMO problem is their low computational complexity. In this section, the computational complexity of DABC for MD-STBC-MIMO symbol detection is compared with that of ZF, MMSE, SD, SDR, EDA, BBO, and GA. The computational complexity of the exhaustive search (an implementation of the ML detector) is $O(M^{N_S})$,

```
if there exists some $x^i \mid \{t^i > t\}$,
    Select one such $x^i$ randomly,
    for each component $j$, $j \in \{1, 2, \ldots, D\}$,
        $v_j^i = \text{randint}[x_{\min}, x_{\max}]$,
    end for,
    Evaluate $F(v^i)$,
    $x^i \leftarrow v^i$,
    $t^i = 0$,
end if,
```

Pseudocode 5: The DABC scout bee phase pseudocode.

so 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 SNR [5]. SD has high complexity of $O(\tilde{n}^6)$ [34] at low SNRs, where $\tilde{n} = n \log_2 M$. However, it has polynomial complexity, often roughly cubic complexity, at high SNRs [5]. MMSE is one of the suboptimal detectors that involves inverting a matrix, and its computational complexity is $O(\tilde{n}^3)$ [35]. The computational complexity of SDR [36] in each iteration is $O({N_T}^{3.5})$ where N_T stands for the number of transmit antennas.

Typically, the computational complexity of population-based algorithms is analyzed in terms of the number of fitness function evaluations, which in this paper would be (6). One important reason is that their complexity is highly dependent on their implementation and coding efficiency. The number of function evaluations in, BBO, GA, and EDA, is the same and equal to GN, where G and N represent the total number of generations and the population size, respectively [9]. The reason is that in all these algorithms, every individual is evaluated just once during one generation.

In the ABC algorithm however, there is more than one fitness function evaluation for each individual during a generation. During one generation in ABC, during the employed bees phase, each employed bee tests a neighbor food source for its quality; thus the fitness function evaluation has to be run for the whole N food source positions once. By the same token, during the onlooker bees phase there are N fitness function evaluations for every food source. So the overall number of fitness function evaluations for the algorithm for these two phases would be $2GN$. In the scout bees phase, the algorithm selects only one individual that exceeds t trials to abandon and replaces its employed bee with a scout. Hence the first individual to exceed t trials would be in the $(t/2)$th algorithm generation. After that, in order to determine a feasible number of function evaluations the worst case has to be considered, in which after the $(t/2)$th generation there is a maximum of one abandoned food source in every generation. As a result, the total number of fitness function evaluations for ABC would be as follows:

$$2GN + \left(G - \frac{t}{2}\right). \tag{12}$$

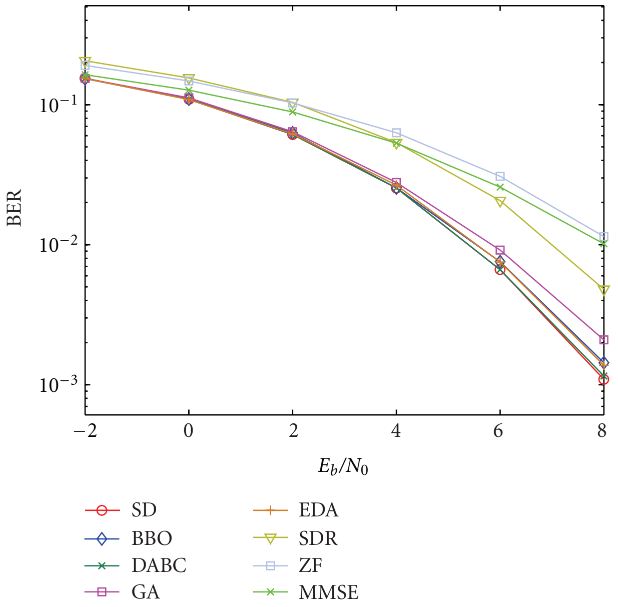
This complexity is higher than the complexity of other aforementioned EAs. However, the complexity of DABC presented in this paper is yet less than (12) because this algorithm does not run the function evaluation procedure for all the individuals in the employed bees and onlooker bees phases due to the conditions in line 7 of Pseudocode 3 and line 9 in Pseudocode 4. This stochastic behavior prevents DABC to have a closed-form number of fitness function evaluations.

6. Simulation Results

This section contains the simulation results of the proposed DABC-based detection and its comparison with other detection techniques applied to an MD-STBC-MIMO system. The channels are assumed to be quasistatic, and different channels in MD-STBC-MIMO are assumed to be independent. In all our simulations, it is assumed that the mobile data is transmitted in a form of 4-QAM modulation from all wireless devices ($M = 4$). For simulation experiments we assumed that each of the K devices transmits the same number of symbols $Q_k = Q$. Therefore, there are $N_S = KQ$ symbols conveyed from the K transmit devices to the receiver. Each point in the plots of Figures 1, 2, 3, 4, 5, 6, 7, and 8 is a value averaged over multiple independent runs. During each simulation runs, the set of symbols transmitted in a space-time code block are generated randomly and independently of other simulation run. Also the noise term is generated randomly and independently of other simulation trials.

In order to present a fair comparison between the EAs, they are tuned to their best performance, and they are sharing the same number of generations, population size, and initial population. Other EA parameters are kept constant during all simulation runs. A list of parameters set for each algorithm is presented in a table for each simulation result next to its figure. Moreover, GA employs a greedy selection scheme [11], and BBO uses a low complex emigration-based migration scheme with a constant emigration curve and linear immigration curve [9].

Each point in the plots of Figures 1–8 is a value averaged over multiple independent runs. In each simulation run, the set of transmitted symbols ($[\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_K]$ in (5)), channel matrices ($[\Gamma_1, \Gamma_2, \ldots, \Gamma_K]^T$ in (5)), and noise (Z in (5)) are generated randomly and independently of other trials. Hence, in each simulation run the received signal y in (6) is set from those randomly generated variables in accordance



(a)

System parameters								
K	N_T	N_R	M	T	Search space	STBC type	Channel type	No. of simulation runs
4	2	6	4	2	4^8	Alamouti	Quasi-static fading	2000

Shared EAs		BBO			DABC	GA			EDA		
Generation	Pop	I	m	Migration	t	P_{xover}	P_{mut}	P_{sel}	P_{xover}	P_{mut}	P_{sel}
60	60	1	0.015	Constant	$0.4 \times pop$	0.9	0.5	0.5	0.99	0.95	0.5

(b)

	−2	0	2	4	6	8
ZF	89	86	80	74	70	64
MMSE	96	92	86	78	74	66
SDR	84	84	80	78	78	77
GA	100	99	96	91	85	83
EDA	100	100	98	95	93	88
BBO	100	100	98	97	96	95
DABC	100	100	100	100	100	100

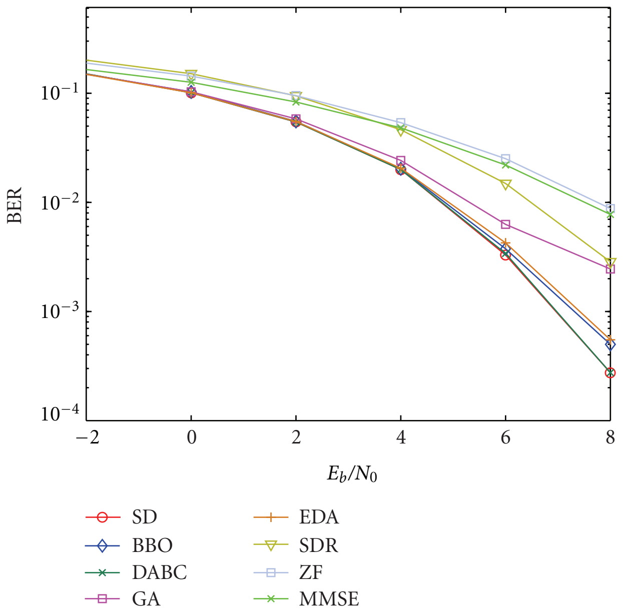
(c)

Figure 1: (a) BER performance comparison of $(K, N_T, N_R, M) = (4, 2, 6, 4)$, (b) simulation parameters, and (c) percentage of decoders with the SD results.

with (5). Then the algorithms are run to search for the value of the integer vector x (which represents $[\mathscr{X}_1, \mathscr{X}_2, \dots, \mathscr{X}_K]$ as mentioned in Section 4.3) that minimizes $\hat{l}$. Therefore, the results averaged over different simulation runs are in fact averaged over the different channel and noise realizations and also different realizations of the algorithm's evolution in the case of probabilistic algorithms such as GA, DABC, EDA, and BBO. This experimental setup compares different algorithms in terms of the averaged performance over different channel and noise realizations.

The simulation results in Figures 1 through 8 show the BER performance comparison between ZF, MMSE, SDR, SD, GA, EDA, BBO, and DABC detectors. The MD-STBC-MIMO system configuration, (K, N_T, N_R, M, T), is set to $(4, 2, 6, 4, 2)$, $(5, 2, 8, 4, 2)$, $(6, 2, 10, 4, 2)$, and $(3, 4, 4, 4, 2)$ for Figures 1, 2, 3, and 4, respectively. The Alamouti space-time coding [24] is used in Figures 1, 2, and 3; but for Figure 4, a nonorthogonal four transmit antennas configuration is used for each mobile device.

EA's shared parameters (G, N), are set to $(60, 60)$, $(100, 100)$, $(100, 120)$, and $(120, 200)$ in Figures 1, 2, 3, and 4, respectively. For these figures, the total number N_S of symbols transmitted from all users is set to 8, 10, 12, and 14; the algorithms are searching through a search space of $4^8, 4^{10}, 4^{12}$, and 4^{12} possible solutions, respectively. For each simulation run, the pair of (G, N) is selected not only to make the EAs' results close to SD's, but to choose the smallest possible G and N to reduce their computational complexity.



(a)

System parameters

K	N_T	N_R	M	T	Search space	STBC type	Channel type	No. of simulation runs
5	2	8	4	2	4^{10}	Alamouti	Quasi-static fading	2000

Shared EAs			BBO		DABC	GA			EDA		
Generation	Pop	I	m	Migration	t	P_{xover}	P_{mut}	P_{sel}	P_{xover}	P_{mut}	P_{sel}
100	100	1	0.015	Constant	$0.4 \times pop$	0.9	0.5	0.5	0.99	0.95	0.5

(b)

	−2	0	2	4	6	8
ZF	88	85	81	75	64	58
MMSE	95	90	85	78	67	59
SDR	85	82	81	79	74	71
GA	100	99	98	95	89	73
EDA	100	100	100	99	96	92
BBO	100	99	100	100	98	93
DABC	100	100	100	100	99	100

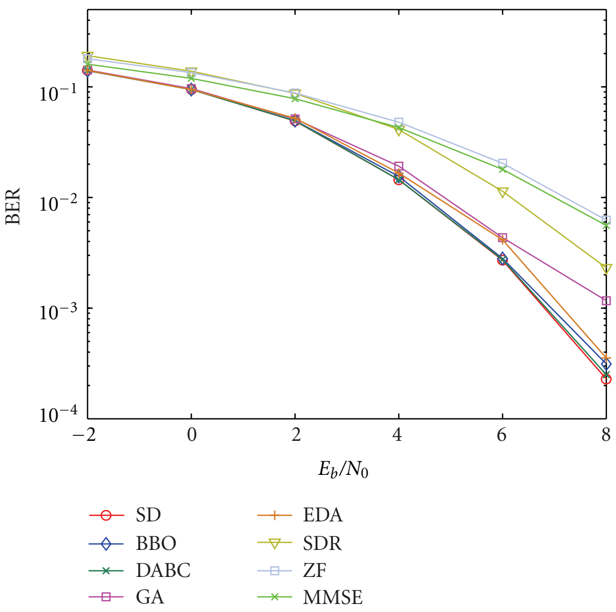
(c)

FIGURE 2: (a) BER performance comparison of $(K, N_T, N_R, M) = (5, 2, 8, 4)$, (b) simulation parameters, and (c) percentage of decoders with the SD results.

In these figures, three EAs (DABC, BBO, and EDA) return the closest results to the SDs. In most of the cases, DABC and SD exactly match together and seem as a united line. Observing these figures shows that the best algorithm that almost always returns the same result as the SD is DABC, followed by the BBO decoder that returns results with about 95% of SD. The third place is for EDA, followed by GA. All EAs outperform other suboptimal detection methods in all the five figures and can meet the optimal result by searching through a much smaller set of individuals by selecting a decent pair of (G, N).

Figures 1, 2, and 3 indicate that in the MD-STBC-MIMO, DABC has significant better BER performance than ZF, MMSE, SDR, GA, EDA, and BBO, while it closely matches SD. In Figure 3, for example, although DABC and other EAs are searching over approximately $GN = 12,000$ individuals over the entire search space of $4^{12} \cong 1.7 \times 10^8$ possible solutions, DABC's BER performance meets the near-optimal SD. According to this figure, DABC matches SD, where BBO, EDA, and GA require about 0.1 dB, 0.5 dB, and 0.6 dB more SNRs than BBO to achieve BER of 10^{-2}, respectively.

The experiment in Figure 4 was performed on a nonorthogonal space-time code, whereas the experiments in the other figures were performed on the Alamouti code (simple and orthogonal). The total number of symbols N_S transmitted from all users in a space time code block is 12. Hence,



(a)

System parameters								
K	N_T	N_R	M	T	Search space	STBC type	Channel type	No. of simulation runs
6	2	10	4	2	4^{12}	Alamouti	Quasi-static fading	2000

Shared EAs			BBO		DABC	GA			EDA		
Generation	Pop	I	m	Migration	t	P_{xover}	P_{mut}	P_{sel}	P_{xover}	P_{mut}	P_{sel}
100	120	1	0.015	Constant	0.4 × *pop*	0.9	0.5	0.5	0.99	0.95	0.5

(b)

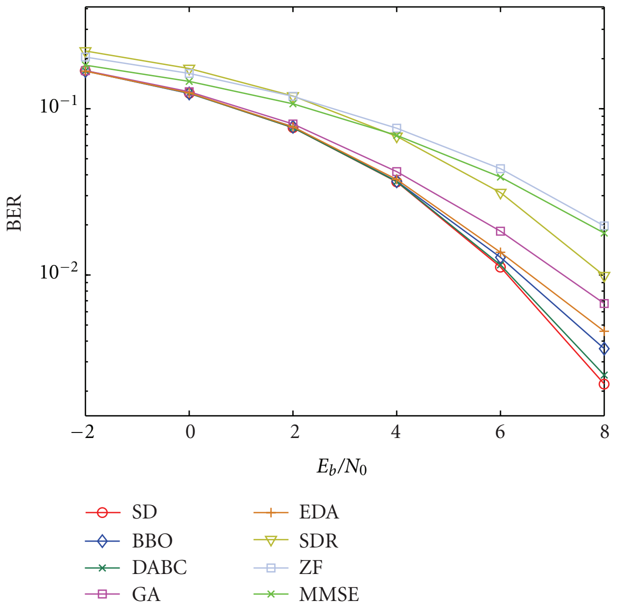
	−2	0	2	4	6	8
ZF	87	85	81	72	66	61
MMSE	93	90	85	74	68	62
SDR	84	84	81	75	76	72
GA	99	99	98	93	92	81
EDA	100	100	98	97	93	95
BBO	100	100	100	98	99	96
DABC	100	100	100	100	100	99

(c)

Figure 3: (a) BER performance comparison of $(K, N_T, N_R, M) = (6, 2, 10, 4)$, (b) simulation parameters, and (c) percentage of decoders with the SD results.

using 4-QAM, the size of the search space would be 4^{12}, while EAs only evaluate $GN = 24,000$ points in the search space, which is significantly smaller than the search space ML has to cover. Similar to other figures, SD and DABC have the best BER performance. In higher SNRs, GA's performance diminishes notably, while DABC pursues the near-optimal SD. It can be observed from the figure that BBO, EDA, and GA require 0.3, 0.4, and 1.5 dB less SNR than SD and DABC to achieve BER of 10^{-2}, respectively. DABC perceptibly behaves as the best detection algorithm among other suboptimal detection methods in all the four figures.

From the computational complexity point of view in EAs, finding the optimal pair of (G, N) is essential in order to minimize the processing power and the required memory. According to the computational complexity order of these algorithms, with a fixed population size N, more iterations until termination means more computation. Figure 5 shows the number of iterations required by each detection scheme to achieve a desirable BER. The MIMO system configuration is $(K, N_T, N_R, M) = (6, 2, 10, 4)$, it uses the Alamouti STBC and quasistatic channel, and the SNR is fixed to 8 dB. Figure 5 shows that the discrete ABC algorithm with the population size fixed to 100 is the first algorithm that can reach SD's performance in less than 65 iterations. Other EAs whether cannot reach the SD results, or require much more iterations to converge to the SD results. This improved performance



(a)

						System parameters						
K	N_T	N_R	M	T	Search space	STBC type		Channel type			No. of simulation runs	
3	4	4	4	2	4^{12}	Alamouti		Quasi-static fading			2000	
	Shared EAs			BBO		DABC		GA			EDA	
	Generation	Pop	I	m	Migration	t	P_{xover}	P_{mut}	P_{sel}	P_{xover}	P_{mut}	P_{sel}
	120	200	1	0.015	Constant	0.4 × pop	0.9	0.5	0.5	0.99	0.95	0.5

(b)

	−2	0	2	4	6	8
ZF	88	85	81	75	64	58
MMSE	95	90	85	78	67	59
SDR	85	82	81	79	74	71
GA	100	99	98	95	89	73
EDA	100	100	100	99	96	92
BBO	100	99	100	100	98	93
DABC	100	100	100	100	99	100

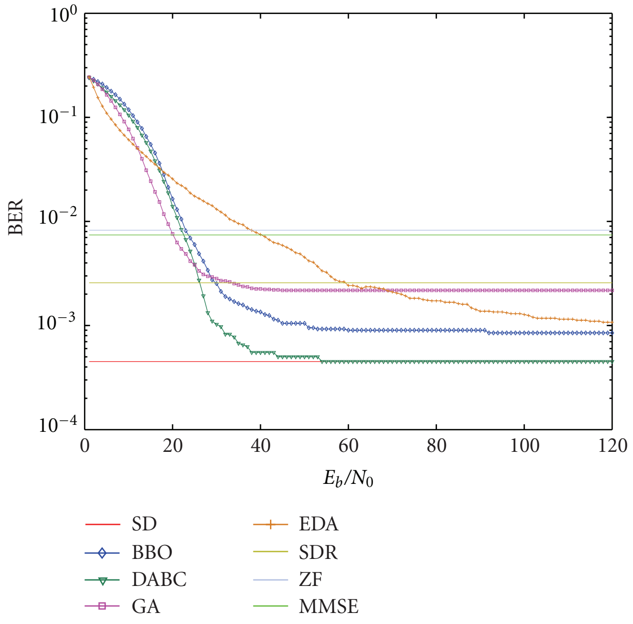
(c)

Figure 4: (a) BER performance comparison of $(K, N_T, N_R, M) = (3, 4, 4, 4)$, (b) simulation parameters, and (c) percentage of decoders with the SD results.

is consistently observed in several other simulations with different system configurations. As a result, not only DABC algorithm outperforms other suboptimal algorithms, it delivers better results than other well-known EAs, such as GA, BBO, and EDA, and can reach SD.

Figures 6 to 8 show the trade-off between the population size and the iterations required to achieve a desired BER in GA, BBO, and DABC. The MIMO system configuration is $(K, N_T, N_R, M) = (4, 2, 4, 4)$, using the Alamouti STBC and quasistatic channel, and the SNR is set to 8 dB. The detailed system configuration is given in Table 1. This trade-off is useful from the system design point of view. If a hardware system has high processing capabilities and low memory, then the population size can set lower to get same BER performance and vice versa. (Higher N and G need more memory.)

Finally, we compare the number of fitness function evaluations between EAs to demonstrate the superiority of DABC over other EAs. As mentioned in Section 5, BBO, EDA, and GA all have the same number of fitness function evaluations equal to GN. For the parameters of a system described in Figure 2 for instance, these EAs require $GN = 10,000$ fitness function evaluations. The original ABC requires $2GN + (G - t/2) = 20,080$, while DABC requires an average of 13,600 evaluations per independent simulation run, which is comparable to other EAs, less than the original ABC and



(a)

System parameters												
K	N_T	N_R	M	T	Search space	STBC type	Channel type			No. of simulation runs		
6	2	10	4	2	4^{12}	Alamouti	Quasi-static fading			2000		
Shared EAs				BBO		DABC		GA			EDA	
Generation		Pop	I	m	Migration	t	P_{xover}	P_{mut}	P_{sel}	P_{xover}	P_{mut}	P_{sel}
1 ~ 150		100	1	0.015	Constant	$0.4 \times pop$	0.9	0.5	0.5	0.99	0.95	0.5

(b)

FIGURE 5: (a) BER performance comparison versus iterations for $(K, N_T, N_R, M) = (6, 2, 10, 4)$ and (b) simulation parameters.

TABLE 1: System parameters for iteration-population size trade-off.

System parameters												
K	N_T	N_R	M	T	Search space	STBC type	Channel type			No. of simulation runs		
5	2	8	4	2	4^{10}	Alamouti	Quasi-static fading			2000		
Shared EAs				BBO		DABC		GA			EDA	
Generation		Pop	I	m	Migration	t	P_{xover}	P_{mut}	P_{sel}	P_{xover}	P_{mut}	P_{sel}
1~120		100	1	0.015	Constant	$0.4 \times pop$	0.9	0.5	0.5	0.99	0.95	0.5

much less than the optimal ML detector with a search space of 4^{10} possible solutions. As a result, DABC would be a significantly considerable choice for joint symbol detection in MD-STBC-MIMO systems.

7. Conclusion and Future Work

In this paper, a modified version of the Artificial Bee Colony algorithm is presented for the optimization problems in discrete domain and is applied to a Multi Device (MD) Space-Time Block Code (STBC) Multi-Input Multi-Output (MIMO) system. The enhancements in this algorithm have reduced its complexity, which is much less as compared with optimal ML detector. Thus it is suitable for cost-effective high-speed real-time communications. In addition, compared to other evolutionary algorithms like GA, EDA, and BBO, the presented discrete ABC (DABC) detection in MD-STBC-MIMO shows a significantly better performance. The proposed algorithm also has consistently better performance-complexity trade-offs at low SNRs, in comparison to the existing algorithms. Even at high SNRs, this algorithm has relatively good performance-complexity trade-offs. In conclusion, the proposed DABC is suitable for cost-effective high-speed real-time communications in MIMO systems.

DABC is a good candidate for solving the same type of computationally complex problems in wireless communication because of its simplistic model, low implementation complexity, and convergence to a nearly optimal solution with a small number of iterations. In this work, our main purpose was to employ a low-complexity algorithm that can

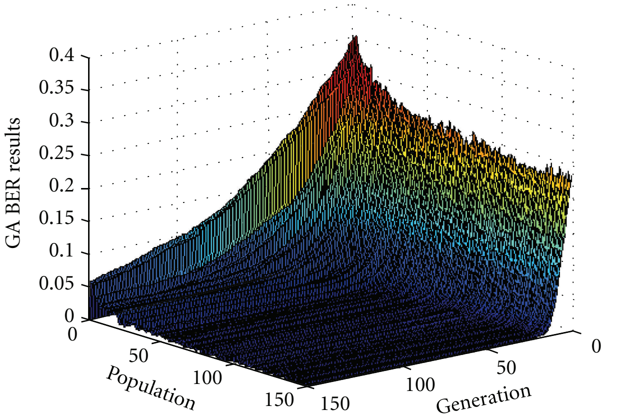


FIGURE 6: A tradeoff between population size and iterations for GA. $(K, N_T, N_R, M) = (4, 2, 6, 4)$.

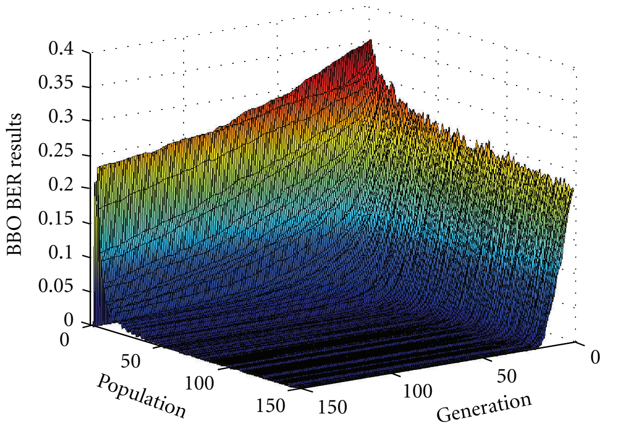


FIGURE 7: A trade-off between population size and iterations for BBO. $(K, N_T, N_R, M) = (4, 2, 6, 4)$.

beat the existing suboptimal detection schemes. To this end, some of the evolutionary procedures of the algorithm have been modified.

Currently, the authors are working on hybridizing DABC with other EAs to take the best aspects of each algorithm for specific applications. The authors are also working on applying DABC to other types of optimization problems, such as multiobjective or constrained optimization problems.

Appendix

Food Source Selection in a Neighborhood in DABC

In (8), ϕ_{ij} is a random number between $[-1, 1]$. We choose the minimum and maximum of this interval and apply it to (8). The minimum is $\phi_{ij} = -1$; hence:

$$v_j^i = x_j^i + (-1)\left(x_j^i - x_j^l\right)$$
$$v_j^i = x_j^l. \tag{A.1}$$

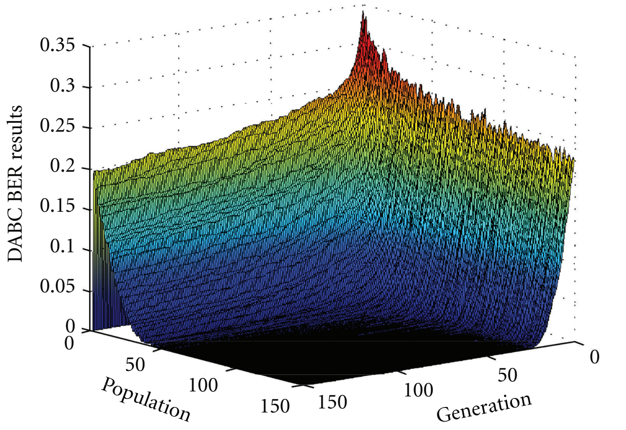


FIGURE 8: A trade-off between population size and iterations for ABC. $(K, N_T, N_R, M) = (4, 2, 6, 4)$.

The maximum is $\phi_{ij} = +1$; therefore:

$$v_j^i = x_j^i + (1)\left(x_j^i - x_j^l\right)$$
$$v_j^i = 2x_j^i - x_j^l. \tag{A.2}$$

Putting (A.1) and (A.2) together and using a random integer number generator function randint(a, b) that returns a random integer between a and b, for the integer values of x we obtain

$$v_j^i = \text{randint}\left[x_j^l, 2x_j^i - x_j^l\right]. \tag{A.3}$$

Acknowledgment

This work was supported in part by a discovery grant from the National Science and Engineering Research Council (NSERC) of Canada.

References

[1] 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, pp. 311–335, 1998.

[2] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[3] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744–765, 1998.

[4] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, Cambridge, UK, 2003.

[5] B. Hassibi and H. Vikalo, "On the sphere decoding algorithm: part I, the expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, 2005.

[6] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1474–1484, 2005.

[7] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization," Tech. Rep. Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[8] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[9] S. Ashrafinia, U. Pareek, M. Naeem, and D. Lee, "Biogeography-based optimization for joint relay assignment and power allocation in cognitive radio systems," in *Proceedings of the Symposium Series on Computational Intelligence, IEEE SSCI 2011— IEEE Symposium on Swarm Intelligence (SIS '11)*, pp. 237–244, Paris, France, April 2011.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, November/December 1995.

[11] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[12] D. Karaboga, C. Ozturk, N. Karaboga, and B. Gorkemli, "Artificial bee colony programming for symbolic regression," *Information Sciences*, vol. 209, pp. 1–15, 2012.

[13] N. Q. Uy, N. X. Hoai, M. O'Neill, R. I. McKay, and E. Galván-López, "Semantically-based crossover in genetic programming: application to real-valued symbolic regression," *Genetic Programming and Evolvable Machines*, vol. 12, no. 2, pp. 91–119, 2011.

[14] S. N. Omkar, J. Senthilnath, R. Khandelwal, G. Narayana Naik, and S. Gopalakrishnan, "Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 489–499, 2011.

[15] J. D. Schaffer, *Multi-objective optimization with vector evaluated genetic algorithms [Ph.D. thesis]*, 1984.

[16] C. Xu, H. Duan, and F. Liu, "Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning," *Aerospace Science and Technology*, vol. 14, no. 8, pp. 535–541, 2010.

[17] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328–348, 2009.

[18] J. Liang, M. Guo, Y. Fan, Y. Yin, and M. Ma, "SAR image segmentation based on Artificial Bee Colony algorithm," *Applied Soft Computing*, vol. 11, pp. 5205–5214, 2011.

[19] M. Cengiz Taplamacioglu and H. Gozde, "Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system," *Journal of the Franklin Institute*, vol. 348, pp. 1927–1946, 2011.

[20] M. Sonmez, "Artificial Bee Colony algorithm for optimization of truss structures," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2406–2418, 2011.

[21] H. Narasimhan, "Parallel artificial bee colony (PABC) algorithm," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 306–311, Coimbatore, India, December 2009.

[22] B. Akay, C. Ozturk, and D. Karaboga, "Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks," in *Proceedings of the 4th International Conference on Modeling Decisions for Artificial Intelligence (MDAI '07)*, Berlin, Germany, 2007.

[23] H. A. A. Bahamish, R. Abdullah, and R. A. Salam, "Protein tertiary structure prediction using artificial bee colony algorithm," in *Proceedings of the 3rd Asia International Conference on Modelling and Simulation (AMS '09)*, pp. 258–263, Bali, Indonesia, May 2009.

[24] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[25] M. Nekuii, M. Kisialiou, T. N. Davidson, and Z. Q. Luo, "Efficient soft demodulation of MIMO QPSK via semidefinite relaxation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 2665–2668, Las Vegas, Nev, USA, April 2008.

[26] J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Springer, 2002.

[27] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[28] B. Hassibi and B. M. Hochwald, "High-rate codes that are linear in space and time," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1804–1824, 2002.

[29] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[30] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.

[31] P. W. Tsai, J. S. Pan, B. Y. Liao, and S. C. Chu, "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 5081–5092, 2009.

[32] J. Wang, T. Li, and R. Ren, "Real time IDSs based on artificial bee colony-support vector machine algorithm," in *Proceedings of the 3rd International Workshop on Advanced Computational Intelligence (IWACI '10)*, pp. 91–96, Suzhou, China, August 2010.

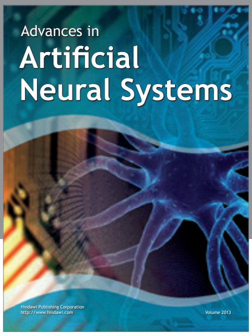
[33] M. Salim and M. T. Vakil-Baghmisheh, "Discrete bee algorithms and their application in multivariable function optimization," *Artificial Intelligence Review*, vol. 35, no. 1, pp. 73–84, 2011.

[34] O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Communications Letters*, vol. 4, no. 5, pp. 161–163, 2000.

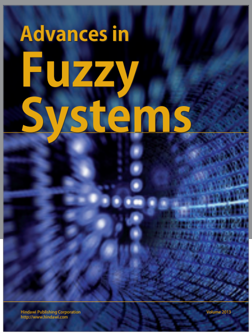
[35] C. Comaniciu, N. B. Mandayam, and H. V. Poor, *Wireless Networks: Multiuser Detection in Cross-Layer Design*, Springer, New York, NY, USA, 2005.

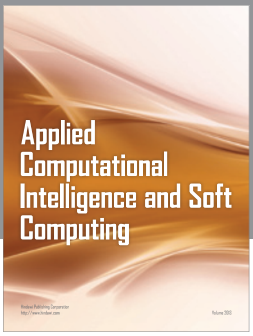
[36] M. Kisialiou and Z. Q. Luo, "Performance analysis of quasi-maximum-likelihood detector based on semi-definite programming," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, pp. III433–III436, March 2005.

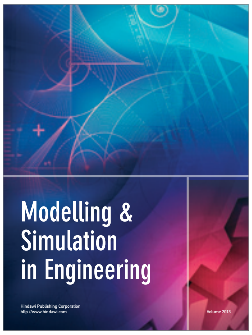
The Scientific World Journal

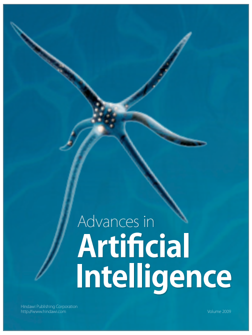

Advances in
Artificial Neural Systems

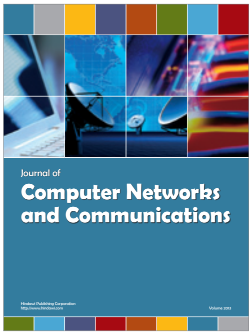
International Journal of
Computer Games Technology


Advances in
Fuzzy Systems

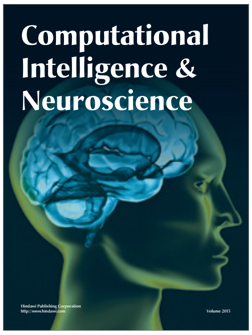

Applied Computational Intelligence and Soft Computing


Modelling & Simulation in Engineering


Advances in
Artificial Intelligence

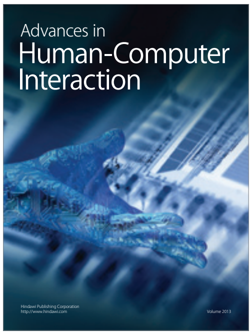

Journal of
Computer Networks and Communications

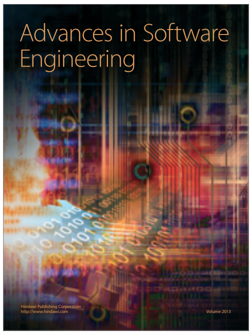
Journal of
Robotics


Computational Intelligence & Neuroscience

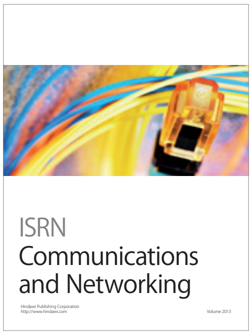
International Journal of
Reconfigurable Computing

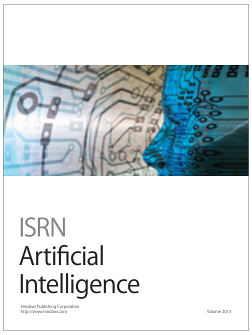
International Journal of
Distributed Sensor Networks

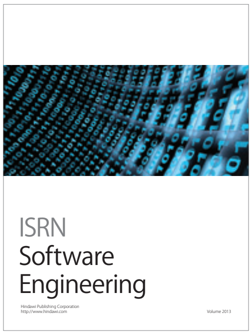

Advances in
Human-Computer Interaction

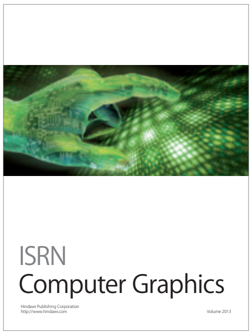

Advances in Software Engineering

ISRN
Machine Vision


ISRN
Communications and Networking


ISRN
Artificial Intelligence


ISRN
Software Engineering


ISRN
Computer Graphics