

Benchmarking Capabilities of Evolutionary Algorithms in Joint Channel Estimation and Turbo Multi-User Detection/Decoding

Jiankang Zhang¹, Sheng Chen^{2,3}, Xiaomin Mu¹, and Lajos Hanzo²

¹School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China

²Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

³Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Abstract—Joint channel estimation (CE) and turbo multi-user detection (MUD)/decoding for space-division multiple-access based orthogonal frequency-division multiplexing communication has to consider both the decision-directed CE optimisation on a continuous search space and the MUD optimisation on a discrete search space, and it iteratively exchanges the estimated channel information and the detected data between the channel estimator and the turbo MUD/decoder to gradually improve the accuracy of both the CE and the MUD. We evaluate the capabilities of a group of evolutionary algorithms (EAs) to achieve optimal or near optimal solutions with affordable complexity in this challenging application. Our study confirms that the EA assisted joint CE and turbo MUD/decoder is capable of approaching both the Cramér-Rao lower bound of the optimal channel estimation and the bit error ratio performance of the idealised optimal turbo maximum likelihood (ML) MUD/decoder associated with the perfect channel state information, respectively, despite only imposing a fraction of the complexity of the idealised turbo ML-MUD/decoder.

Keywords—Genetic algorithm, repeated weighted boosting search, particle swarm optimisation, differential evolution algorithm

I. INTRODUCTION

Evolutionary algorithms (EAs) have found ever-increasing applications in all walks of engineering, where attaining optimal designs at affordable computational costs are critical. Popular EAs include genetic algorithms (GAs) [1], [2], repeated weighted boosting search (RWBS) [3], [4], particle swarm optimisation (PSO) [5], [6], and differential evolution algorithm (DEA) [7], [8]. There are many other popular EAs, e.g. ant colony optimisation [9], [10], but given a limited space, only the above four algorithms are included. Vast amounts of empirical results have demonstrated that appropriately tuned EAs are capable of attaining the optimal or near optimal solutions even for the most challenging optimisation problems, with affordable computational complexity. Moreover, theoretical analysis of EAs has made significant progresses [11]. We now know that many NP-hard problems can be turned into the so-called EA-easy class [11], namely, they can be solved by a well-tuned EA algorithm with a complexity at most polynomial in the problem size, instead of at least exponential in the problem size. Empirical experience and theoretical analysis all suggest that whether a particular NP-hard problem can be turned into an EA-easy one by a given EA critically depends on whether the algorithmic parameter settings of this EA are appropriately matched to the given problem.

Orthogonal frequency-division multiplexing (OFDM) [12], [13] has found its way into many wireless network standards, owing to its virtues of resilience to frequency selective fading and low-complexity hardware implementations with the aid of inverse fast Fourier transform (IFFT) and fast Fourier transform (FFT). To increase the achievable system capacity, space-

division multiple-access (SDMA) systems are conceived [14], [15], where several users, roaming in different geographical locations and sharing the same bandwidth and time slots, are differentiated by their user-specific channel impulse responses (CIRs). In the uplink (UL) of an SDMA induced multi-user OFDM (SDMA/OFDM) system [16], [17], the transmitted signals of several single-antenna mobile stations (MSs) are received by an array of antennas at the base station (BS), where a powerful turbo soft-in soft-out (SISO) multi-user detection (MUD) and decoding technique [18] is invoked for recovering the information sequences of the different MSs, based on the estimates of their user-specific CIRs. A turbo MUD/decoder can exploit the error correction capability of the channel code by exchanging extrinsic information between the MUD and the channel decoder [18]. Naturally, its performance critically depends on the accuracy of the CIR estimates, and this motivates intensive efforts to develop an iterative blind joint channel estimation (CE) and turbo MUD/decoding structure [19], within which the turbo MUD/decoder can feed back more reliable detected data to assist the decision-directed channel estimator, and likewise, more accurate channel estimates will result in a more accurate MUD/decoder.

Joint CE and MUD/decoding for SDMA/OFDM systems represents an ideal benchmark application for evaluating various EAs. Firstly, turbo MUD/decoding optimisation given the channel state information (CSI) is NP-hard, and the optimal maximum likelihood (ML) solution is computationally prohibitive in general. Furthermore, within the joint optimisation of iterative CE and MUD/decoding, the CE optimisation is defined on a continuous search space while the MUD optimisation is defined on a discrete search space. Thus, both discrete and continuous EAs are required. We evaluate the four EAs, GA, RWBS, PSO and DEA, under the challenging framework of joint CE and MUD/decoding, in terms of achievable performance, computational complexity and convergence characteristics. Continuous EAs are employed in solving the associated CE optimisation, while the corresponding discrete binary EAs are employed to find the ML or near ML solution for the turbo MUD/decoding. We demonstrate the power and efficiency of the EA aided joint CE and turbo MUD/decoder in an extensive simulation study. Our results confirm that the channel estimate and the bit error ratio (BER) performance of the EA assisted iterative CE and turbo MUD/decoder scheme approach the Cramér-Rao lower bound (CRLB) of the optimal channel estimation [20] and the optimal ML MUD/decoding performance, respectively, with a computational complexity that is a fraction of the NP-hard optimal ML complexity.

II. MULTI-USER SDMA/OFDM SYSTEM

The multi-user SDMA/OFDM system considered supports U single-antenna MSs simultaneously transmitting in the UL

to the BS employing an array of Q receiving antennas. The quadrature amplitude modulation (QAM) [21] is adopted.

A. System model

The bit sequence \mathbf{b}^u of user u , $1 \leq u \leq U$, is encoded by a channel encoder. The bit stream after the encoder is passed through an interleaver to yield a coded bit stream, which is then modulated every $A = \log_2 M$ bits as a unit into a stream of M -QAM symbols. The modulated data are serial to parallel converted and the pilot symbols are added to yield the frequency domain (FD) representation of the OFDM symbol $X^u[s, k]$, $1 \leq k \leq K$, where K is the number of subcarriers and s is the OFDM symbol index. The parallel modulated data are fed to a K -point IFFT modulator to generate the time-domain (TD) signal $x^u[s, k]$. After concatenating the cyclic-prefix (CP) of K_{cp} samples, the resulting sequence is transmitted. The length of the CP is chosen as $K_{cp} \geq L_{cir}$, where L_{cir} is the length of the CIRs.

At the BS, the received signals \mathbf{y}_q , $1 \leq q \leq Q$, are parallel to serial converted and the CPs are discarded. The resulting signals are fed into the K -point FFT receiver. The signal received by the q -th receive antenna in the k -th subcarrier of the s -th OFDM symbol is expressed by [13]

$$Y_q[s, k] = \sum_{u=1}^U H_q^u[s, k] X^u[s, k] + W_q[s, k], \quad (1)$$

where $H_q^u[s, k]$ is the FD channel transfer function (FD-CHTF) coefficient of the link between the u -th user and the q -th receiver antenna in the k -th subcarrier of the s -th OFDM symbol, while $W_q[s, k]$ is the FD additive white Gaussian noise (AWGN) having the power of $N_o = 2\sigma_n^2$. Let $\mathbf{h}_q^u[s] \in \mathbb{C}^{L_{cir} \times 1}$ be the CIR vector between the u -th user and the q -th receive antenna during the s -th OFDM symbol period, which contains the L_{cir} CIR coefficients. The FD-CHTF vector, defined by

$$\mathbf{H}_q^u[s] = [H_q^u[s, 1] \ H_q^u[s, 2] \ \cdots \ H_q^u[s, K]]^T = \mathbf{F} \mathbf{h}_q^u[s], \quad (2)$$

is the K -point FFT of $\mathbf{h}_q^u[s]$, where $\mathbf{F} \in \mathbb{C}^{K \times L_{cir}}$ denotes the FFT matrix [13]. Let $\mathbf{Y}_q[s] \in \mathbb{C}^{K \times 1}$ host the subcarrier related signals $Y_q[s, k]$ be

$$\mathbf{Y}_q[s] = [Y_q[s, 1] \ Y_q[s, 2] \ \cdots \ Y_q[s, K]]^T, \quad 1 \leq q \leq Q, \quad (3)$$

and arrange each user's data in a diagonal matrix

$$\mathbf{X}^u[s] = \text{diag}\{X^u[s, 1], \dots, X^u[s, K]\}, \quad 1 \leq u \leq U. \quad (4)$$

Further define the CIR vector $\mathbf{h}_q[s] \in \mathbb{C}^{UL_{cir} \times 1}$ as

$$\mathbf{h}_q[s] = [(\mathbf{h}_q^1[s])^T \ (\mathbf{h}_q^2[s])^T \ \cdots \ (\mathbf{h}_q^U[s])^T]^T, \quad 1 \leq q \leq Q. \quad (5)$$

The basic operations of the BS receiver are: given $\{\mathbf{Y}_q[s]\}_{q=1}^Q$, find $\{\mathbf{h}_q[s]\}_{q=1}^Q$ as well as $\{\mathbf{X}^u[s]\}_{u=1}^U$ and hence recover the transmitted users' information bit streams $\{\mathbf{b}^u\}_{u=1}^U$.

B. Optimisation in joint CE and MUD

The joint ML CE and MUD is defined as

$$(\hat{\mathbf{h}}[s], \hat{\mathbf{X}}[s]) = \arg \left\{ \min_{\mathbf{h}[s], \mathbf{X}[s]} J(\mathbf{h}[s], \mathbf{X}[s]) \right\}, \quad (6)$$

where the cost function (CF) is given by

$$J(\mathbf{h}[s], \mathbf{X}[s]) = \sum_{q=1}^Q \|\mathbf{Y}_q[s] - \mathbf{X}^T[s] \bar{\mathbf{F}} \mathbf{h}_q[s]\|^2, \quad (7)$$

with the block-diagonal matrix $\bar{\mathbf{F}} \in \mathbb{C}^{UK \times UL_{cir}}$ given by

$$\bar{\mathbf{F}} = \text{diag}\{\underbrace{\mathbf{F}, \mathbf{F}, \dots, \mathbf{F}}_U\}, \quad (8)$$

while

$$\mathbf{h}[s] = [\mathbf{h}_1^T[s] \ \mathbf{h}_2^T[s] \ \cdots \ \mathbf{h}_Q^T[s]]^T, \quad (9)$$

$$\mathbf{X}[s] = [\mathbf{X}^1[s] \ \mathbf{X}^2[s] \ \cdots \ \mathbf{X}^U[s]]^T. \quad (10)$$

The joint ML optimisation (6) can be solved by an iterative search loop carried out first over the continuous space of possible channels $\mathbf{h}[s]$ and then over the discrete set of all the possible transmitted data $\mathbf{X}[s]$. The CE can be performed given the detected data $\hat{\mathbf{X}}[s]$ fed back from the MUD/decoder, while the MUD can be carried out with the estimated CIRs $\hat{\mathbf{h}}[s]$ provided by the channel estimator. The iterative procedure between the channel estimator and the MUD gradually improves the both solutions, and a few iterations are required to approach the joint ML solution (6).

1) *ML CE*: Since the CIRs $\mathbf{h}_q[s]$, $1 \leq q \leq Q$, only relate to the received signals $\mathbf{Y}_q[s]$, the ML CE solution $\hat{\mathbf{h}}[s]$ is obtained as the solutions of the Q smaller minimisation problems

$$\hat{\mathbf{h}}_q[s] = \arg \left\{ \min_{\mathbf{h}_q[s]} J_{ce}(\mathbf{h}_q[s]) \right\}, \quad 1 \leq q \leq Q, \quad (11)$$

in which the CF is expressed as

$$J_{ce}(\mathbf{h}_q[s]) = \|\mathbf{Y}_q[s] - \hat{\mathbf{X}}^T[s] \bar{\mathbf{F}} \mathbf{h}_q[s]\|^2. \quad (12)$$

As $\mathbf{h}_q[s] \in \mathbb{C}^{UL_{cir} \times 1}$, the search space for each optimisation defined in (11) is a continuous $(2UL_{cir})$ -dimensional space. We will omit the OFDM symbol index $[s]$ in the sequel.

2) *ML MUD*: Define the received data vector of the Q antennas and the transmitted signal vector of the U users as

$$\mathbf{Y}[s, k] = [Y_1[s, k] \ Y_2[s, k] \ \cdots \ Y_Q[s, k]]^T, \quad (13)$$

$$\mathbf{X}[s, k] = [X^1[s, k] \ X^2[s, k] \ \cdots \ X^U[s, k]]^T, \quad (14)$$

respectively. Further denote the FD-CHTF matrix linking $\mathbf{X}[s, k]$ to $\mathbf{Y}[s, k]$ as $\mathbf{H}[s, k] \in \mathbb{C}^{Q \times U}$. Since the M -QAM has a finite alphabet \mathcal{S} of size $|\mathcal{S}| = M$, there are M^U candidate solutions for $\mathbf{X}[s, k]$. The ML MUD exhaustively searches the full discrete space of \mathcal{S}^U to find the solution

$$\hat{\mathbf{X}}[s, k] = \arg \min_{\mathbf{X}[s, k] \in \mathcal{S}^U} J_{mud}(\mathbf{X}[s, k]), \quad (15)$$

where the MUD optimisation CF is expressed as

$$J_{mud}(\mathbf{X}[s, k]) = \|\mathbf{Y}[s, k] - \hat{\mathbf{H}}[s, k] \mathbf{X}[s, k]\|^2. \quad (16)$$

Since each $X^u[s, k]$ contains $A = \log_2 M$ bits, the bit-stream representation of $X^u[s, k]$ is $\mathbf{b}^u[s, k] = [b_1^u[s, k] \ b_2^u[s, k] \ \cdots \ b_A^u[s, k]]^T$, where $b_i^u[s, k] \in \{0, 1\}$. Thus, the bit-stream representation for $\mathbf{X}[s, k]$ is

$$\mathbf{b}[s, k] = [b_1^1[s, k] \ \cdots \ b_A^1[s, k] \ \cdots \ b_1^U[s, k] \ \cdots \ b_A^U[s, k]]^T. \quad (17)$$

For notational simplification, the MUD CF is equivalently denoted as $J_{mud}(\mathbf{b}[s, k]) = J_{mud}(\mathbf{X}[s, k])$. Also the OFDM index and the subcarrier index $[s, k]$ will be omitted.

III. EVOLUTIONARY ALGORITHMS

The continuous GA aided CE (CGA-CE), CRWBS aided CE (CRWBS-CE), CPSO aided CE (CPSO-CE) and CDEA aided CE (CDEA-CE) are used to assist the CE optimisation, while the discrete-binary (DB) GA aided MUD (DBGA-MUD), DBRWBS aided MUD (DBRWBS-MUD), DBPSO aided MUD (DBPSO-MUD) and DBDEA aided MUD (DBDEA-MUD) are used to assist the MUD optimisation.

A. GA for iterative CE and MUD

1) *CGA-CE*: It evolves the population of the P_s candidate solutions, representing the estimates of \mathbf{h}_q . The p_s -th individual of the population in the g -th generation is expressed as

$$\hat{\mathbf{h}}_{q,g,p_s} = [\hat{h}_{q,g,p_s,1}^1 \cdots \hat{h}_{q,g,p_s,L_{cir}}^1 \cdots \hat{h}_{q,g,p_s,L_{cir}}^U]^T. \quad (18)$$

$\{\hat{\mathbf{h}}_{q,g,p_s}\}_{p_s=1}^{P_s}$ for $g = 1$ are randomly generate in the search space $(-1-j, +1+j)^{UL_{cir}}$, where $j = \sqrt{-1}$ is the imaginary axis. The fitness of $\hat{\mathbf{h}}_{q,g,p_s}$ is related to its CF by

$$f(\hat{\mathbf{h}}_{q,g,p_s}) = J_{ce}^{-1}(\hat{\mathbf{h}}_{q,g,p_s}) = \|\mathbf{Y}_q - \hat{\mathbf{X}}^T \mathbf{F} \hat{\mathbf{h}}_{q,g,p_s}\|^{-2}. \quad (19)$$

With the roulette wheel selection, the selection ratio r_s determines the fraction of the P_s individuals that survive. Denote two randomly selected parents for crossover as

$$\begin{cases} \hat{\mathbf{h}}_{q,g,mum} = [\hat{h}_{q,g,mum,1}^1 \cdots \hat{h}_{q,g,mum,l^*-1}^{u^*} \\ \hat{h}_{q,g,mum,l^*}^{u^*} \hat{h}_{q,g,mum,l^*+1}^{u^*} \cdots \hat{h}_{q,g,mum,L_{cir}}^U]^T, \\ \hat{\mathbf{h}}_{q,g,dad} = [\hat{h}_{q,g,dad,1}^1 \cdots \hat{h}_{q,g,dad,l^*-1}^{u^*} \\ \hat{h}_{q,g,dad,l^*}^{u^*} \hat{h}_{q,g,dad,l^*+1}^{u^*} \cdots \hat{h}_{q,g,dad,L_{cir}}^U]^T, \end{cases} \quad (20)$$

respectively, where the integers, u^* and l^* , are randomly and uniformly selected within $\{1, 2, \dots, U\}$ and $\{1, 2, \dots, L_{cir}\}$, respectively. The two new offsprings are produced as

$$\begin{cases} \hat{\mathbf{h}}_{q,g,os1} = [\hat{h}_{q,g,mum,1}^1 \cdots \hat{h}_{q,g,mum,l^*-1}^{u^*} \\ \hat{h}_{q,g,os1,l^*}^{u^*} \hat{h}_{q,g,os1,l^*+1}^{u^*} \cdots \hat{h}_{q,g,os1,L_{cir}}^U]^T, \\ \hat{\mathbf{h}}_{q,g,os2} = [\hat{h}_{q,g,dad,1}^1 \cdots \hat{h}_{q,g,dad,l^*-1}^{u^*} \\ \hat{h}_{q,g,os2,l^*}^{u^*} \hat{h}_{q,g,os2,l^*+1}^{u^*} \cdots \hat{h}_{q,g,os2,L_{cir}}^U]^T, \end{cases} \quad (21)$$

where for $l^* \leq l \leq L_{cir}$

$$\begin{cases} \hat{h}_{q,g,os1,l}^{u^*} = \hat{h}_{q,g,mum,l}^{u^*} - \beta(\hat{h}_{q,g,mum,l}^{u^*} - \hat{h}_{q,g,dad,l}^{u^*}), \\ \hat{h}_{q,g,os2,l}^{u^*} = \hat{h}_{q,g,dad,l}^{u^*} + \beta(\hat{h}_{q,g,mum,l}^{u^*} - \hat{h}_{q,g,dad,l}^{u^*}), \end{cases} \quad (22)$$

and β is a random value chosen in the range of $(0, 1)$. Assume that a gene $\hat{h}_{q,g,p_s,l}^{u^*}$ of the individual $\hat{\mathbf{h}}_{q,g,p_s}$ is randomly chosen for the mutation, which is given by

$$\check{h}_{q,g,p_s,l}^{u^*} = \hat{h}_{q,g,p_s,l}^{u^*} + \gamma(\alpha_m + j\beta_m), \quad (23)$$

where α_m and β_m are randomly generated within the range $(-1, 1)$, while γ is the mutation parameter. How many genes will mutate is governed by the mutation probability M_b . The optimisation procedure is terminated when a pre-defined maximum number of generations, G_{max} , is reached.

2) *DBGA-MUD*: The p_s -th individual of the population in the g -th generation is expressed as

$$\hat{\mathbf{b}}_{g,p_s} = [\hat{b}_{g,p_s,1}^1 \cdots \hat{b}_{g,p_s,A}^1 \cdots \hat{b}_{g,p_s,1}^U \cdots \hat{b}_{g,p_s,A}^U]^T, \quad (24)$$

which is related to a modulated signal $\hat{\mathbf{X}}_{g,p_s}$ that is a candidate solution of the MUD optimisation (15). The P_s binary-valued initial individuals $\{\hat{\mathbf{b}}_{1,p_s}\}_{p_s=1}^{P_s}$ are randomly generated. The fitness of $\hat{\mathbf{b}}_{g,p_s}$ is related to its CF by

$$f(\hat{\mathbf{b}}_{g,p_s}) = J_{mud}^{-1}(\hat{\mathbf{b}}_{g,p_s}) = \|\mathbf{Y} - \hat{\mathbf{H}} \hat{\mathbf{X}}_{g,p_s}\|^{-2}. \quad (25)$$

The selection ratio r_s specifies the percentage of the P_s individuals that form the mating pool, and the roulette wheel selection is adopted, while the uniform crossover is used. Given the mutation probability M_b , $[M_b P_s U A]$ bits are randomly selected from the total of $P_s U A$ bits in the P_s individuals for mutation, where $\lfloor \bullet \rfloor$ denotes the integer floor.

B. RWBS for iterative CE and MUD

1) *CRWBS-CE*: The outer generation loop begins with initialising the population $\{\hat{\mathbf{h}}_{q,g,p_s}\}_{p_s=1}^{P_s}$ according to:

$$\hat{\mathbf{h}}_{q,g,1} = \hat{\mathbf{h}}_{q,g-1,best}, \quad (26)$$

$$\hat{\mathbf{h}}_{q,g,p_s} = \hat{\mathbf{h}}_{q,g-1,best} + \gamma(\mathbf{G}\mathbf{v} + j\mathbf{G}\mathbf{v}), 2 \leq p_s \leq P_s, \quad (27)$$

where $\mathbf{G}\mathbf{v}$ denotes the (UL_{cir}) -dimensional random vector whose elements follow the normal distribution with zero mean and unit variance, $\hat{\mathbf{h}}_{q,g-1,best}$ denotes the best individual found in the previous generation, and γ is the mutation rate. The estimate $\hat{\mathbf{h}}_{q,0,best}$ can be either randomly generated or chosen as the initial training based channel estimate. The CF values associated with the population are given by $J_{g,p_s} = J_{ce}(\hat{\mathbf{h}}_{q,g,p_s})$, $1 \leq p_s \leq P_s$, and each individual $\hat{\mathbf{h}}_{q,g,p_s}$ is initially assigned an equal weight $\delta_{p_s}(0) = \frac{1}{P_s}$, for $1 \leq p_s \leq P_s$.

In the inner weighted boosting search (WBS) loop, the best and worst individuals, $\hat{\mathbf{h}}_{q,g,p_{best}}$ and $\hat{\mathbf{h}}_{q,g,p_{worst}}$, are first found, where $p_{best} = \arg \min_{1 \leq p_s \leq P_s} J_{g,p_s}$ and $p_{worst} = \arg \max_{1 \leq p_s \leq P_s} J_{g,p_s}$. With the normalised CF values

$$\bar{J}_{g,p_s} = J_{g,p_s} / \sum_{j=1}^{P_s} J_{g,j}, 1 \leq p_s \leq P_s, \quad (28)$$

the weighting factor $\beta(t)$ is computed according to

$$\beta(t) = \frac{\eta(t)}{1 - \eta(t)} \text{ with } \eta(t) = \sum_{p_s=1}^{P_s} \delta_{p_s}(t-1) \bar{J}_{g,p_s}, \quad (29)$$

where t is the the WBS index, Then the weights are adapted:

$$\tilde{\delta}_{p_s}(t) = \begin{cases} \delta_{p_s}(t-1) (\beta(t))^{\bar{J}_{g,p_s}}, & \beta(t) \leq 1, 1 \leq p_s \leq P_s, \\ \delta_{p_s}(t-1) (\beta(t))^{1-\bar{J}_{g,p_s}}, & \beta(t) > 1, \end{cases} \quad (30)$$

$$\delta_{p_s}(t) = \tilde{\delta}_{p_s}(t) / \sum_{j=1}^{P_s} \tilde{\delta}_j(t), 1 \leq p_s \leq P_s. \quad (31)$$

Two new individuals are generated by the convex combination and the associated reflection operation, respectively, as

$$\hat{\mathbf{h}}_{q,g,P_s+1} = \sum_{p_s=1}^{P_s} \delta_{p_s}(t) \hat{\mathbf{h}}_{q,g,p_s}, \quad (32)$$

$$\hat{\mathbf{h}}_{q,g,P_s+2} = \hat{\mathbf{h}}_{q,g,p_{best}} + (\hat{\mathbf{h}}_{q,g,p_{best}} - \hat{\mathbf{h}}_{q,g,P_s+1}). \quad (33)$$

$\hat{\mathbf{h}}_{q,g,p_{\text{worst}}}$ in the population is then replaced by $\hat{\mathbf{h}}_{q,g,p_*}$, where $p_* = \arg \min_{i=P_s+1, P_s+2} J_{ce}(\hat{\mathbf{h}}_{q,g,i})$.

The inner WBS iterative procedure is terminated when the maximum number of WBS iterations T_{wbs} has been reached. The outer generation procedure is stopped when the maximum number of generations G_{max} has been reached.

2) *DBRWBS-MUD*: The outer generation loop begins with initialising the population $\{\hat{\mathbf{b}}_{g,p_s}\}_{p_s=1}^{P_s}$ as: set $\hat{\mathbf{b}}_{g,1} = \hat{\mathbf{b}}_{g-1,\text{best}}$, while the remaining $P_s - 1$ individuals $\hat{\mathbf{b}}_{g,p_s}$, $2 \leq p_s \leq P_s$, are generated by randomly muting a certain percentage of the bits in $\hat{\mathbf{b}}_{g-1,\text{best}}$, the best individual found in the previous generation. The percentage of bits muted are governed by the mutation probability M_b . The initial binary-valued $\hat{\mathbf{b}}_{0,\text{best}}$ can either be randomly generated or as the linear minimum mean square error (MMSE) MUD solution [22].

The operations of the DBRWBS-MUD are similar to those of the CRWBS-CE, except for the population updating. Specifically, given that the P_s individuals' weights $\delta_{p_s}(t)$ with $1 \leq p_s \leq P_s$ have been acquired, define

$$\begin{cases} \Delta\delta_0(t) = 0, \\ \Delta\delta_{p_s}(t) = \Delta\delta_{p_s-1}(t) + \delta_{p_s}(t), 1 \leq p_s \leq P_s. \end{cases} \quad (34)$$

Then the four new individuals $\hat{\mathbf{b}}_{g,P_s+i}$, $1 \leq i \leq 4$, are generated as follows: for $1 \leq a \leq A$ and $1 \leq u \leq U$,

$$\hat{b}_{g,P_s+i,a}^u = \hat{b}_{g,p_s,a}^u, \text{ if } \Delta\delta_{p_s-1}(t) < \text{rand}(0,1) \leq \Delta\delta_{p_s}(t), \quad (35)$$

where $\text{rand}(0,1)$ denotes the random number generator which randomly returns a value from the interval $[0, 1)$. The newly generated individuals replace the worst individuals in the population, whose CF values are larger than them.

C. PSO for iterative CE and MUD

1) *CPSO-CE*: The initial individuals $\{\hat{\mathbf{h}}_{q,1,p_s}\}_{p_s=1}^{P_s}$ are randomly generated in the search space $(-1-j, +1+j)^{UL_{\text{cir}}}$, and the initial velocities $\{\mathbf{v}_{q,1,p_s}\}_{p_s=1}^{P_s}$ are also randomly generated in the velocity space $(-1-j, +1+j)^{UL_{\text{cir}}}$. Each particle $\hat{\mathbf{h}}_{q,g,p_s}$ remembers its best position visited so far, which defines its cognitive information, denoted as $\hat{\mathbf{h}}_{q,g,p_s}^{\text{ci}}$. Every particle also knows the best position visited so far among the entire swarm, which defines the social information, denoted by $\hat{\mathbf{h}}_{q,g}^{\text{si}}$. The cognitive information $\{\hat{\mathbf{h}}_{q,g,p_s}^{\text{ci}}\}_{p_s=1}^{P_s}$ and the social information $\hat{\mathbf{h}}_{q,g}^{\text{si}}$ are updated at each generation. The velocities and positions are updated according to

$$\mathbf{v}_{q,g+1,p_s} = \omega \mathbf{v}_{q,g,p_s} + c_1 \text{rand}(0,1) (\hat{\mathbf{h}}_{q,g,p_s}^{\text{ci}} - \hat{\mathbf{h}}_{q,g,p_s}) + c_2 \text{rand}(0,1) (\hat{\mathbf{h}}_{q,g}^{\text{si}} - \hat{\mathbf{h}}_{q,g,p_s}), \quad (36)$$

$$\hat{\mathbf{h}}_{q,g+1,p_s} = \hat{\mathbf{h}}_{q,g,p_s} + \mathbf{v}_{q,g+1,p_s}, \quad (37)$$

for $1 \leq p_s \leq P_s$, where ω is the inertia weight, while c_1 and c_2 are the cognitive learning rate and the social learning rate, respectively. The optimisation procedure is terminated when the maximum number of generations G_{max} is reached.

2) *DBPSO-MUD*: Each individual $\hat{\mathbf{b}}_{g,p_s}$, as defined in (24), is associated with the velocity

$$\mathbf{v}_{g,p_s} = [v_{g,p_s,1}^1 \cdots v_{g,p_s,A}^1 \cdots v_{g,p_s,1}^U \cdots v_{g,p_s,A}^U]^T, \quad (38)$$

where $\mathbf{v}_{g,p_s} \in (0, 1)^{UA}$ [23]. Associated with $\hat{\mathbf{b}}_{g,p_s}$ are the two bit-changing probability vectors given respectively by

$$\mathbf{v}_{g,p_s}^0 = [v_{g,p_s,1}^{1,0} \cdots v_{g,p_s,A}^{1,0} \cdots v_{g,p_s,1}^{U,0} \cdots v_{g,p_s,A}^{U,0}]^T, \quad (39)$$

$$\mathbf{v}_{g,p_s}^1 = [v_{g,p_s,1}^{1,1} \cdots v_{g,p_s,A}^{1,1} \cdots v_{g,p_s,1}^{U,1} \cdots v_{g,p_s,A}^{U,1}]^T, \quad (40)$$

where $v_{g,p_s,l}^{u,0}$ is the probability of the bit $\hat{b}_{g,p_s,l}^u$ to change to 0, while $v_{g,p_s,l}^{u,1}$ is the probability of $\hat{b}_{g,p_s,l}^u$ to change to 1. The initial $\{\hat{\mathbf{b}}_{1,p_s}\}_{p_s=1}^{P_s}$ are randomly generated, and the initial $\{\mathbf{v}_{1,p_s}^0\}_{p_s=1}^{P_s}$ and $\{\mathbf{v}_{1,p_s}^1\}_{p_s=1}^{P_s}$ are also randomly generated over the probability space $[0, 1]^{UA}$. At each generation, the cognitive information $\{\hat{\mathbf{b}}_{g,p_s}^{\text{ci}}\}_{p_s=1}^{P_s}$ and the social information $\hat{\mathbf{b}}_g^{\text{si}}$ are updated. The two sets of the bit-changing probability vectors are updated according to [24]

$$\mathbf{v}_{g+1,p_s}^0 = \omega \mathbf{v}_{g,p_s}^0 + c_1 \text{rand}(0,1) (\mathbf{1}_{UA} - 2\hat{\mathbf{b}}_{g,p_s}^{\text{ci}}) + c_2 \text{rand}(0,1) (\mathbf{1}_{UA} - 2\hat{\mathbf{b}}_g^{\text{si}}), \quad (41)$$

$$\mathbf{v}_{g+1,p_s}^1 = \omega \mathbf{v}_{g,p_s}^1 + c_1 \text{rand}(0,1) (2\hat{\mathbf{b}}_{g,p_s}^{\text{ci}} - \mathbf{1}_{UA}) + c_2 \text{rand}(0,1) (2\hat{\mathbf{b}}_g^{\text{si}} - \mathbf{1}_{UA}), \quad (42)$$

for $1 \leq p_s \leq P_s$, where $\mathbf{1}_{UA}$ is the UA -dimensional vector with all elements equal to 1. The intermediate velocity of the bit $\hat{b}_{g,p_s,l}^u$, $1 \leq l \leq A$ and $1 \leq u \leq U$, is defined as [24]

$$\tilde{v}_{g+1,p_s,l}^u = \begin{cases} v_{g+1,p_s,l}^{u,1}, & \text{if } \hat{b}_{g,p_s,l}^u = 0, \\ v_{g+1,p_s,l}^{u,0}, & \text{if } \hat{b}_{g,p_s,l}^u = 1, \end{cases} \quad (43)$$

and the velocity associated with $\hat{b}_{g,p_s,l}^u$ is then given by [23]

$$v_{g+1,p_s,l}^u = \frac{1}{1 + e^{-\tilde{v}_{g+1,p_s,l}^u}}. \quad (44)$$

Next, the individuals are updated as follows

$$\hat{b}_{g+1,p_s,l}^u = \begin{cases} \hat{b}_{g,p_s,l}^u, & \text{if } \text{rand}(0,1) \leq v_{g+1,p_s,l}^u, \\ 1 - \hat{b}_{g,p_s,l}^u, & \text{if } \text{rand}(0,1) > v_{g+1,p_s,l}^u, \end{cases} \quad (45)$$

for $1 \leq p_s \leq P_s$, $1 \leq u \leq U$, and $1 \leq l \leq A$.

D. DEA for iterative CE and MUD

1) *CDEA-CE*: The mean value of the crossover probability C_r is initialised to $\mu_{C_r} = 0.5$, while the location parameter of the scaling factor λ is initialised to $\mu_\lambda = 0.5$. The archive of DEA contains P_s best solutions that the population has explored, and is updated every generation by adding the $\lfloor P_s \cdot p \rfloor$ parent solutions that are in the top 100p% of high fitness, where p is known as the greedy factor. If the archive size exceeds P_s , some solutions are randomly removed from the archive to keep the archive size at P_s . The mutation operation perturbs each base population vector $\hat{\mathbf{h}}_{q,g,p_s}$ as follows

$$\hat{\mathbf{h}}_{q,g,p_s} = \hat{\mathbf{h}}_{q,g,p_s} + \lambda_{p_s} (\hat{\mathbf{h}}_{q,g,\text{best},r_1}^p - \hat{\mathbf{h}}_{q,g,p_s}) + \lambda_{p_s} (\hat{\mathbf{h}}_{q,g,r_2} - \hat{\mathbf{h}}_{q,g,r_3}), \quad (46)$$

where the scaling factor $\lambda_{p_s} \in (0, 1]$ is randomly generated for each individual according to a normal distribution of mean μ_λ and standard deviation 0.1, $\hat{\mathbf{h}}_{q,g,\text{best},r_1}^p$ is a randomly selected

archive value, while r_2 and r_3 are two random values fetched from $\{1, 2, \dots, (p_s - 1), (p_s + 1), \dots, P_s\}$. A trial vector $\tilde{\mathbf{h}}_{q,g,p_s}$ is generated by replacing certain elements of $\hat{\mathbf{h}}_{q,g,p_s}$ with the corresponding elements of $\tilde{\mathbf{h}}_{q,g,p_s}$ according to

$$\tilde{h}_{q,g,p_s,l}^u = \begin{cases} \tilde{h}_{q,g,p_s,l}^u, & \text{rand}(0, 1) \leq C_{r_{p_s}}, \\ \hat{h}_{q,g,p_s,l}^u, & \text{otherwise,} \end{cases} \quad (47)$$

where $C_{r_{p_s}} \in [0, 1]$ is the crossover probability, which is randomly generated for each individual according to a Cauchy distribution with the location parameter μ_{C_r} and the scale parameter 0.1. Whether the target vector $\hat{\mathbf{h}}_{q,g,p_s}$ or the trial vector $\tilde{\mathbf{h}}_{q,g,p_s}$ survives to the next generation is decided by

$$\hat{\mathbf{h}}_{q,(g+1),p_s} = \begin{cases} \tilde{\mathbf{h}}_{q,g,p_s}, & J_{ce}(\tilde{\mathbf{h}}_{q,g,p_s}) \leq J_{ce}(\hat{\mathbf{h}}_{q,g,p_s}), \\ \hat{\mathbf{h}}_{q,g,p_s}, & \text{otherwise.} \end{cases} \quad (48)$$

μ_{C_r} and μ_λ are self adapted according to [8]

$$\mu_{C_r} = (1 - c) \cdot \mu_{C_r} + c \cdot \text{mean}_A(S_{C_r}), \quad (49)$$

$$\mu_\lambda = (1 - c) \cdot \mu_\lambda + c \cdot \text{mean}_L(S_\lambda), \quad (50)$$

where $c \in (0, 1]$ controls the rate of adaptation, $\text{mean}_A(\cdot)$ and $\text{mean}_L(\cdot)$ denote the arithmetic-mean and Lehmer-mean [8] operators, respectively, while S_{C_r} and S_λ denote the sets of successful crossover probabilities C_{r_i} and scaling factors λ_i , respectively. The optimisation procedure is terminated when the maximum number of generations G_{\max} is reached.

2) DBDEA-MUD: The algorithm evolves the binary-valued individuals $\hat{\mathbf{b}}_{g,p_s}$ as defined in (24). Specifically, the mutant vector to the base vector $\hat{\mathbf{b}}_{g,i}$ is created according to

$$\hat{\mathbf{v}}_{g,i} = \hat{\mathbf{b}}_{g,i} \oplus \left(\mathbf{z}_i^b \otimes (\hat{\mathbf{b}}_{g,\text{best},r_1}^p \oplus \hat{\mathbf{b}}_{g,i}) \right) \oplus \left(\mathbf{z}_i^b \otimes (\hat{\mathbf{b}}_{g,r_2} \oplus \hat{\mathbf{b}}_{g,r_3}) \right), \quad (51)$$

where $\hat{\mathbf{b}}_{g,\text{best},r_1}^p$ is randomly chosen from the archive, $\hat{\mathbf{b}}_{g,r_2}$ and $\hat{\mathbf{b}}_{g,r_3}$, $r_2 \neq i$ and $r_3 \neq i$, are randomly selected from the current population, and \mathbf{z}_i^b is a randomly generated $(U \times A)$ -length binary vector known as the bit-scaling factor, while \oplus denotes the bit-wise exclusive-OR and \otimes denotes the bit-wise exclusive-AND. The trial vector $\hat{\mathbf{t}}_{g,p_s}$ is generated by

$$\hat{t}_{g,p_s,j}^u = \begin{cases} \hat{v}_{g,p_s,j}^u, & \text{rand}(0, 1) \leq C_{r_{p_s}} \text{ or } j = j_{\text{rand}}, \\ \hat{b}_{g,p_s,j}^u, & \text{otherwise,} \end{cases} \quad (52)$$

where $C_{r_{p_s}} \in [0, 1]$ is randomly generated by a normal distribution of mean μ_{C_r} and standard deviation 0.1, while j_{rand} is a randomly chosen integer in the range of $\{1, 2, \dots, P_s\}$. By mapping the binary-valued $\hat{\mathbf{t}}_{g,p_s}$ onto the corresponding M-QAM symbol vector $\hat{\mathbf{X}}_{g,p_s}^t$, the selection is defined by

$$\hat{\mathbf{b}}_{g+1,p_s} = \begin{cases} \hat{\mathbf{t}}_{g,p_s}, & J_{\text{mud}}(\hat{\mathbf{X}}_{g,p_s}^t) \leq J_{\text{mud}}(\hat{\mathbf{X}}_{g,p_s}^b), \\ \hat{\mathbf{b}}_{g,p_s}, & \text{otherwise.} \end{cases} \quad (53)$$

IV. EA AIDED JOINT CE AND TURBO MUD/DECODER

The iterative SISO detector and SISO channel decoder [18] is illustrated in Fig. 1, where the extrinsic information, which are the associated *a posteriori* information subtracted out the corresponding *a priori* information, are first interleaved/de-interleaved and exchanged a number of times between the

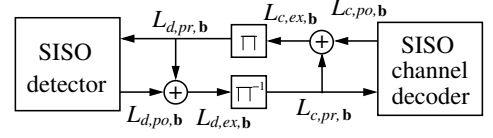


Fig. 1. Schematic of two-stage iterative detector and decoder, where L denotes the LLR, the subscripts d and c are associated with the detector and the channel decoder, respectively, while the subscripts pr , po and ex represent the *a priori*, *a posteriori* and extrinsic information, respectively.

detector and the decoder. The EA aided iterative CE and turbo MUD/decoder scheme exploits this powerful “turbo” iterative procedure in which the enhanced symbol estimates are fed back to the decision-directed channel estimator for improving the accuracy of the CE and more accurate channel estimates in turn result in an increasingly more accurate turbo MUD/decoder output. We now outline the operations of this EA aided iterative CE and turbo MUD/decoder.

1) Initialisation. Obtain the initial training based channel estimate, and start the iterative joint CE and turbo MUD/decoder with the iteration index $loop = 1$.

2) Iterative CE and turbo MUD/decoder.

2.1). Given the channel estimate provided by the continuous EA (CEA) aided channel estimator (if $loop = 1$, the training-based channel estimate), the turbo MUD/decoder starts with the iteration index $Iter = 1$.

3) Turbo MUD/decoder. The discrete-binary EA (DBEA) aided MUD detects the users’ data.

3.1). The SISO MUD delivers the *a posteriori* information of bit $b^u(i)$ in terms of log-likelihood ratio (LLR) as [18]

$$L_{m,po,b^u(i)} = \ln \frac{Pr\{\hat{X}^u | b^u(i) = 0\}}{Pr\{\hat{X}^u | b^u(i) = 1\}} + \ln \frac{Pr\{b^u(i) = 0\}}{Pr\{b^u(i) = 1\}} \\ = L_{m,e,b^u(i)} + L_{m,pr,b^u(i)}, \quad (54)$$

where $b^u(i)$ is the i -th bit in the bit stream that maps to the M-QAM symbol stream of user u . The second term in (54), $L_{m,pr,b^u(i)}$, represents the *a priori* LLR of the interleaved and encoded bits $b^u(i)$, while the first term in (54), $L_{m,e,b^u(i)}$, is the extrinsic information delivered by the SISO MUD, based on the received signal \mathbf{Y} and the *a priori* information about the encoded bits, except for the i -th bit of user u .

3.2). The extrinsic information provided by the SISO MUD is de-interleaved and fed into the SISO channel decoder as its *a priori* information, denoted as $L_{c,pr,b^u(i)}$. The SISO channel decoder then delivers the *a posteriori* information of decoded bits, in terms of LLRs $L_{c,po,b^u(i)}$, given by [18]

$$L_{c,po,b^u(i)} = L_{c,e,b^u(i)} + L_{c,pr,b^u(i)}. \quad (55)$$

The extrinsic information output by the SISO decoder, denoted by $L_{c,e,b^u(i)}$, will then be interleaved to provide the *a priori* information for the next iteration of the SISO MUD.

3.3). If $Iter < I_{tb}$, where I_{tb} defines the maximum number of turbo MUD/decoder iterations, set $Iter = Iter + 1$ and go to **3.1)**. Otherwise, the turbo MUD/decoder has converged, and the detected and decoded bit streams are encoded by the channel encoder, interleaved by the interleaver and then mapped to the corresponding M-QAM symbol streams, which will be used by the CEA based channel estimator.

4) Decision-directed channel estimator.

4.1). The CEA aided channel estimator uses the re-encoded and re-modulated data $\{\widehat{\mathbf{X}}^u\}_{u=1}^U$ to perform the CIR estimation. The resulting CIR estimate $\widehat{\mathbf{h}}$ is transformed to $\widehat{\mathbf{H}}$ by the FFT, which will then be used by the turbo MUD/decoder so that the iterative process can continue.

4.2). If $loop < I_{ce}$, where I_{ce} defines the maximum number of joint CE and turbo MUD/decoder iterations, set $loop = loop + 1$ and go to 2.1). Otherwise, the iterative CE and turbo MUD/decoder has converged.

Given the CSI, the computational complexity of the optimal ML-MUD, denoted as C_{MUD}^{ML} , and the complexity of the EA aided MUD, denoted as C_{MUD}^{EA} , can be analysed. Similarly, given the CSI, the complexity of the optimal turbo ML-MUD/decoder, denoted by C_{turbo}^{ML} , and the complexity of the EA assisted turbo MUD/decoder, denoted by C_{turbo}^{EA} , can be derived. We can further analyse the total complexity, denoted as C_{joint}^{EA} , of the EA assisted joint CE and turbo MUD/decoder. Space limitation preclude the explicit presentation of the results. In the following simulation study, we will demonstrate that the EA assisted joint CE and turbo MUD/decoder is capable of approaching the optimal performance of the idealised turbo ML-MUD/decoder associated with the perfect CSI, while imposing a complexity that is only a fraction of the complexity of the idealised turbo ML-MUD/decoder.

V. SIMULATION EXPERIMENTAL RESULTS

Our simulated SDMA/OFDM system's parameters are listed in Table I. A four-path Rayleigh fading channel model was employed for each link. At the beginning of every frame, which contained $S = 100$ OFDM symbols, a new channel tap was generated for each path according to the complex-valued white Gaussian process with its power specified by the square of the corresponding average path gain. Within the frame, each tap experienced independent Rayleigh fading having the normalised Doppler frequency $F_D = 10^{-7}$. A half-rate recursive systematic convolutional (RSC) code was employed as the channel code. The values of the EAs' algorithmic parameters are listed in Table II. The first OFDM symbol of each frame was employed as pilots for the initial training based CE, yielding a training overhead of 1%. The system's signal to noise ratio (SNR) was specified by $SNR = E_b/N_o$ in dB, where E_b denoted the energy per bit and N_o was the power spectral density of the channel AWGN.

TABLE I. SIMULATION SYSTEM'S PARAMETERS.

Encoder	Type	RSC
	Code rate	1/2
	Constraint length	3
	Polynomial	$(g_0, g_1) = (7, 5)$
Channel	Number of paths L_{cir}	4
	Delays	$\{0, 1, 2, 3\}$
	Average path gains	$\{0, -5, -10, -15\}$ (dB)
	Taps: frame to frame	Complex white Gaussian
	Taps: within frame	fading rate $F_D = 10^{-7}$
System	MSS U	4
	Receiver antennas Q	3
	Modulation	16-QAM
	Subcarriers K	64
	Cyclic prefix K_{cp}	16

TABLE II. THE ALGORITHMIC PARAMETERS FOR THE EA ASSISTED CE AND MUD

Scheme	Parameter	Value
CGA-CE	Population size P_s	100
	Selection ratio r_s	0.5
	Mutation parameter γ	0.01
	Mutation probability M_b	0.2
DBGA-MUD	Population size P_s	100
	Selection ratio r_s	0.5
	Mutation probability M_b	0.15
CRWBS-CE	Population size P_s	100
	Mutation parameter γ	0.001
	Weighted boosting search T_{wbs}	40
DBRWBS-MUD	Population size P_s	100
	Mutation probability M_b	0.5
	Weighted boosting search T_{wbs}	40
CPSO-C	Population size P_s	100
	Cognition learning rate c_1	2
	Social learning rate c_2	2
DBPSO-MUD	Population size P_s	100
	Cognition learning rate c_1	0.1
	Social learning rate c_2	0.3
CDEA-CE	Population size P_s	100
	Greedy factor p	0.1
	Adaptive update factor c	0.1
DBDEA-MUD	Population size P_s	100
	Greedy factor p	0.7
	Adaptive update factor c	0.8

A. Efficiency and reliability of EAs

We first quantified the efficiency and reliability of the CEA aided channel estimators and the DBEA based MUD schemes over $N_{tot} = 1000$ independent simulation runs. The perfect CSI was assumed when evaluating the DBEA assisted MUD schemes, while the transmitted data were available when evaluating the CEA aided CE schemes. There was no information exchange between the MUD and the decoder, i.e. $Iter = 1$. Moreover, the channel AWGN was absent, that is, $N_o = 0$. For an EA aided CE scheme, we specified a "successful" run as one in which the algorithm achieved the target CF value $J_{ce}(\widehat{\mathbf{h}}_{q, G_{max}^i, best}) < 10^{-4}$ within the set upper limit for the number of CF evaluations $\overline{N}_{CF-EV_s}^{lim} = P_s \cdot G_{max}^{lim} = 100 \times 1000$, where G_{max}^i denoted the number of generations in the i -th simulation run. Otherwise, the run was declared as a "failure" one. We collected the statistics of the number of successful runs, denoted as N_{suc} , the number of failure runs, denoted as N_{fai} , the total number of CF evaluations in the N_{suc} successful runs, defined by $N_{CF-EV_s}^{suc}$, and the total number of CF evaluations in the N_{fai} failure runs, defined by $N_{CF-EV_s}^{fai}$, using the following symbolic algorithm:

```

for run = 1 : N_tot
  if ( $G_{max}^{run} \leq G_{max}^{lim}$ ) and ( $J_{ce}(\widehat{\mathbf{h}}_{q, G_{max}^{run}, best}) < 10^{-4}$ )
     $N_{suc} = N_{suc} + 1$ 
     $N_{CF-EV_s}^{suc} = N_{CF-EV_s}^{suc} + P_s \cdot G_{max}^{run}$ 
  else
     $N_{fai} = N_{fai} + 1$ 
     $N_{CF-EV_s}^{fai} = N_{CF-EV_s}^{fai} + P_s \cdot G_{max}^{lim}$ 
  end if
end for

```

After obtaining these statistics, the average number of CF evaluations per run was given by

$$\overline{N}_{CF-EV_s}^{tot} = (N_{CF-EV_s}^{suc} + N_{CF-EV_s}^{fai}) / N_{tot}, \quad (56)$$

while the average number of CF evaluations per successful run was defined by

$$\overline{N}_{CF-EV_s}^{suc} = N_{CF-EV_s}^{suc} / N_{suc}. \quad (57)$$

Then the normalised average number of CF evaluations per run or the ratio of $\bar{N}_{CF-EV_s}^{\text{tot}}$ to $\bar{N}_{CF-EV_s}^{\text{lim}}$

$$\bar{R}_{CF-EV_s}^{\text{tot}} = \bar{N}_{CF-EV_s}^{\text{tot}} / \bar{N}_{CF-EV_s}^{\text{lim}} \quad (58)$$

and the normalised average number of CF evaluations per successful run or the ratio of $\bar{N}_{CF-EV_s}^{\text{suc}}$ to $\bar{N}_{CF-EV_s}^{\text{lim}}$

$$\bar{R}_{CF-EV_s}^{\text{suc}} = \bar{N}_{CF-EV_s}^{\text{suc}} / \bar{N}_{CF-EV_s}^{\text{lim}} \quad (59)$$

were used to quantify the efficiency of the EA aided CE scheme. The smaller $\bar{R}_{CF-EV_s}^{\text{tot}}$ or $\bar{R}_{CF-EV_s}^{\text{suc}}$, the more efficient the EA aided CE scheme. The reliability of the EA aided channel estimator was measured by the failure ratio

$$R_{\text{fai}} = N_{\text{fai}} / N_{\text{tot}}. \quad (60)$$

The smaller R_{fai} , the more reliable the EA aided CE scheme. The efficiency and reliability of the four CEA assisted CE schemes are illustrated in Fig. 2. It can be seen from Fig. 2 that the CDEA-CE outperformed the other three schemes and it always arrived at the target CF value with the average computational complexity of 15000 CF evaluations. The CRWBS-CE came a close second and it always attained the target CF value with the average computational complexity of 22000 CF evaluations. The CGA-CE was the the worst CE candidate, having the failure rate $R_{\text{fai}} \approx 7\%$ and requiring the average computational complexity of 90000 CF evaluations.

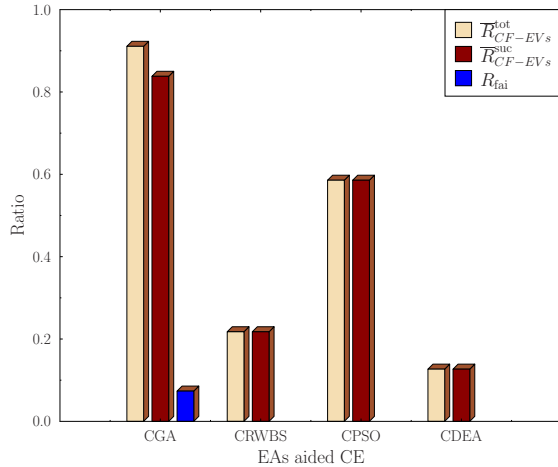


Fig. 2. Histograms of the efficiency and reliability measures, in terms of $\bar{R}_{CF-EV_s}^{\text{tot}}$, $\bar{R}_{CF-EV_s}^{\text{suc}}$ and R_{fai} , for the four CEA assisted CE schemes.

A similar procedure was carried out to investigate the efficiency and reliability of the four DBEA assisted MUDs by setting $G_{\text{max}}^{\text{lim}} = 500$ and $\bar{N}_{CF-EV_s}^{\text{lim}} = M^U = 16^4$. A successful detection run was confirmed, if ($G_{\text{max}}^{\text{run}} \leq G_{\text{max}}^{\text{lim}}$) and the BER of the best individual $\hat{\mathbf{X}}_{G_{\text{max}}^{\text{run}}, \text{best}}$ attains zero. Otherwise, the run was declared a failure. Note that $\bar{N}_{CF-EV_s}^{\text{lim}} = M^U$ was the number of CF evaluations required by the optimal full search ML MUD. Fig. 3 compares the efficiency and reliability of the four DBEA assisted MUDs. Observe in Fig. 3 that the DBGA-MUD was the “winning” MUD candidate with a zero failure rate and requiring the lowest average number of CF evaluations. The complexity of the DBGA-MUD was only 3.2% of the optimal ML-MUD’s complexity. The DBDEA-MUD came a close second with an extremely low failure rate

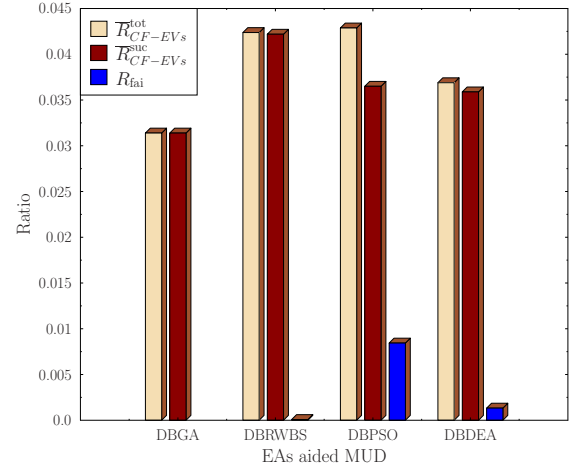


Fig. 3. Histograms of the efficiency and reliability measures, in terms of $\bar{R}_{CF-EV_s}^{\text{tot}}$, $\bar{R}_{CF-EV_s}^{\text{suc}}$ and R_{fai} , for the four DBEA assisted MUDs.

and an average complexity that was 3.7% of the optimal ML-MUD’s complexity. In terms of reliability, the DBRWBS-MUD also performed well with an almost zero failure rate, while the DBPSO-MUD had the highest failure rate.

B. EA aided joint CE and turbo MUD/decoder schemes

Figs. 4 and 5 compare the mean square error (MSE) and BER performance, respectively, of all the four EA aided iterative joint CE and turbo MUD/decoder schemes, when fixing the number of the inner turbo iterations to $Iter = 3$, the number of CF evaluations for EA aided CE to $N_{CF-EV_s}^{\text{ce}} = 20000$ ($G_{\text{max}} = 200$), and the number of CF evaluations for EA aided MUD to $N_{CF-EV_s}^{\text{mud}} = 10000$ ($G_{\text{max}} = 100$). Observe from Fig. 4 that, with the $loop = 5$ outer iterations, the MSEs of the two channel estimates associated with the RWBS and DEA aided iterative joint CE and turbo MUD/decoder schemes approached the CRLB for $E_b/N_0 \geq 10$ dB, but the PSO and GA aided iterative joint CE and turbo MUD/decoder schemes exhibited divergence. Similarly, from Fig. 5 it can be

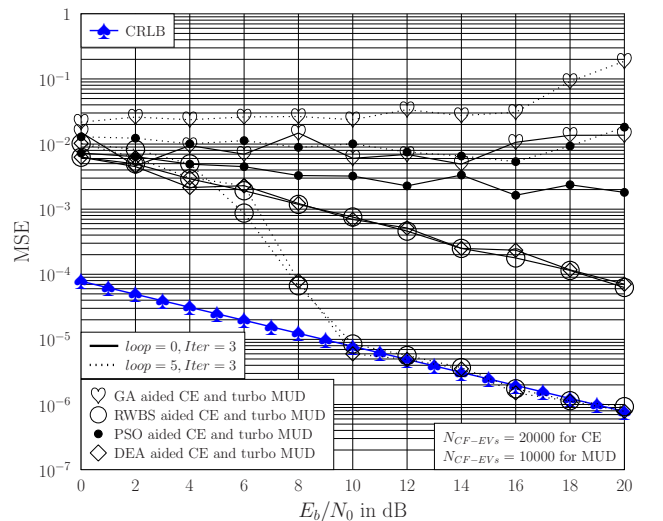


Fig. 4. Comparison of the MSE performance for the four EA aided iterative joint CE and turbo MUD/decoder schemes recorded at the outer iterations $loop = 0$ and $loop = 5$, respectively, when fixing the number of the inner turbo iterations to $Iter = 3$, the number of CF evaluations for EA aided CE to 20000, and the number of CF evaluations for EA aided MUD to 10000.

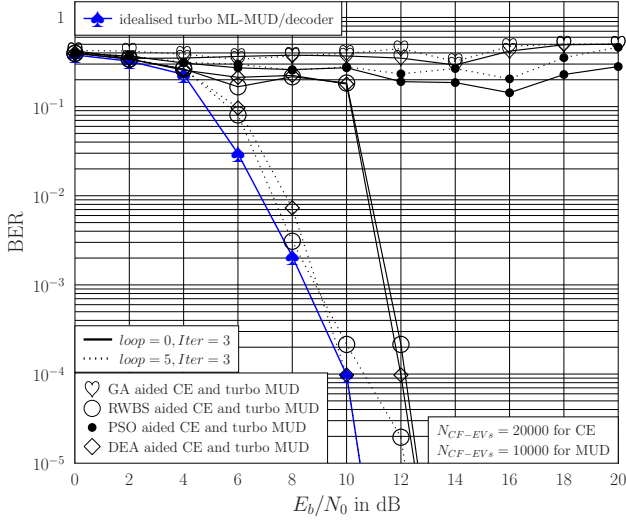


Fig. 5. Comparison of the BER performance for the four EA aided iterative joint CE and turbo MUD/decoder schemes recorded at the outer iterations $loop = 0$ and $loop = 5$, respectively, when fixing the number of the inner turbo iterations to $I_{ter} = 3$, the number of CF evaluations for EA aided CE to 20000, and the number of CF evaluations for EA aided MUD to 10000.

seen that, with the $loop = 5$ outer iterations, the RWBS and DEA aided iterative joint CE and turbo MUD/decoder schemes attained the optimal BER performance of the idealised turbo ML-MUD/decoder, while the PSO and GA aided iterative joint CE and turbo MUD/decoder schemes failed to find the optimal solution of the idealised turbo ML-MUD/decoder.

From the results of Subsections V-A, we suspect that the PSO and GA aided joint CE and turbo MUD/decoder schemes may be less efficient in comparison with the RWBS and DEA aided iterative joint CE and turbo MUD/decoder schemes, and $N_{CF-EV_s}^{ce} = 20000$ and $N_{CF-EV_s}^{mud} = 10000$ may not be sufficient for the PSO and GA aided schemes. We therefore increased $N_{CF-EV_s}^{ce} = 40000$ ($G_{max} = 400$) and $N_{CF-EV_s}^{mud} = 20000$ ($G_{max} = 200$), and carried out the simulation for all the four EA aided iterative joint CE and turbo MUD/decoder schemes again. Figs. 6 and 7 show the achievable MSE and BER performance, respectively, for the four EA aided iterative joint CE and turbo MUD/decoder schemes. It can readily be seen from Fig. 6 that the MSEs of the four channel estimates associated with the four EA aided joint CE and turbo MUD/decoder schemes all approached the CRLB with the $loop = 5$ outer iterations for $E_b/N_0 \geq 10$ dB. Furthermore, the four EA aided iterative joint CE and turbo MUD/decoder schemes all attained the optimal BER performance of the idealised turbo ML-MUD/decoder associated with the perfect CSI, as can be seen from Fig. 7.

We now explicitly compare the computational complexity of an EA assisted iterative joint CE and turbo MUD/decoder scheme with the complexity of the idealised turbo ML-MUD/decoder associated with the perfect CSI. Computational complexity comparison results are given in Table III in the three ratios $C_{MUD}^{EA}/C_{MUD}^{ML}$, $C_{turbo}^{EA}/C_{turbo}^{ML}$ and $C_{joint}^{EA}/C_{turbo}^{ML}$. The ratio $C_{MUD}^{EA}/C_{MUD}^{ML}$ is the ratio of the complexity of an EA aided MUD to that of the optimal full search ML MUD. It can be seen from Table III that all the four EA aided MUDs impose only 0.1% of the ML MUD's complexity. Recall that, given the CSI, a turbo MUD/decoder involves I_{ter} turbo iterations between the MUD and the channel

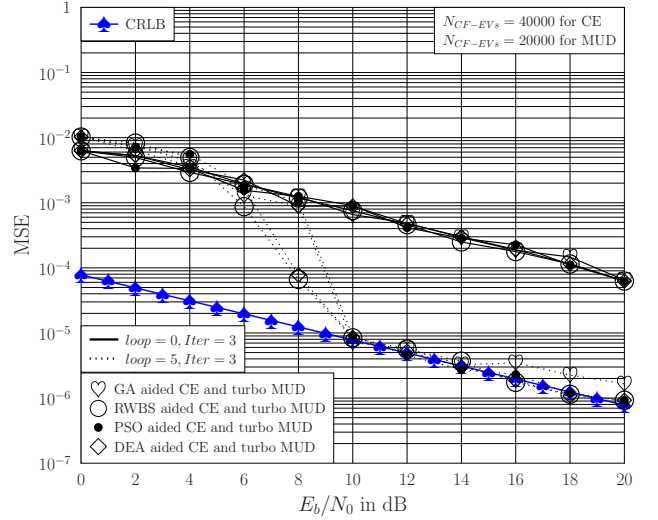


Fig. 6. Comparison of the MSE performance for the four EA aided iterative joint CE and turbo MUD/decoder schemes recorded at the outer iterations $loop = 0$ and $loop = 5$, respectively, when fixing the number of the inner turbo iterations to $I_{ter} = 3$, the number of CF evaluations for EA aided CE to 40000, and the number of CF evaluations for EA aided MUD to 20000.

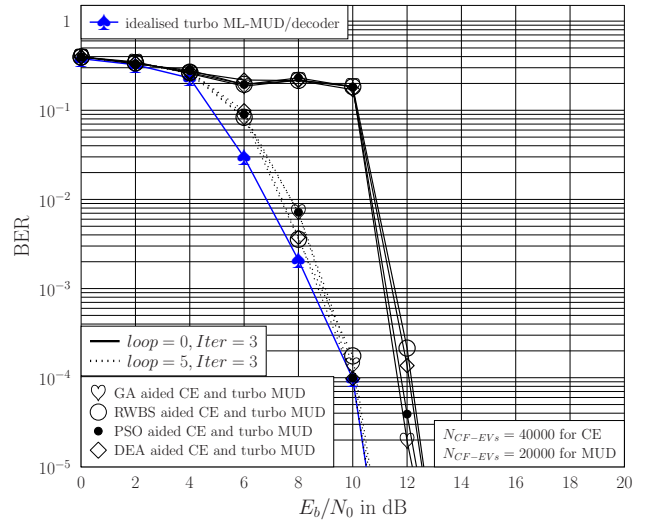


Fig. 7. Comparison of the BER performance for the four EA aided iterative joint CE and turbo MUD/decoder schemes recorded at the outer iterations $loop = 0$ and $loop = 5$, respectively, when fixing the number of the inner turbo iterations to $I_{ter} = 3$, the number of CF evaluations for EA aided CE to 40000, and the number of CF evaluations for EA aided MUD to 20000.

decoder. The complexity of the RWBS and DEA assisted turbo MUD/decoder algorithms is less than 3.5% of the complexity of the turbo ML-MUD/decoder, while the complexity of the GA and PSO aided turbo MUD/decoder algorithms is less than 6.6% of the turbo ML-MUD/decoder's complexity. The difference is owing to the fact that the RWBS and DEA appear to be more efficient requiring $N_{CF-EV_s}^{mud} = 10000$ compared with $N_{CF-EV_s}^{mud} = 20000$ needed by the GA and PSO. Note that an EA aided iterative joint CE and turbo MUD/decoder involves $loop$ outer iterations between the EA aided decision-directed channel estimator and the EA assisted turbo MUD/decoder, and it performs blind joint CE and data detection. Comparing its complexity with the complexity of the idealised turbo ML-MUD/decoder, which is given with the perfect CSI, is really "unfair". Even so, from Table III, we can

TABLE III. COMPUTATIONAL COMPLEXITY COMPARISON IN TERMS OF THE RATIO OF THE COMPLEXITY OF AN EA ASSISTED ITERATIVE JOINT CE AND TURBO MUD/DECODER TO THE COMPLEXITY OF THE IDEALISED TURBO ML-MUD/DECODER ASSOCIATED WITH THE PERFECT CHANNEL STATE INFORMATION.

Scheme	Operation	$C_{MUD}^{EA}/C_{MUD}^{ML}$	$C_{turbo}^{EA}/C_{turbo}^{ML}$	$C_{joint}^{EA}/C_{turbo}^{ML}$
GA assisted joint CE and turbo MUD/decoder	multiplications	0.10%	5.69%	62.24%
	additions	0.10%	7.45%	91.41%
RWBS assisted joint CE and turbo MUD/decoder	multiplications	0.10%	3.00%	31.27%
	additions	0.10%	3.88%	45.86%
PSO assisted joint CE and turbo MUD/decoder	multiplications	0.10%	5.69%	62.24%
	additions	0.10%	7.45%	91.41%
DE assisted joint CE and turbo MUD/decoder	multiplications	0.10%	3.00%	31.27%
	additions	0.10%	3.88%	45.86%

see that the total complexity of the RWBS and DEA assisted joint CE and turbo MUD/decoder schemes is less than 39% of the idealised turbo ML-MUD/decoder's complexity, while the GA and PSO assisted joint CE and turbo MUD/decoder schemes impose the total complexity that is less than 77% of the idealised turbo ML-MUD/decoder's complexity.

VI. CONCLUSIONS

In this paper, the four evolutionary algorithms, the GA, RWBS, PSO and DEA, have been applied to the challenging application of joint blind channel estimation and turbo multi-user detection/decoding for SDMA/OFDM communication systems. Extensive analysis and simulation study have been carried out to demonstrate that, by iteratively exchanging the information between a continuous EA aided decision-directed channel estimator and a discrete binary EA assisted turbo MUD/decoder, an EA aided joint blind CE and turbo MUD/decoder is capable of approaching both the Cramér-Rao lower bound associated with optimal channel estimate and the bit error rate of the idealised optimal turbo ML-MUD/decoder associated with the perfect channel state information, despite of only imposing a fraction of the complexity of the idealised turbo ML-MUD/decoder. Our study thus has provided benchmarking empirical results to support the capabilities of EAs for finding global or near global optimal designs in challenging practical applications with affordable computational complexity, and it compliments well the current intensive efforts of the computational intelligence community to better understand EAs.

REFERENCES

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.
- [2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [3] S. Chen, X. X. Wang, C. J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 35, no. 4, pp. 682–693, Aug. 2005.
- [4] S. F. Page, S. Chen, C. J. Harris, and N. M. White, "Repeated weighted boosting search for discrete or mixed search space and multiple-objective optimisation," *Applied Soft Computing*, vol. 12, no. 9, pp. 2740–2755, Sept. 2012.
- [5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in: *Proc. 1995 IEEE Int. Conf. Neural Networks* (Perth, Australia), Nov.27-Dec.1, 1995, vol. 4, pp. 1942–1948.
- [6] J. Kennedy and R. Eberhart, *Swarm Intelligence*. Morgan Kaufmann, 2001.
- [7] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer Verlag, 2005.
- [8] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, April 2009.
- [9] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, April 1997.
- [10] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
- [11] X. Yao, "Unpacking and understanding evolutionary algorithms," in: J. Liu, C. Alippi, B. Bouchon-Meuniev, G. W. Greenwood, and H. A. Abbass, eds., *Advances in Computational Intelligence*, Berlin, Heidelberg: Springer-Verlag, 2012, pp. 60–76.
- [12] J. A. C. Bingham, "Multicarrier modulation for data transmission: an idea whose time has come," *IEEE Communications Magazine*, Vol. 28, no. 5, pp. 5–14, May 1990.
- [13] L. Hanzo, M. Münster, B. J. Choi, and T. Keller, *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs, and Broadcasting*. Chichester, UK: Wiley, 2003.
- [14] P. Vandenameele, L. Van Der Perre, and M. Engels, *Space Division Multiple Access for Wireless Local Area Networks*. Boston, MA: Kluwer Academic, 2001.
- [15] S. Chen, L. Hanzo, and A. Livingstone, "MBER space-time decision feedback equalization assisted multiuser detection for multiple antenna aided SDMA systems," *IEEE Trans. Signal Processing*, vol. 54, no. 8, pp. 3090–3098, Aug. 2006.
- [16] P. Vandenameele, L. Van Der Perre, M. Engels, B. Gyselinckx, and H. De Man, "A combined OFDM/SDMA approach," *IEEE J. Selected Areas in Communications*, vol. 18, no. 11, pp. 2312–2321, Nov. 2000.
- [17] J. Zhang, L. Hanzo, and X. Mu, "Joint decision-directed channel and noise-variance estimation for MIMO OFDM/SDMA systems based on expectation-conditional maximization," *IEEE Trans. Vehicular Technology*, vol. 60, no. 5, pp. 2139–2151, June 2011.
- [18] L. Hanzo, O. R. Alamri, M. El-Hajjar, and N. Wu, *Near-Capacity Multi-Functional MIMO Systems: Sphere-Packing, Iterative Detection and Cooperation*. Chichester, UK: Wiley, 2009.
- [19] J. Zhang, S. Chen, X. Mu and L. Hanzo, "Turbo multi-user detection for OFDM/SDMA systems relying on differential evolution aided iterative channel estimation," *IEEE Trans. Communications*, vol. 60, no. 6, pp. 1621–1663, 2012.
- [20] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ: Prentice-Hall, 1993.
- [21] L. Hanzo, S. Ng, T. Keller, and W. Webb, *Quadrature Amplitude Modulation: from Basics to Adaptive Trellis-Coded Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems*. Chichester, UK: Wiley, 2004.
- [22] M. Jiang and L. Hanzo, "Multiuser MIMO-OFDM for next-generation wireless systems," *Proc. IEEE*, vol. 95, no. 7, pp. 1430–1469, July 2007.
- [23] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. 1997 IEEE Int. Conf. Systems, Man, and Cybernetics, and 1997 Computational Cybernetics and Simulation* (Orlando, Florida), Oct. 12-15, 1997, vol. 5, pp. 4104–4108.
- [24] M. A. Khanesar, M. Teshnehlab, and M. A. Shoorehdeli, "A novel binary particle swarm optimization," in *Proc. 2007 Mediterranean Conf. Control and Automation* (Athens, Greece), July 27-29, 2007, pp. 1–6.