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

Jiankang Zhang\textsuperscript{1}, Sheng Chen\textsuperscript{2,3}, Xiaomin Mu\textsuperscript{1}, and Lajos Hanzo\textsuperscript{2}
\textsuperscript{1}School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China
\textsuperscript{2}Electronics and Computer Science, University of Southampton, Southampton SO17 1BI, UK
\textsuperscript{3}Faculty 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 iterative 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) \cite{1, 2}, repeated weighted boosting search (RWBS) \cite{3, 4}, particle swarm optimisation (PSO) \cite{5, 6}, and differential evolution algorithm (DEA) \cite{7, 8}. There are many other popular EAs, e.g. ant colony optimisation \cite{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 obtaining 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 \cite{11}. We now know that many NP-hard problems can be turned into the so-called EA-easy class \cite{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) \cite{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 \cite{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 \cite{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 \cite{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 \cite{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 \cite{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 \cite{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 \textit{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 \( b^u \) 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 \( x^u[t] \). 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 \( y_q, 1 \leq q \leq Q \), 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]

\[
Y_q[s,k] = \sum_{u=1}^{U} H^u_q[s,k]X^u[s,k] + W_q[s,k],
\]

where \( H^u_q[s,k] \) 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 \( N_0 = 2\sigma_n^2 \). Let \( h^u_q[s] \in \mathbb{C}^{L_{cir} \times 1} \) 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

\[
H^u_q[s] = [H^u_q[s,1] H^u_q[s,2] \cdots H^u_q[s,K]]^T = Fh^u_q[s],
\]

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

\[
Y_q[s] = [Y_q[s,1] Y_q[s,2] \cdots Y_q[s,K]]^T, 1 \leq q \leq Q,
\]

and arrange each user’s data in a diagonal matrix

\[
X^u[s] = \text{diag}(X^u[s,1], \cdots, X^u[s,K]), 1 \leq u \leq U.
\]

Further define the CIR vector \( h_q[s] \in \mathbb{C}^{U L_{cir} \times 1} \) as

\[
h_q[s] = [(h^1_q[s])^T \ h^2_q[s])^T \ \cdots \ (h^U_q[s])^T]^T, 1 \leq q \leq Q.
\]

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

B. Optimisation in joint CE and MUD

The joint ML CE and MUD is defined as

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

where the cost function (CF) is given by

\[
J(h[s], X[s]) = \sum_{q=1}^{Q} \|Y_q[s] - X^T[s]Fh_q[s]\|^2, \tag{7}
\]

with the block-diagonal matrix \( F \in \mathbb{C}^{U K \times U L_{cir}} \) given by

\[
F = \text{diag}(F, F, \cdots F), \tag{8}
\]

while

\[
h_q[s] = [h^1_q[s] h^2_q[s] \cdots h^K_q[s]]^T, \tag{9}
\]

\[
X_q[s] = [X^1[s] X^2[s] \cdots X^K[s]]^T. \tag{10}
\]

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

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

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

in which the CF is expressed as

\[
J_{ce}(h_q[s]) = \|Y_q[s] - \hat{X}^T[s]Fh_q[s]\|^2. \tag{12}
\]

As \( h_q[s] \in \mathbb{C}^{U L_{cir} \times 1} \), the search space for each optimisation defined in (11) is a continuous \((2U L_{cir})\)-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

\[
Y[s,k] = [Y_1[s,k] \ Y_2[s,k] \ \cdots \ Y_Q[s,k]]^T, \tag{13}
\]

\[
X[s,k] = [X^1[s,k] \ X^2[s,k] \ \cdots \ X^U[s,k]]^T, \tag{14}
\]

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

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

where the MUD optimisation CF is expressed as

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

Since each \( X^u[s,k] \) contains \( A = \log_2 M \) bits, the bit-stream representation of \( X^u[s,k] \) is \( b^u[s,k] = [b^u_{1}[s,k] b^u_{2}[s,k] \cdots b^u_{Q}[s,k]]^T \), where \( b^u_{k}[s,k] \in \{0,1\} \). Thus, the bit-stream representation for \( X[s,k] \) is

\[
b[s,k] = [b^1[s,k] \ b^2[s,k] \ \cdots \ b^U[s,k]]^T. \tag{17}
\]

For notational simplification, the MUD CF is equivalently denoted as \( J_{mud}(b[s,k]) = J_{mud}(X[s,k]) \). 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), DCRWBS aided MUD (DCRWBS-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 \(P_t\) candidate solutions, representing the estimates of \(h_q\). The \(p_t\)-th individual of the population in the \(g\)-th generation is expressed as

\[
\hat{h}_{q,g,p_t} = [\hat{h}_{q,g,p_t,1} \cdots \hat{h}_{q,g,p_t,L_{cir}}]^T.
\]  

(18)

\(\{\hat{h}_{q,g,p_t}\}_{p_t=1}^{P_t}\) for \(g = 1\) are randomly generated in the search space \((-1 - J, +1 + J)^{U_{cir}}\), where \(J = \sqrt{-T}\) is the imaginary axis. The fitness of \(\hat{h}_{q,g,p_t}\) is related to its CF by

\[
f(\hat{h}_{q,g,p_t}) = J_{ce}^{-1} \hat{h}_{q,g,p_t} = \| Y - \hat{X}_{q,g,p_t} \|_2^{-2}.
\]  

(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

\[
\hat{h}_{q,g,mum} = [\hat{h}_{q,g,mum,1} \cdots \hat{h}_{q,g,mum,l-1}^u \hat{h}_{q,g,mum,l}^u \cdots \hat{h}_{q,g,mum,L_{cir}}^u]^T,
\]

\[
\hat{h}_{q,g,dad} = [\hat{h}_{q,g,dad,1} \cdots \hat{h}_{q,g,dad,l-1}^u \hat{h}_{q,g,dad,l}^u \cdots \hat{h}_{q,g,dad,L_{cir}}^u]^T,
\]

(20)

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

\[
\hat{h}_{q,g,os1} = [\hat{h}_{q,g,os1,1} \cdots \hat{h}_{q,g,os1,l-1}^u \hat{h}_{q,g,os1,l}^u \cdots \hat{h}_{q,g,os1,L_{cir}}^u]^T,
\]

\[
\hat{h}_{q,g,os2} = [\hat{h}_{q,g,os2,1} \cdots \hat{h}_{q,g,os2,l-1}^u \hat{h}_{q,g,os2,l}^u \cdots \hat{h}_{q,g,os2,L_{cir}}^u]^T,
\]

(21)

where for \(l^* \leq l \leq L_{cir}\)

\[
\hat{h}_{q,g,os1,1} = \hat{h}_{q,g,mum,1}^u - \beta(\hat{h}_{q,g,mum,1}^u - \hat{h}_{q,g,dad,1}^u),
\]

\[
\hat{h}_{q,g,os1,l} = \hat{h}_{q,g,os1,l-1}^u - \beta(\hat{h}_{q,g,os1,l-1}^u-1 - \hat{h}_{q,g,dad,l-1}^u),
\]

\[
\hat{h}_{q,g,os2,1} = \hat{h}_{q,g,mum,1}^u + \beta(\hat{h}_{q,g,mum,1}^u - \hat{h}_{q,g,dad,1}^u),
\]

\[
\hat{h}_{q,g,os2,l} = \hat{h}_{q,g,os2,l-1}^u + \beta(\hat{h}_{q,g,os2,l-1}^u - \hat{h}_{q,g,dad,l-1}^u),
\]

(22)

and \(\beta\) is a random value chosen in the range of \((0, 1)\). Assume that a gene \(\hat{h}_{q,g,p_t,l}\) of the individual \(\hat{h}_{q,g,p_t}\) is randomly chosen for the mutation, which is given by

\[
\hat{h}_{q,g,p_t,l} = \hat{h}_{q,g,p_t,l}^u + \gamma(\alpha_m + j\beta_m),
\]

where \(\alpha_m\) and \(\beta_m\) are randomly chosen within the range \((-1, 1)\), while \(\gamma\) is the mutation parameter. How many genes will mutate is governed by the mutation probability \(M_t\). The optimisation procedure is terminated when a pre-defined maximum number of generations, \(G_{max}\), is reached.

2) DBGA-MUD: The \(p_t\)-th individual of the population in the \(g\)-th generation is expressed as

\[
\hat{h}_{q,g,p_t} = [\hat{h}_{q,g,p_t,1} \cdots \hat{h}_{q,g,p_t,A} \cdots \hat{h}_{q,g,p_t,1}^U \cdots \hat{h}_{q,g,p_t,A}^U]^T,
\]

(24)

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

\[
f(\hat{b}_{q,g,p_t}) = J_{mud}^{-1} (\hat{b}_{q,g,p_t}) = \| Y - \hat{X}_{q,g,p_t} \|_2^{-2}.
\]  

(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 \(M_b\), \(\{M_bP_tUA\}\) bits are randomly selected from the total of \(P_t\)U\(A\) bits in the \(P_s\) individuals for mutation, where \([\bullet]\) denotes the integer floor.

B. RWBS for iterative CE and MUD

1) CRWBS-CE: The outer generation loop begins with initialising the population \(\{\hat{h}_{q,g,p_t}\}_{p_t=1}^{P_t}\) according to

\[
\hat{h}_{q,g,1} = \hat{h}_{q,g,1\text{best}},
\]

\[
\hat{h}_{q,g,p_t} = \hat{h}_{q,g,1\text{best}} + \gamma(\text{Gv - jGv}), 2 \leq p_t \leq P_s,
\]

(27)

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

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

\[
\bar{J}_{q,g,p_t} = J_{q,g,p_t}/\sum_{j=1}^{P_s} J_{q,g,j}, 1 \leq p_t \leq P_s,
\]

(28)

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

\[
\beta(t) = \frac{\eta(t)}{1 - \eta(t)} \text{ with } \eta(t) = \sum_{p_t=1}^{P_s} \bar{J}_{q,g,p_t} (t - 1) \bar{J}_{q,g,p_t},
\]

(29)

where \(t\) is the WBS index. Then the weights are adapted:

\[
\bar{J}_{p_t}(t) = \left(\frac{\delta_{p_t}(t - 1)}{(1 - \beta(t))^{\frac{1}{2}}} \right) J_{q,g,p_t}, \quad \beta(t) \leq 1, 1 \leq p_t \leq P_s,
\]

\[
\delta_{p_t}(t) = \frac{\beta(t)}{(1 - \beta(t))^{\frac{1}{2}}} J_{q,g,p_t}, \quad 1 \leq p_t \leq P_s.
\]

(30)

(31)

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

\[
\hat{h}_{q,g,P_t+1} = \sum_{p_t=1}^{P_s} \bar{J}_{p_t}(t) \hat{h}_{q,g,p_t} + (\hat{h}_{q,g,p_{worst}} - \hat{h}_{q,g,P_t+1}).
\]

(32)

\[
\hat{h}_{q,g,P_t+2} = \hat{h}_{q,g,p_{best}} + (\hat{h}_{q,g,p_{worst}} - \hat{h}_{q,g,P_t+1}).
\]

(33)
The inner WBS iterative procedure is terminated when the maximum number of WBS iterations $T_{	ext{inner}}$ has been reached. The outer generation procedure is stopped when the maximum number of generations $G_{\text{max}}$ has been reached.

2) **DBRWBS-MUD:** The outer generation loop begins with initialising the population \( \{ \tilde{b}_{g,p} \}_{p=1}^{P_s} \) as: set $\tilde{b}_{g,1} = \tilde{b}_{g-1,\text{best}}$, while the remaining $P_s - 1$ individuals $\tilde{b}_{g,p}$, $2 \leq p \leq P_s$, are generated by randomly mutating a certain percentage of the bits in $\tilde{b}_{g-1,\text{best}}$, the best individual found in the previous generation. The percentage of bits mutated are governed by the mutation probability $M_b$. The initial binary-valued $\tilde{b}_{0,\text{best}}$ 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 $\delta_{p}(t)$ with $1 \leq p \leq P_s$ have been acquired, define

\[
\Delta \delta_{p}(t) = \Delta \delta_{p-1}(t) + \delta_{p}(t), 1 \leq p \leq P_s.
\]

Then the four new individuals $\tilde{b}_{g,p+i,a}$ are updated according to the following procedure: $1 \leq a \leq A$ and $1 \leq u \leq U$, where $\tilde{b}_{g,p+i,a}$ is the best individual found so far, which defines its cognitive information, denoted as $\tilde{b}_{g,p+i,a}$. Every particle also knows the best position visited so far among the entire swarm, which defines the social information, denoted by $\tilde{b}_{g,p+i,a}$. The cognitive information $\tilde{b}_{g,p+i,a}$ and the social information $\tilde{b}_{g,p+i,a}$ are updated at each generation. The velocities and positions are updated according to

\[
v_{g+1,\text{p}} = \omega v_{g,\text{p}} + c_1 r a n d(0,1)(\tilde{b}_{g,\text{p}} - \tilde{b}_{g,\text{p}}),
\]

\[
\tilde{b}_{g,\text{p}} = \tilde{b}_{g,\text{p}} + v_{g,\text{p}},
\]

for $1 \leq p \leq P_s$, where $\omega$ 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 $G_{\text{max}}$ is reached.

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

\[
v_{g,p} = [v_{g,p,1}^1 \cdots v_{g,p,A}^1 \cdots v_{g,p,R}^1 \cdots v_{g,p,A}^U]^T,
\]

where $v_{g,p} \in \{0, 1\}^{U \times A}$ [23]. Associated with $\tilde{b}_{g,p}$ are the two bit-changing probability vectors given respectively by

\[
v_{g,p} = [v_{g,p,1}^1 \cdots v_{g,p,A}^1 \cdots v_{g,p,R}^1 \cdots v_{g,p,A}^U]^T.
\]

The initial DBPSO-MUD is similar to the CRWBS-CE, except for the population updating. Specifically, given that the $P_s$ individuals’ weights $\delta_{p}(t)$ with $1 \leq p \leq P_s$ have been acquired, define

\[
\Delta \delta_{p}(t) = \Delta \delta_{p-1}(t) + \delta_{p}(t), 1 \leq p \leq P_s.
\]

Then the four new individuals $\tilde{b}_{g,p+i,a}$ are updated according to the following procedure: $1 \leq a \leq A$ and $1 \leq u \leq U$, where $\tilde{b}_{g,p+i,a}$ is the best individual found so far, which defines its cognitive information, denoted as $\tilde{b}_{g,p+i,a}$. Every particle also knows the best position visited so far among the entire swarm, which defines the social information, denoted by $\tilde{b}_{g,p+i,a}$. The cognitive information $\tilde{b}_{g,p+i,a}$ and the social information $\tilde{b}_{g,p+i,a}$ are updated at each generation. The velocities and positions are updated according to

\[
v_{g+1,\text{p}} = \omega v_{g,\text{p}} + c_1 r a n d(0,1)(\tilde{b}_{g,\text{p}} - \tilde{b}_{g,\text{p}}),
\]

\[
\tilde{b}_{g,\text{p}} = \tilde{b}_{g,\text{p}} + v_{g,\text{p}},
\]

for $1 \leq p \leq P_s$, where $\omega$ 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 $G_{\text{max}}$ is reached.

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

\[
v_{g,p} = [v_{g,p,1}^1 \cdots v_{g,p,A}^1 \cdots v_{g,p,R}^1 \cdots v_{g,p,A}^U]^T,
\]

where $v_{g,p} \in \{0, 1\}^{U \times A}$ [23]. Associated with $\tilde{b}_{g,p}$ are the two bit-changing probability vectors given respectively by

\[
v_{g,p} = [v_{g,p,1}^1 \cdots v_{g,p,A}^1 \cdots v_{g,p,R}^1 \cdots v_{g,p,A}^U]^T.
\]

The initial DBPSO-MUD is similar to the CRWBS-CE, except for the population updating. Specifically, given that the $P_s$ individuals’ weights $\delta_{p}(t)$ with $1 \leq p \leq P_s$ have been acquired, define

\[
\Delta \delta_{p}(t) = \Delta \delta_{p-1}(t) + \delta_{p}(t), 1 \leq p \leq P_s.
\]

Then the four new individuals $\tilde{b}_{g,p+i,a}$ are updated according to the following procedure: $1 \leq a \leq A$ and $1 \leq u \leq U$, where $\tilde{b}_{g,p+i,a}$ is the best individual found so far, which defines its cognitive information, denoted as $\tilde{b}_{g,p+i,a}$. Every particle also knows the best position visited so far among the entire swarm, which defines the social information, denoted by $\tilde{b}_{g,p+i,a}$. The cognitive information $\tilde{b}_{g,p+i,a}$ and the social information $\tilde{b}_{g,p+i,a}$ are updated at each generation. The velocities and positions are updated according to

\[
v_{g+1,\text{p}} = \omega v_{g,\text{p}} + c_1 r a n d(0,1)(\tilde{b}_{g,\text{p}} - \tilde{b}_{g,\text{p}}),
\]

\[
\tilde{b}_{g,\text{p}} = \tilde{b}_{g,\text{p}} + v_{g,\text{p}},
\]

for $1 \leq p \leq P_s$, where $\omega$ 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 $G_{\text{max}}$ is reached.
is illustrated in Fig. 1, where the extrinsic information, which is randomly generated for each individual according to a Cauchy distribution with the location parameter $\mu_{C_r}$ and the scale parameter 0.1. Whether the target vector $\hat{b}_{g,p,s}$ or the trial vector $\hat{b}_{g,p,s}$ survives to the next generation is decided by

$$\hat{b}_{g,(g+1),p,s} = \begin{cases} \hat{b}_{g,p,s}, & J_{ce}(\hat{b}_{g,p,s}) \leq J_{ce}(\hat{b}_{g,p,s}), \\ \hat{b}_{g,p,s}, & \text{otherwise}, \end{cases}$$

where $J_{ce}$ is a function that calculates the channel error rate.

2) DBDBEA-MUD: The algorithm evolves the binary-valued individuals $b_{g,p,s}$ as defined in (24). Specifically, the mutant vector to the base vector $b_{g,i}$ is created according to

$$\tilde{b}_{g,i} = b_{g,i} \oplus (b_{g,best,r_1} \oplus b_{g,i}) \oplus (z_i \oplus (b_{g,r_2} \oplus b_{g,r_3})), \tag{51}$$

where $b_{g,best,r_1}$ is randomly chosen from the archive, $b_{g,r_2}$ and $b_{g,r_3}$, $r_2 \neq i$ and $r_3 \neq i$, are randomly selected from the current population, and $z_i$ 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 $\oplus$ denotes the bit-wise exclusive-AND. The trial vector $t_{g,p}$ is generated by

$$t_{g,p,i} = \begin{cases} \tilde{z}_{g,p,i}, & \text{rand}(0,1) \leq C_{r_p} \text{ or } j = j_{rand}, \\ 0_{g,p,i}, & \text{otherwise}, \end{cases} \tag{52}$$

where $C_{r_p} \in [0,1]$ is normally distributed for $\mu_{C_r}$ and standard deviation 0.1, while $j_{rand}$ is randomly chosen in the range of $\{1, 2, \ldots, P_s\}$. By mapping the binary-valued $t_{g,p,s}$ onto the corresponding $M$-QAM symbol vector $X_{g,p,s}$, the selection is defined by

$$\hat{b}_{g+1,p,s} = \begin{cases} \hat{b}_{g,p,s}, & J_{mud}(\hat{X}_{g,p,s}) \leq J_{mud}(\hat{X}_{g,p,s}), \\ \hat{b}_{g,p,s}, & \text{otherwise}, \end{cases} \tag{53}$$
4) Decision-directed channel estimator.

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

4.2) If \( \text{loop} < \text{I} \), where \( \text{I} \) defines the maximum number of joint CE and turbo MUD/decoder iterations, set \( \text{loop} = \text{loop} + 1 \) 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 \( C_{\text{ML-MUD}} \), and the complexity of the EA aided MUD, denoted as \( C_{\text{EA-MUD}} \), can be analysed. Similarly, given the CSI, the complexity of the optimal turbo ML-MUD/decoder, denoted by \( C_{\text{tML-turbo}} \), and the complexity of the EA assisted turbo MUD/decoder, denoted by \( C_{\text{tEA-turbo}} \), can be derived. We can further analyse the total complexity, denoted as \( C_{\text{total}} \), 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 \( 100 \) 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 \( f_{D} = 10^{-2} \). 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 \( \text{SNR} = E_{b}/N_{o} \) 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

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Type</th>
<th>Code rate</th>
<th>Constraint length</th>
<th>Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1/2</td>
<td>(90, 91)</td>
<td>(7, 5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Channel</th>
<th>Number of paths ( L_{\text{cir}} )</th>
<th>Delays</th>
<th>Average path gains</th>
<th>Taps: frame to frame</th>
<th>Taps: within frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( 0, 1, 2, 3 )</td>
<td></td>
<td>( 0, -5, -10, -15 ) (dB)</td>
<td>Complex white Gaussian</td>
<td>fading rate ( F_{D} = 10^{-2} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>MSS ( t )</th>
<th>Receiver antennas ( Q )</th>
<th>Modulation</th>
<th>Subcarriers ( K )</th>
<th>Cyclic prefix ( K_{p} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>16-6QAM</td>
<td>64</td>
<td>16</td>
</tr>
</tbody>
</table>

### Table II. The Algorithmic Parameters for the EA Assisted CE and MUD

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CML-CLE</td>
<td>Population size ( P_{s} )</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Selection ratio ( r_{s} )</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Mutation parameter ( \gamma )</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Mutation probability ( M_{s} )</td>
<td>0.2</td>
</tr>
</tbody>
</table>

| DBGA-MUD | Population size \( P_{s} \) | 100 |
|         | Selection ratio \( r_{s} \) | 0.5 |
|         | Mutation probability \( M_{s} \) | 0.15 |

| CRWBS-C | Population size \( P_{s} \) | 100 |
|         | Mutation parameter \( \gamma \) | 0.001 |
|         | Weighted boosting search \( T_{\text{wbs}} \) | 40 |

| DBRWS-MUD | Population size \( P_{s} \) | 100 |
|         | Mutation probability \( M_{s} \) | 0.5 |
|         | Weighted boosting search \( T_{\text{wbs}} \) | 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-C | 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 added channel estimators and the DBEA based MUD schemes over \( N_{\text{tot}} = 1000 \) 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. \( \text{Iter} = 1 \). 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 \( J_{\text{CE}}(\hat{h}_{q}, G_{\text{max, best}}) < 10^{-4} \) within the set upper limit for the number of CF evaluations \( N_{\text{lim}}^{\text{CF-EVs}} = P_{s} \cdot G_{\text{max}} \leq 100 \times 1000 \), where \( G_{\text{max}} \) 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 \( N_{\text{suc}} \), the number of failure runs, denoted as \( N_{\text{fail}} \), the total number of CF evaluations in the \( N_{\text{suc}} \) successful runs, defined by \( N_{\text{suc}}^{\text{CF-EVs}} \), and the total number of CF evaluations in the \( N_{\text{fail}} \) failure runs, defined by \( N_{\text{fail}}^{\text{CF-EVs}} \), using the following symbolic algorithm:

for \( \text{run} = 1 : N_{\text{tot}} \)

if \( \left( G_{\text{lim}}^{\text{max}} \leq G_{\text{lim}} \right) \) and \( \left( J_{\text{CE}}(\hat{h}_{q}, G_{\text{max, best}}) < 10^{-4} \right) \)

\( N_{\text{suc}} = N_{\text{suc}} + 1 \)

\( N_{\text{suc}}^{\text{CF-EVs}} = N_{\text{suc}}^{\text{CF-EVs}} + P_{s} \cdot G_{\text{lim}}^{\text{max}} \)

else

\( N_{\text{fail}} = N_{\text{fail}} + 1 \)

\( N_{\text{fail}}^{\text{CF-EVs}} = N_{\text{fail}}^{\text{CF-EVs}} + P_{s} \cdot G_{\text{lim}}^{\text{max}} \)

end if

end for

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

\[ N_{\text{tot}}^{\text{CF-EVs}} = \left( N_{\text{suc}}^{\text{CF-EVs}} + N_{\text{fail}}^{\text{CF-EVs}} \right) / N_{\text{tot}}, \]  

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

\[ N_{\text{suc}}^{\text{CF-EVs}} = N_{\text{suc}}^{\text{CF-EVs}} / N_{\text{suc}}. \]
Then the normalised average number of CF evaluations per run or the ratio of \( N_{CF-EVs}^{tot} \) to \( N_{CF-EVs}^{lim} \):

\[
\tilde{R}_{CF-EVs}^{tot} = \frac{N_{CF-EVs}^{tot}}{N_{CF-EVs}^{lim}} \quad (58)
\]

and the normalised average number of CF evaluations per successful run or the ratio of \( N_{CF-EVs}^{unc} \) to \( N_{CF-EVs}^{lim} \):

\[
\tilde{R}_{CF-EVs}^{unc} = \frac{N_{CF-EVs}^{unc}}{N_{CF-EVs}^{lim}} \quad (59)
\]

were used to quantify the efficiency of the EA aided CE scheme. The smaller \( \tilde{R}_{CF-EVs}^{tot} \) or \( \tilde{R}_{CF-EVs}^{unc} \), the more efficient the EA aided CE scheme. The reliability of the EA aided channel estimator was measured by the failure ratio

\[
R_{fail} = \frac{N_{fail}}{N_{tot}}. \quad (60)
\]

The smaller \( R_{fail} \), 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 worst CE candidate, having the failure rate \( R_{fail} \approx 7\% \) and requiring the average computational complexity of 90000 CF evaluations.

Fig. 2. Histograms of the efficiency and reliability measures, in terms of \( \tilde{R}_{CF-EVs}^{tot} \), \( \tilde{R}_{CF-EVs}^{unc} \), and \( R_{fail} \), for the four CEAs assisted CE schemes.

A similar procedure was carried out to investigate the efficiency and reliability of the four DBEA assisted MUDs by setting \( G_{max}^{lim} = 500 \) and \( N_{CF-EVs}^{lim} = M^U = 16^4 \). A successful detection run was confirmed, if \((G_{run}^{max} \leq G_{max}^{lim}) \). Otherwise, the run was declared a failure. Note that \( N_{CF-EVs}^{lim} = M^U \) was the number of CF evaluations required by the optimal full search ML MUD. Fig. 3 compares the efficiency and reliability of the four DBEAs assisted MUDs. Observe in Fig. 3 that the DBGA-MUD was the “winning” MUD candidate with a zero failure rate and requiring the 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 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 DBFPSO-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 \( N_{CE}^{CF-EVs} = 20000 \) (\( G_{max}^{max} = 200 \)), and the number of CF evaluations for EA aided MUD to \( N_{mud}^{CF-EVs} = 10000 \) (\( G_{max}^{max} = 100 \)). 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
schemes, and DEA aided iterative joint CE and turbo MUD/decoder schemes may be less efficient in comparison with the RWBS and the PSO and GA aided joint CE and turbo MUD/decoder solution of the idealised turbo ML-MUD/decoder. Iter

Recall that, given the CSI, a turbo MUD/decoder involves MUD. It can be seen from Table III that all the four EA

We therefore increased the number of the inner iterations between the EA aided MUD and the channel

eease of the idealised turbo ML-MUD/decoder, which is given with the perfect CSI, is really “unfair”. Even so, from Table III, we can 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 $N_{\text{MUD}}^{\text{turbo}}/N_{\text{MUD}}^{\text{CE}} = 10000$ compared with $N_{\text{MUD}}^{\text{turbo}}/N_{\text{MUD}}^{\text{CE}} = 20000$ needed by the GA and PSO. Note that an EA aided iterative joint CE and turbo MUD/decoder involves 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
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