This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Ant-Colony Based Near-ML Block Decoder in Asynchronous Cooperative MIMO Systems Using a Linear Dispersion Structure

Chong Xu and Hamid Gharavi
National Institute of Standards and Technology
Gaithersburg, USA
Email: {chong.xu, hamid.gharavi}@nist.gov

***Abstract*— The linear dispersion structure is employed to accommodate the dynamic topology of cooperative networks, as well as to achieve higher throughput than conventional space-time codes based on orthogonal designs. Asynchronous Cooperative Liner Dispersion Codes (ACLDC) are applied to achieve full diversity in the presence of asynchronous reception. To achieve a near ML performance with much lower decoding complexity, we present an Ant-Colony Optimization (ACO) based decoder. We demonstrate that the proposed decoder is able to provide the system with a near-ML performance at a significantly reduced complexity.**

I. Introduction

Transmit diversity [1]–[3] is a very significant characteristic which a Multiple-Input Multiple-Output (MIMO) communcation system possesses. Nevertheless, due to the size of a mobile phone, the 10λ distance between two antennas allowing each channel to be independent is difficult to fulfill. Against this background, the idea of 'cooperative diversity' is proposed in [4] to exploit independent nodes instead of multiple antennas as the sources to transmit signals, while achieving the same transmit diversity gain.

However, synchronization of signals transmitted from different nodes is difficult to achieve, due to the inequivalent propagation delay introduced by different channels. Hence, the disadvantage of asynchronous transmission - Inter-Symbol Interference (ISI) - needs to be taken into account when a certain MIMO technique is applied to provide transmission diversity gain.

In [5], the authors proposed a time-domain approach called the Asynchronous Cooperative Linear Dispersion Code (ACLDC) to deal with the ISI and employed a Maximum Likelihood (ML) algorithm to decode the ACLDC code blocks. Both the throughput of the system as well as the decoding complexity increase with the block size. This contradiction results in a throughput-complexity tradeoff that significantly restricts the improvement of the system throughput.

In this paper, we would propose a novel low-complexity Ant-Colony-Optimization (ACO) based algorithm to solve the optimization problem presented by the ACLDC decoder. This population-based approach has recently been applied to a large number of so-called Non-deterministic Polynomial (NP)-hard combinatorial optimization problems [6]–[8]. We will firstly apply the ACO algorithm to solve the optimization problem in the context of the ACLDC decoding issue. We will then demonstrate by simulation that the novel ACO-based ACLDC decoder is able to provide the system with near-ML performance at a significantly reduced complexity.

The outline of the paper is as follows. The model of the CLDC and ACLDC structure used in the cooperative wireless network will be detailed in Section II. Each significant components of the ACO-aided ACLDC block decoder will be interpreted in Section III. Then the main procedure included in the flow-chart of the ACO-based ACLDC block decoder will be presented in Section IV. Simulation results will be given in Section V. Finally, the conclusion will be provided in Section VI.

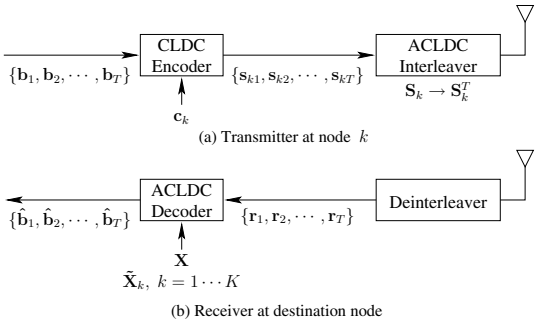

Fig. 1: (a) Block diagram of the ACLDC processer at the transmitter of node k; (b) Block diagram of the receiver at the destination node.

II. System Description

Fig. 1 shows the main components of the Asynchronous Cooperative Linear Codes (ACLDC)s scheme employed at each of the K transmitting nodes, after the broadcast stage is finished and the Q symbols belonging to the M nodes are all known to individual K nodes. The major difference between the ACLDC scheme and the CLDC scheme is the insertion of a guard interval between any two adjacent ACLDC blocks containing T number of CLDC codewords. ACLDC interleaver prevents the ISI from contaminating the symbols belonging to the same CLDC codeword. After deinterleavering, the signal received during the $\bar{T}$ symbol durations related to the tth CLDC codeword can be jointly quantified as

$$\begin{aligned}\mathbf{r}_t &= [\ r_{t1} \quad r_{t2} \quad \cdots \quad r_{t\bar{T}}\ ]^T \\ &= \bar{\mathbf{H}}\mathbf{X}\mathbf{v}_t + \sum_{k=2}^{K} \tilde{\mathbf{H}}_k \tilde{\mathbf{X}}_k \tilde{\mathbf{v}}_t + \mathbf{n}_t, \qquad t = 1, \ldots, T, \end{aligned} \tag{1}$$

where $\mathbf{n}_t = [n_{t1}, n_{t2}, \cdots, n_{t\bar{T}}]^T$denotes the AWGN encountered within the $\bar{T}$ symbol intervals associated with the tth CLDC block. In Eq. (1), the $(KQ \times 1)$-element uncoded symbol-vector, which was transmitted from the desired CLDC codeword t, can be formulated as

$$\mathbf{v}_t = [\mathbf{b}_t, \mathbf{b}_t, \cdots, \mathbf{b}_t]^T \tag{2}$$

with $\mathbf{b}_t = [b_{t1}, b_{t2}, \cdots, b_{tQ}]^T$ recording all the Q information symbols shared by the K transmitting nodes and that are encoded as the tth block of CLDC codeword. Furthermore, the $(\bar{T}K \times \bar{T}Q)$-

element matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_K \end{bmatrix} \tag{3}$$

contains all the K node-specific CLDC matrices $\mathbf{C}_1$, $\mathbf{C}_2$, ..., $\mathbf{C}_K$ with $\mathbf{0}$ being a $(\bar{T} \times Q)$-element all-zero matrix, Additionally, the $(\bar{T} \times \bar{T}K)$-element CIR matrix $\bar{\mathbf{H}}$ for the asynchronous cooperative system is formulated as:

$$\bar{\mathbf{H}} = \bar{\mathbf{h}}^T \otimes \mathbf{I}_{\bar{T}} = [\ h_1\mathbf{I}_{\bar{T}} \quad Ph_2\mathbf{I}_{\bar{T}} \quad \cdots \quad Ph_K\mathbf{I}_{\bar{T}}\ ], \tag{4}$$

where h_k, with $k = 1, 2, \ldots, K$ is a Rayleigh distributed complex number quantifying the Channel Impulse Response (CIR) from the kth transmitting node to the destination and P is the power of the symbol transmitted from the asynchronous node during the desired symbol duration.

In (1), the second term quantifies the ISI from the adjacent CLDC codes at each of the $(K-1)$ nodes that are asynchronous with the reception at the destination node. Hence, the $(2Q \times 1)$-element uncoded ISI symbol-vector, which was transmitted from the asynchronous node k during the previous $(t-1)$th and the next $(t+1)$th symbol duration, can be formulated as:

$$\tilde{\mathbf{v}}_t = [\ \mathbf{b}_{t-1} \quad \mathbf{b}_{t+1}\ ]^T. \quad \forall t = 1, 2, \ldots, T \tag{5}$$

where $\mathbf{b}_0$ and $\mathbf{b}_{T+1}$ are guard information symbols inserted between ACLDC blocks. The compound CLDC encoder matrix $\tilde{\mathbf{X}}_k$ for the ISI symbols from the kth node in Eq. (1) is made up only from the CLDC encoding matrix $\mathbf{C}_k$ of the kth node, yielding

$$\tilde{\mathbf{X}}_k = \begin{bmatrix} \mathbf{C}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k \end{bmatrix}. \tag{6}$$

The CIR associated with the ISI components of the signal-vector $\mathbf{r}_t$ received throughout the $\bar{T}$ symbol-intervals of the tth CLDC codeword from node k can be represented as the $(\bar{T} \times 2\bar{T})$-element CIR matrix $\tilde{\mathbf{H}}_k$, where

$$\tilde{\mathbf{H}}_k = [\ P_1h_k \quad P_2h_k\ ] \otimes \mathbf{I}_{\bar{T}} = [\ P_1h_k\mathbf{I}_{\bar{T}} \quad P_2h_k\mathbf{I}_{\bar{T}}\ ], \tag{7}$$

where P_1 and P_2 quantified the fraction of the power loss of the symbol transmitted from the asynchronous node leaked to the previous and the next symbol duration. Then the $(QT \times 1)$-element vector $\hat{\mathbf{b}}$ redeemed to contain all the QT number of symbols transmitted during the current block can be selected by exhaustively searching for the $(QT \times 1)$-element vector $\dot{\mathbf{b}}$ that satisfies the condition

$$\hat{\mathbf{b}} = \arg \min_{\dot{\mathbf{b}} \in \mathbb{B}^{(QT)}} \left\{ \sum_{t=1}^{T} \left\| \mathbf{r}_t - \bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{v}}_t - \sum_{k=2}^{K} \tilde{\mathbf{H}}_k \tilde{\mathbf{X}}_k \tilde{\dot{\mathbf{v}}}_t \right\|^2 \right\}, \tag{8}$$

where the full-set $\mathbb{B}^{(QT)}$ comprises of 2^{QT} possible $(QT \times 1)$-element BPSK modulated candidate solutions to minimize the objective function,

The ML algorithm will exhaustively evaluate the objective function (8) of all the 2^{QT} vectors in the full-set $\mathbb{B}^{(QT)}$, which impose an unaffordable massive complexity especially when the block-size T enhances. Conversely, the proposed near-optimal Ant-Colony-Optimization (ACO) algorithm will intelligently avoid searching through the entire candidate space $\mathbb{B}^{(QT)}$ while locating the solution for Eq. (8), More exactly, the major components and flow chart of the novel ACO aided block decoder algorithm will be detailed in Sections III and IV, respectively.

III. Components of ACO-Aided Block Decoder

The two important aspects of this novel ACO-aided block decoding algorithm compared to the classical ACO-aided Multi-User Detection (MUD) are, firstly, the calculation of the intrinsic affinity takes into account the inter-symbol interference from the interfering symbol-durations, as opposed to calculating the intrinsic affinity applied in the ACO-aided MUD under a simplified assumption that the symbol or symbol-group (when MIMO is applied) is transmitted independently without interference by symbol/symbol-groups transmitted by other users or during other symbol-intervals. The other aspect of this new ACO-aided ACLDC block decoder is the symbol-value probability-guided mutation that will provide more diversity to the search-pool, thereby enhancing the possibility of attaining the optimal solution.

The flow chart of the ACO-aided block decoder algorithm is depicted in Fig. 2. Below, an introduction of the ACO-aided ACLDC block decoding algorithm will begin by reviewing the major components that make up the ACO-aided ACLDC block decoder.

A. Route-Table

All the 2^{2T} number of $2T$-element solutions for the optimization problem described in Eq. (8) can be represented by the 2^{2T} number of T-piece route on the $(4 \times T)$-element route-table. When b_{1t} and b_{2t} are BPSK modulated with symbol values $\in \{+1, -1\}$, the four possible symbol-value combinations for the tth uncoded information symbol pair, b_{1t} and b_{2t} respectively, are represented by the four rows of the tth column on the route-table.

B. Symbol-Log Likelihood Function (LLF)

When the power loss is non-neglectable as a result of an increasingly longer delay, the effect of the inter-symbol interference may also need to be considered when the symbol-LLF is calculated. Therefore, by using only two symbols encoded as the current CLDC codeword, an additional two-symbol combination pair(s), e.g. encoded as the next and/or previous interference CLDC codeword, can also be considered. When the current CLDC codeword, together with the next (or previous) CLDC codeword is used, the number of rows of the symbol-LLF matrix recording all the trial symbol-LLF values will be $N_{\mathrm{r}} = 16$. When taking into account all three pairs of CLDC codewords (previous, current, and next), the number of rows in the matrix characterizing the symbol-LLF values will be $N_{\mathrm{r}} = 64$. More quantitatively, the uncoded information symbol-vector $\mathbf{u}_t$ considered for the tth CLDC codeword is made up of

$$\mathbf{u}_t = \begin{cases} \left[\mathbf{b}_t^T \ \ \mathbf{b}_{t+1}^T\right]^T & \text{when} \quad N_{\mathrm{r}} = 16 \\ \left[\mathbf{b}_{t-1}^T \ \ \mathbf{b}_t^T \ \ \mathbf{b}_{t+1}^T\right]^T & \text{when} \quad N_{\mathrm{r}} = 64 \end{cases} \tag{9}$$

We denote the four-symbol vector, when $N_{\mathrm{r}} = 16$ (six-symbol vector, when $N_{\mathrm{r}} = 64$) in the jth row of the $(N_{\mathrm{r}} \times T)$-dimensional symbol-LLF matrix as $\dot{\mathbf{u}}^{(j)}$, with $j = 1, 2, \ldots, N_{\mathrm{r}}$ when $t = 1, \ldots, T$. Then we have

$$\dot{\mathbf{u}}^{(j)} = \begin{cases} \left[\dot{u}_1^{(j)} \ \ \dot{u}_2^{(j)} \ \ \dot{u}_3^{(j)} \ \ \dot{u}_4^{(j)}\right], & \text{when} \quad N_{\mathrm{r}} = 16 \\ \left[\dot{u}_1^{(j)} \ \ \dot{u}_2^{(j)} \ \ \dot{u}_3^{(j)} \ \ \dot{u}_4^{(j)} \ \ \dot{u}_5^{(j)} \ \ \dot{u}_6^{(j)}\right], & \text{when} \quad N_{\mathrm{r}} = 64 \end{cases} \tag{10}$$

Hence, the symbol-LLF $\imath_{jt}$ associated with the symbol-vector $\dot{\mathbf{u}}^{(j)}$ in the jth row of the symbol-LLF matrix is a function of:

$$\imath_{jt} = f\left(\left\|\mathbf{r}_t - \bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{s}}^{(j)} - \tilde{\mathbf{H}}\tilde{\mathbf{X}}\tilde{\dot{\mathbf{s}}}^{(j)}\right\|^2\right), \tag{11}$$

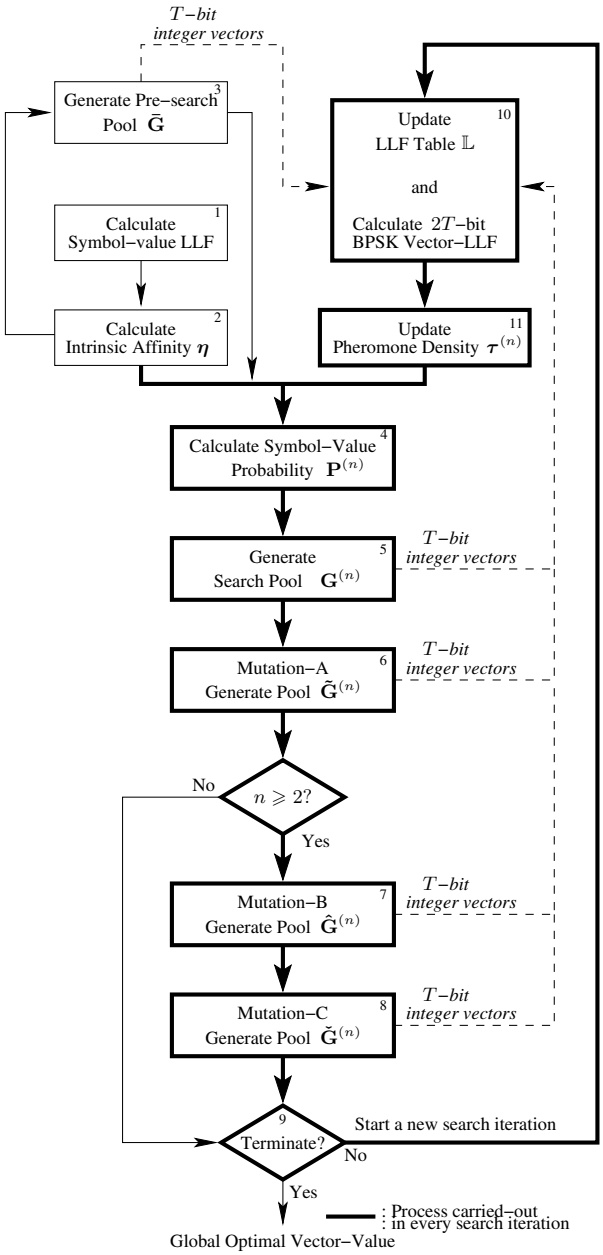

Fig. 2: Flow-chart of the ACO-based block decoder applied in the asynchronous cooperative MIMO systems using a linear dispersion structure and a block size of T symbols.

where we have

$$\dot{\mathbf{s}}^{(j)} = \begin{cases} \left[\dot{u}_1^{(j)}, \dot{u}_2^{(j)}, \dot{u}_1^{(j)}, \dot{u}_2^{(j)}\right], & \text{when } N_\mathrm{r} = 16 \\ \left[\dot{u}_3^{(j)}, \dot{u}_4^{(j)}, \dot{u}_3^{(j)}, \dot{u}_4^{(j)}\right], & \text{when } N_\mathrm{r} = 64 \end{cases} \tag{12}$$

and

$$\tilde{\dot{\mathbf{s}}}_t = \begin{cases} \left[0, 0, \dot{u}_3^{(j)}, \dot{u}_4^{(j)}\right], & \text{when } N_\mathrm{r} = 16 \\ \left[\dot{u}_1^{(j)}, \dot{u}_2^{(j)}, \dot{u}_5^{(j)}, \dot{u}_6^{(j)}\right], & \text{when } N_\mathrm{r} = 64 \end{cases} \tag{13}$$

In order to simplify the calculation of Eq. (11), we may omit the common terms, regardless of the value of j when the value of t is fixed. As a result, the symbol-LLF of $\mathfrak{l}_{jt}$ associated with the symbol-vector $\dot{\mathbf{u}}^{(j)}$ in the jth row and the tth column of the $(N_\mathrm{r} \times T)$-element symbol-LLF matrix can be ultimately defined as

$$\begin{aligned} \mathfrak{l}_{jt} = & 2\Re\left\{\left[\bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{s}}^{(j)} + \tilde{\mathbf{H}}\tilde{\mathbf{X}}\tilde{\dot{\mathbf{s}}}^{(j)}\right]^H \mathbf{r}_t\right\} - \\ & \left[\bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{s}}^{(j)} + \tilde{\mathbf{H}}\tilde{\mathbf{X}}\tilde{\dot{\mathbf{s}}}^{(j)}\right]^H \left[\bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{s}}^{(j)} + \tilde{\mathbf{H}}\tilde{\mathbf{X}}\tilde{\dot{\mathbf{s}}}^{(j)}\right], \end{aligned} \tag{14}$$

for all $j = 1, \ldots, N_\mathrm{r}$ when $t = 1, \ldots, T$. Additionally, $\Re\{a\}$ stands for the real part of the complex number a.

C. Intrinsic Affinity

Intrinsic affinity is a property of each of the $4T$ two-symbol vectors constituting the route-table, which determines the initial stage of the search pool. The sub intrinsic affinity $\hat{\eta}_{jt}$ associated with the jth symbol combination, for $j = 1, \ldots, N_\mathrm{r}$, is formualted as:

$$\hat{\eta}_{jt} = \frac{\sum_{i=1}^{N_\mathrm{r}} d_{it}}{\sum_{i=1}^{N_\mathrm{r}} d_{it} - d_{jt}}, \tag{15}$$

with $d_{jt} = 1/[1 + \exp(-\mathfrak{l}_{jt})]$, for $j = 1, \ldots, N_\mathrm{r}$. Hence, the intrinsic affinity associated with the symbol-vector at the ith row, with $i = 1, \ldots, 4$ in the tth column of the route table can be formulated as:

$$\eta_{it} = \begin{cases} \sum_{\forall j:[\dot{u}_1^{(j)}, \dot{u}_2^{(j)}] = f_{10\to 2}(i)} \hat{\eta}_{jt}^{\beta}, & \text{when } N_\mathrm{r} = 16 \\ \sum_{\forall j:[\dot{u}_3^{(j)}, \dot{u}_4^{(j)}] = f_{10\to 2}(i)} \hat{\eta}_{jt}^{\beta}, & \text{when } N_\mathrm{r} = 64 \end{cases} \tag{16}$$

where $f_{10\to 2}(i)$ stands for the equivalent two-bit expression of the binary number that is translated from the decimal number i. Furthermore, β in (16) is a weighting factor that quantifies the effect of the intrinsic affinity on the probabilities of choosing each candidate symbol-combination.

D. Vector-LLF

Given a certain route $\dot{\mathbf{b}}$ belonging to the full set $\mathbb{B}^{(2T)}$ containing all the $(2T \times 1)$-element BPSK vector, with the structure $\dot{\mathbf{b}} = [\dot{\mathbf{b}}_1 \;\; \dot{\mathbf{b}}_2 \;\; \cdots \;\; \dot{\mathbf{b}}_T^T]^T$, the vector-LLF can be finally quantified by:

$$\begin{aligned} \mathfrak{L}(\dot{\mathbf{b}}) = \sum_{t=1}^{T} \Bigg\{ & 2\Re\left[(\bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{v}}_t + \tilde{\mathbf{H}}\tilde{\mathbf{X}}\tilde{\dot{\mathbf{v}}}_t)^H \mathbf{r}_t\right] \\ & -(\bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{v}} + \tilde{\mathbf{H}}\tilde{\mathbf{X}}\tilde{\dot{\mathbf{v}}})^H (\bar{\mathbf{H}}\mathbf{X}\dot{\mathbf{v}} + \tilde{\mathbf{H}}\tilde{\mathbf{X}}\tilde{\dot{\mathbf{v}}}) \Bigg\}, \end{aligned} \tag{17}$$

where the desired signal vector $\dot{\mathbf{v}}$ and the interfered signal vector $\tilde{\dot{\mathbf{v}}}$ are defined in Eqs. (2) and (5).

IV. Flow-Chart of the ACO-Aided Block Decoder

The flow chart of the ACO-aided block decoder applied in the asynchronous cooperative MIMO systems using a linear dispersion structure is depicted in Fig. 2. Compared with the ACO-based MUD algorithm addressed in [9], [10], there are four new procedures included in the flow-chart of the ACO-aided block decoder, which are denoted as Steps 3, 6, 7 and 8 respectively in Fig. 2. The four novel components included in the block decoding procedure are: generating the pre-search pool $\bar{\mathbf{G}}$ before the first iteration begins, generating $\tilde{\mathbf{G}}^{(n)}$ according to mutation Rule A when $n \geqslant 1$, as well as $\hat{\mathbf{G}}^{(n)}$ and $\check{\mathbf{G}}^{(n)}$ according to mutation, Rule B and Rule C during the nth search iteration, when $n \geqslant 2$.

On the contrary, Steps 1, 2, 4, 5, 9, 10, and 11 are applied in the same way as they were applied in the ACO-MUD algorithms. Interest readers are referred to [9], [10].

A. Generating Pre-Search Pool - Step 3

A $(N_\alpha \times T)$-element pre-search pool $\bar{\mathbf{G}}$ is generated before the first iteration starts, as marked by Step 3 in Fig. 2. The pre-search pool is generated based on the sub intrinsic affinity given in Eq. (15). The number of columns of the pre-search pool $\bar{\mathbf{G}}$ is T and the number of rows depends on the number of different symbol-pairs between neighboring symbol durations. More exactly, during the T symbols, T-element integer vector $\hat{\mathbf{I}} = [\hat{I}_1, \hat{I}_2, \ldots, \hat{I}_T]$ can be generated based on the intrinsic-affinity matrix $\hat{\boldsymbol{\eta}}$ as given in Eq. (15). More exactly, $\hat{I}_t$ is the index of the row having the highest intrinsic affinity among N_r weighted intrinsic affinities $\hat{\eta}_{jt}^{\beta}$, $\forall\ j = 1, 2, \ldots, N_\mathrm{r}$.

Index $\hat{I}_t$ can be alternatively interpreted by two zero-to-three-ranging integers $\hat{I}_{t,1}$ and $\hat{I}_{t,2}$ when $N_\mathrm{r} = 16$, or by three zero-to-three-ranging integers $\hat{I}_{t,1}$, $\hat{I}_{t,2}$ and $\hat{I}_{t,3}$ when $N_\mathrm{r} = 64$. Ultimately, the pre-search pool $\bar{\mathbf{G}}$ is generated on a column-by-column basis from $t = 1$ to $t = T$. Before the element of the $t = 1$st column of the pre-search pool $\bar{\mathbf{G}}$ is generated, $\bar{\mathbf{G}}$ is a $(1 \times T)$-element all-zero row vector. When $t = 1$, the first and second column of $\bar{\mathbf{G}}$ will be assigned with $\hat{I}_{1,1}$ and $\hat{I}_{1,2}$ when $N_\mathrm{r} = 16$ or $\hat{I}_{1,2}$ and $\hat{I}_{1,3}$ when $N_\mathrm{r} = 64$. Then the value of t will be increased from 1 to 2.

Below the process given when the tth index $\hat{I}_t$ is considered with $t \geqslant 2$ is explained, Firstly, as long as $\hat{I}_{t-1,2} \neq \hat{I}_{t,1}$ when $N_\mathrm{r} = 16$ or $[\hat{I}_{t-1,2}, \hat{I}_{t-1,3}] \neq [\hat{I}_{t,1}, \hat{I}_{t,2}]$ when $N_\mathrm{r} = 64$, the following extra step will be taken before the value of the $(t+1)$th column of $\bar{\mathbf{G}}$ is assigned. More exactly, the existent rows of $\bar{\mathbf{G}}$ will be exactly copied and appended to the existent $\bar{\mathbf{G}}$. Consequentially, the number of rows of $\bar{\mathbf{G}}$ is doubled after this step. Then, the $(t-1)$th and the tth column of the newly generated counterpart of the original $\bar{\mathbf{G}}$ are assigned with $\hat{I}_{t,1}$ and $\hat{I}_{t,2}$ instead of $\hat{I}_{t-1,2}$ and $\hat{I}_{t-1,3}$.

B. Mutation-A: Mutation based on Most Likely Vector - Step 6

An additional pool $\tilde{\mathbf{G}}^{(n)}$ is generated in Step 6 as observed in Fig. 2. The mutation prototype of all vectors in $\tilde{\mathbf{G}}^{(n)}$ is an identical vector - $\hat{\mathbf{I}}^{(n)} = [\hat{I}_1^{(n)}, \hat{I}_2^{(n)}, \ldots, \hat{I}_T^{(n)}]$, corresponding to the T indexes of the highest probability in each column of the $(4 \times T)$-element probability matrix $\mathbf{P}^{(n)}$. Moreover, the first vector included in the additional pool $\tilde{\mathbf{G}}^{(n)}$ is also enforced to be $\hat{\mathbf{I}}^{(n)}$. Suppose the ith row-vector of $\tilde{\mathbf{G}}^{(n)}$ is denoted as $\tilde{\mathbf{g}}_i^{(n)}$ and $\tilde{g}_{it}^{(n)}$ represents the element on the ith row and the tth column of $\tilde{\mathbf{G}}^{(n)}$, we then have

$$\tilde{g}_{1t}^{(n)} = \arg \max_{i=1,2,3,4} P_{it}^{(n)}. \tag{18}$$

When $i \geqslant 2$, the ith row-vector $\tilde{\mathbf{g}}_i^{(n)}$ has only one element difference from the first row-vector $\tilde{\mathbf{g}}_1^{(n)}$. The different element $\tilde{g}_{it}^{(n)}$ has been mutated from $\tilde{g}_{1t}^{(n)}$ because the condition

$$P^{(n)}_{\tilde{g}_{1t}^{(n)} t} - P^{(n)}_{\tilde{g}_{it}^{(n)} t} < \tilde{\epsilon} \tag{19}$$

is satisfied. In Eq. (19), $\tilde{\epsilon}$ is a parameter set preceding the search iterations.

C. Mutation-B: Mutation based on Elite Vector - Step 7

The mutation process following Rule-B is denoted as Step 7 in Fig. 2 and is carried out only after the $n = 2$nd search iteration is activated. The mutation prototype of this step is the elite vector $\hat{\mathbf{G}}^{(n)}$ obtained when n search-iterations have been completed. The $(1 \times T)$-element integer vector $\dot{\mathbf{g}}^{(n)}$ is the elite vector that has the highest LLF among all the vectors generated so far. More particularly, each element $\dot{g}_t^{(n)}$ of $\dot{\mathbf{g}}^{(n)}$ for $t = 1, \ldots, T$ is an integer ranging from 1 to 4, which corresponds to the four different possible BPSK modulated symbol-pairs that may be transmitted during the tth symbol duration. One or two elements of $\dot{\mathbf{g}}^{(n)}$ will be mutated to generate a new vector. More exactly, the one-element mutated vector will be denoted as $\hat{\mathbf{g}}$ and the two-element mutated vector will be denoted as $\hat{\hat{\mathbf{g}}}$.

Given a fixed $t \in [1, T]$, the four rows in the tth column of $\mathbf{P}^{(n)}$ can be rearranged in an ascending order according to the value of $P_{it}^{(n)}$ for all $i = 1, 2, 3, 4$. If this rearrangement of rows is manipulated in all the columns of $\mathbf{P}^{(n)}$, the achieved newly sorted $(4 \times T)$-element probability matrix will be denoted as $\bar{\mathbf{P}}^{(n)}$. Then, the $(4 \times T)$-element integer matrix $\mathbf{I}^{(n)}$ with all the elements ranging from 1 to 4 records the original row-indexes of each element of $\bar{\mathbf{P}}^{(n)}$ in $\mathbf{P}^{(n)}$. More quantitatively, $I_{it}^{(n)}$ records the the ith highest probability in the tth column of $(4 \times T)$-element matrix $\mathbf{P}^{(n)}$.

Additionally, the $(1 \times T)$-element probability vector $\hat{\mathbf{P}}^{(n)}$ records the T highest probabilities in the T columns of $\mathbf{P}^{(n)}$. And the $(1 \times T)$-element probability vector $\tilde{\mathbf{P}}^{(n)}$ records the T second-highest probabilities in the T columns of $\mathbf{P}^{(n)}$.

The outer loop starts from $t = 1$ to $t = T$ with an increment of one at each step. The inner loop corresponds to the four rows and begins from $i = 1$ to $i = 4$ with an increment of one each time. The condition for the tth bit of the prototype vector $\dot{\mathbf{g}}^{(n)}$ to be mutated to I_{it} can be listed as follows.

$$\begin{aligned} &\left\{\dot{g}_t^{(n)} \neq I_{it}\right\} \cup \\ &\left\{(\bar{P}_{it}^{(n)} = \hat{P}_t^{(n)}) \parallel (\bar{P}_{it}^{(n)} = \tilde{P}_t^{(n)}) \parallel (\hat{P}_t^{(n)} - \bar{P}_{it}^{(n)} < \hat{\epsilon})\right\} \end{aligned} \tag{20}$$

where $\hat{\epsilon}$ is a pre-defined fractional parameter ranging in the area $[0, 1]$. When $t = t_1$, condition (20) can be alternatively broken into four conditions, which will be interpreted separately as follows. The first condition, namely the basic condition for the t_1th bit of the prototype vector $\dot{\mathbf{g}}^{(n)}$ to be mutated to I_{it_1} requires $\dot{g}_{t_1}^{(n)}$ not to be equivalent to I_{it_1}, i.e. $\dot{g}_{t_1}^{(n)} \neq I_{it_1}$. The second condition requires the probability associated with the combination at the I_{it_1}th row and the t_1th column of the route-table is the highest among the four candidate combinations, that is $\bar{P}_{it_1}^{(n)} = \hat{P}_{t_1}^{(n)}$. The third condition is satisfied when the probability associated with the combination at the I_{it_1}th row of the t_1th column of the route-table is the second highest, corresponding to $\bar{P}_{it_1}^{(n)} = \tilde{P}_{t_1}^{(n)}$. The fourth condition is fulfilled when the difference between the probability of the current combination at the t_1th column of the route-table and the highest probability at the t_1th column of the route-table is less than the bound $\hat{\epsilon}$, i.e. $\hat{P}_t^{(n)} - \bar{P}_{it}^{(n)} < \hat{\epsilon}$. According to (20), as long as the first condition in combination of any of the second, the third or the fourth condition is satisfied, the t_1th bit of $\dot{\mathbf{g}}^{(n)}$ will be mutated to I_{it_1}. The LLF value of the newly generated vector $\hat{\mathbf{g}}$ will be obtained. Furthermore, if the LLF of the newly generated vector $\hat{\mathbf{g}}$ satisfies the following condition, a second bit other than $\dot{g}_t^{(n)}$ of $\hat{\mathbf{g}}$ may be mutated and another new vector will be created. More exactly, given the LLF of the newly generated vector with one bit mutated from the prototype vector, $\mathfrak{L}(\hat{g})$, and the highest LLF of all the vectors generated during the nth iteration, $\dot{L}^{(n)}$, the condition can be detailed as

$$(\mathfrak{L}(\hat{g}) > \dot{L}^{(n)}) \parallel (\dot{L}^{(n)} - \mathfrak{L}(\hat{g}) < \hat{\delta}\dot{L}^{(n)}) \tag{21}$$

where $\hat{\delta}$ is a pre-defined scaling parameter with its value ranging in the area $[0, 1]$. (21) implies that the LLF of the newly generated vector $\hat{\mathbf{g}}$ should reach a certain LLF level to qualify for the prototype vector so that the second bit can be mutated.

The same outer and inner, loops corresponding to the column and row index of the second mutation bit, will be activated. However, the index of the outer loop t will skip t_1. And then, when the condition of (20) applies when $t = t_2$, the t_2th bit of $\hat{\mathbf{g}}$ will be mutated to $I_{it_2}^{(n)}$.

D. Mutation-C: Mutation based on Sub-Optimal Vectors - Step 8

Apart from the mutation process following Rule-B as detailed in Section IV-C, as long as the $n = 2$nd search iteration has been activated, another kind of mutation mechanism following Rule-C, denoted by Step 8 in Fig. 2 will be adopted to generate more dissimilar vectors under the guide of the probability matrix $\mathbf{P}^{(n)}$. The mutation prototype vectors of Step 8 are the several vectors having the top LLFs. Only one element will be mutated each time a new vector $\check{\mathbf{g}}$ is generated during this step. Except for the prototype vector of type-C mutation, the condition for tth bit of the prototype vector to be mutated to $I_{it_2}^{(n)}$ is identical with the one applied in Step 7, as listed in (20).

V. Simulation Results

In order to quantify the BER performance and complexity consumption, we applied the proposed ACO algorithm compared with the ML algorithm, to decode the ACLDC block in a cooperative network. The $(2, 2, 2)$ ACLDC system, with $Q = 2$, $\bar{T} = 2$ and $K = 2$ as detailed in [5], is employed as the system model for both results presented in this section. The parameters employed by the ACO algorithm take the same value as in [9], [10].

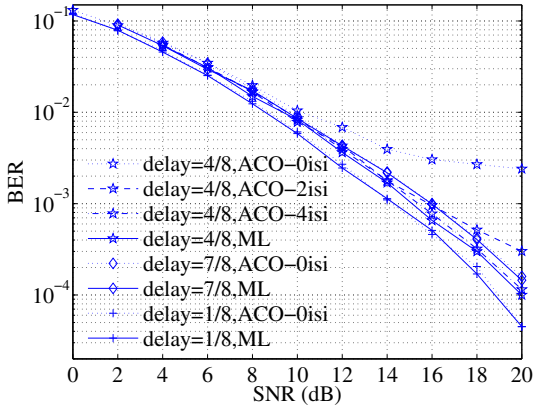

Fig. 3: Comparison of the BER performances achieved by the ACLDC $(2, 2, 2)$ system considered when $T = 6$ is the size of the block and ML, ACO decoding algorithm with $N_r = 4$, $N_r = 16$ and $N_r = 64$ are respectively employed for different delays.

As can be seen from Fig. 3, when delay is $1/8$ or $7/8$, the ACO algorithm denoted as 'ACO-0isi' corresponding to $N_r = 4$, which does not consider the effect of ISI when the symbol-LLF is calculated as discussed in Section III-B, is able to provide a near-ML BER performance. As most of the power of a symbol is focused on one symbol duration. However, when the delay is $4/8$, the ACO algorithm denoted as 'ACO-0isi' presents an error floor when the SNR is increased. By gradually considering two or four ISI symbols as the symbol-LLF is calculated, the BER performance is improved correspondingly. When the ISI encountered from both the previous and the next symbol duration is considered when the symbol-LLF is calculated, the BER performance as indicated by 'ACO-4isi', approaches that of the ML algorithm.

The number of times calculating the vector-LLF value according to Eq. (17) is regarded as the metric of quantifying the complexity of different decoding algorithms as shown in Fig. 4. Furthermore, when the block size increases as $T = 5, 10, 15, 20$, the corresponding throughput increases as 0.83, 0.91, 0.94, 0.95. However, as the throughput approaches its benchmark value, which is one for the $(2, 2, 2)$ CLDC system, the complexity of the ML algorithm is unaffordable.

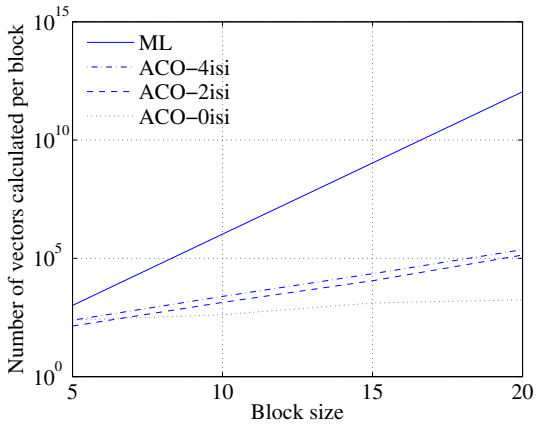

Fig. 4: Comparison of the number of vectors, whose LLF value have been calculated in an $(2, 2, 2)$ ACLDC block when block size equals to different values, when delay is $4/8$ and SNR is 20 dB.

Despite significant BER improvement achieved by 'ACO-2isi' and 'ACO-4isi' algorithms as shown in Fig. 3, the incremental complexity imposed by 'ACO-2isi' and 'ACO-4isi' algorithms compared with the 'ACO-0isi' is moderate.

VI. Conclusion

In this paper, the ACLDC structure was employed to cooperatively transmit information symbols from different nodes. A novel ACO-based decoding algorithm was proposed, against the ML algorithm. The simulation results have proved that, with moderate increment of the complexity, which is significantly lower as compared with the ML algorithm, the novel ACO algorithm is able to provide the decoder with gradually improved BER performance that is capable of approaching that of the ML algorithm.

References

[1] S. M. Alamouti, "A simple transmit diversity technique for wireless com- munications," IEEE J. Sel. Areas Commun., vol. 16, no. 8, pp. 14511458, Oct. 1998.

[2] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Spacetime block codes from orthogonal designs," IEEE Trans. Inf. Theory, vol. 45, no. 5, pp. 14561467, Jul. 1999.

[3] B. Hassibi and B. M. Hochwald, "High-rate codes that are linear in space and time," IEEE Trans. Inf. Theory, vol. 48, no. 7, pp. 18041824, Jul. 2002.

[4] J. N. Laneman and G. W. Wornell, "Distributed spacetime-coded protocols for exploiting cooperative diversity in wireless networks," IEEE Trans. Inf. Theory, vol. 49, no. 10, pp. 24152425, Oct. 2003.

[5] N. Wu and H. Gharavi, "Asynchronous Cooperative MIMO Systems Using a Linear Dispersion Structure," *IEEE Transactions on Vehicular Technology*, vol. 59, No. 2, pp. 779–787, February, 2010.

[6] A. Colorni, M. Dorigo and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the European Conference on Artificial Life*, 11-13, Dec, 1991, Paris, France, pp. 134–142.

[7] T. Stützle and H.-H. Hoos,"MAX-MIN Ant system." *Future Generation Computer Systems*, vol. 16, pp. 889–914, June,2000.

[8] E.-G. Talbi, O. Roux, C. Fonlupt and D. Robillard, "Parallel ant colonies for the quadratic assignment problem," *Future Generation Computer Systems*, vol. 17, pp. 441–449, Janurary, 2001.

[9] C. Xu, L.-L. Yang and L. Hanzo, "Ant-colony-based multiuser detection for MC DS-CDMA systems," in *Proceedings of the IEEE Vehicular Technology Conference*, 30, Sep-3, Oct, 2007, Baltimore, MD, USA, pp. 960–964.

[10] C. Xu, B. Hu, L.-L. Yang and L. Hanzo, "Ant-colony-based multiuser detection for multi-functional antenna array assisted MC DS-CDMA systems," *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 658-663, Jan, 2008.