A Reactive Tabu Search Based Equalizer for Severely Delay-Spread UWB MIMO-ISI Channels

N. Srinidhi, Saif K. Mohammed and A. Chockalingam
Department of ECE, Indian Institute of Science, Bangalore 560012, INDIA

***Abstract*—We propose a novel equalizer for ultrawideband (UWB) multiple-input multiple-output (MIMO) channels characterized by severe delay spreads. The proposed equalizer is based on *reactive tabu search* (RTS), which is a heuristic originally designed to obtain approximate solutions to combinatorial optimization problems. The proposed RTS equalizer is shown to perform increasingly better for increasing number of multipath components (MPC), and achieve near maximum likelihood (ML) performance for large number of MPCs at a much less complexity than that of the ML detector. The proposed RTS equalizer is shown to perform close to within 0.4 dB of single-input multiple-output AWGN performance at 10^{-3} uncoded BER on a severely delay-spread UWB MIMO channel with 48 equal-energy MPCs.**

Keywords *– Ultrawideband MIMO-ISI channels, equalization, severe delay spread, near-ML performance, reactive tabu search.*

I. INTRODUCTION

Wireless communication systems using ultrawideband (UWB) techniques typically use high transmission bandwidths to accommodate very high data rates [1]. Such UWB channels are highly frequency-selective, and are characterized by severe inter-symbol interference (ISI) due to large delay spreads [2]-[5]. The number of multipath components (MPC) in such channels in indoor/industrial environments has been observed to be of the order of several *tens to hundreds*; number of MPCs ranging from 12 to 120 are common in UWB channel models [2],[5]. Such large number of MPCs is often viewed as a source of severe ISI that hurts performance. On the other hand, these MPCs, if carefully exploited, can provide the opportunity to achieve increased time-diversity benefits [2]. But the complexity of receivers (e.g., RAKE [6]) that combine signal energies in all the MPCs can get prohibitive in complexity for large number of MPCs. Consequently, achieving near-optimum receiver/equalizer performance for channels with large number of MPCs at low complexities has been a challenging issue. We address this important issue in this paper. Particularly, we propose a novel equalizer algorithm, based on *reactive tabu search*, that achieves near maximum likelihood (ML) performance for large number of MPCs.

Tabu search (TS) is a heuristic originally designed to obtain approximate solutions to combinatorial optimization problems [7]-[10]. TS is increasingly applied in communication problems [11]-[13]. For e.g., in [11], design of constellation label maps to maximize asymptotic coding gain is formulated as a quadratic assignment problem, which is solved using a reactive TS (RTS) strategy [10]. RTS approach is shown to be effective in terms of bit error performance and efficient in terms of computational complexity in multiuser detection [12]. In [13], a fixed TS based multiple-input multiple output (MIMO) detection for V-BLAST is presented. In [14], RTS is used to decode large non-orthogonal space-time block codes in MIMO systems.

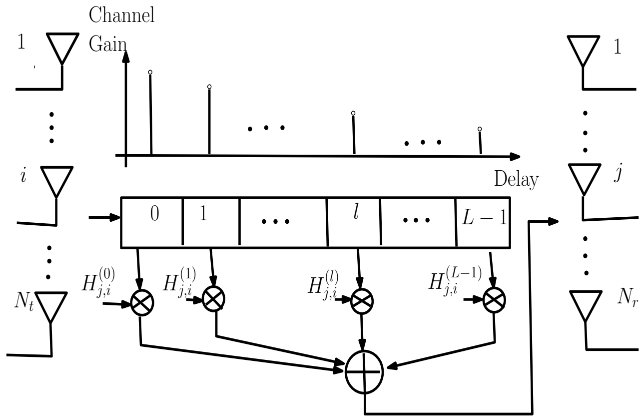

Fig. 1. MIMO-ISI Channel Model

In this paper, we propose a RTS based low-complexity equalizer algorithm suited for UWB MIMO channels characterized by severe delay spreads. We note that RTS for equalization in UWB MIMO channels has not been reported so far. Our simulation results show that the proposed RTS equalizer performs increasingly better for increasing number of MPCs, and achieve near-ML performance for large number of MPCs. It achieves this excellent performance at a much less complexity than that of the ML detector. For e.g., the RTS equalizer performs close to within 0.4 dB of single-input multiple-output (SIMO) AWGN performance for 4-QAM at 10^{-3} uncoded bit error rate (BER) on a severely delay-spread UWB MIMO channel with 48 equal-energy MPCs. Comparing the performance of the proposed RTS with those of other algorithms based on likelihood ascent search (LAS)/bit flipping [16],[17],[18] and factor graphs [15], we show that the proposed RTS equalizer achieves better performance due its inherent escape strategy from local minima.

The rest of the paper is organized as follows. In Section II we present the considered system model. In Section III, we present the proposed RTS equalizer algorithm. Section IV presents the simulation results of the BER performance of the proposed RTS equalizer. Conclusions are given in Section V.

II. SYSTEM MODEL

Consider a frequency-selective MIMO channel with N_t transmit and N_r receive antennas as shown in Fig. 1. Let L denote the number of MPCs. Data is transmitted in frames, where each frame has K data symbols preceded by a cyclic prefix (CP) of length L symbols, $K \geq L$. While CP avoids interframe interference, there will be ISI within the frame. Let $\widetilde{\mathbf{x}}_q \in \mathbb{A}^{N_t}$ be the transmitted symbol at time q, $0 \leq q \leq K-1$, where $\mathbb{A}$ is the transmit symbol alphabet, which is taken to be a QAM constellation. The received signal vector at time q can be written as

$$\mathbf{y}_q = \sum_{l=0}^{L-1} \widetilde{\mathbf{H}}_l \widetilde{\mathbf{x}}_{q-l} + \mathbf{w}_q, \qquad q = 0, \cdots, K-1, \quad (1)$$

where $\mathbf{y}_q \in \mathbb{C}^{N_r \times 1}$, $\widetilde{\mathbf{H}}_l \in \mathbb{C}^{N_r \times N_t}$ is the channel gain matrix for the lth MPC such that, $H_{j,i}^{(l)}$ denotes the entry on the jth row and ith column of the $\widetilde{\mathbf{H}}_l$ matrix, i.e., $H_{j,i}^{(l)}$ is the channel from ith transmit antenna to the jth receive antenna on the lth MPC. The entries of $\widetilde{\mathbf{H}}_l$ are assumed to be random with distribution $\mathcal{CN}(0,1)$. It is further assumed that $\widetilde{\mathbf{H}}_l$, $l = 0, \cdots, L-1$ do not change for one frame duration. $\mathbf{w}_q \in \mathbb{C}^{N_r \times 1}$ is the additive white Gaussian noise vector at time q, whose entries are independent, each with variance N_0. The CP will render the linearly convolving channel to a circularly convolving one, and so the channel will be multiplicative in frequency domain. Because of the CP, the received signal in frequency domain, for the ith frequency index ($0 \leq i \leq K-1$), can be written as

$$\mathbf{r}_i = \mathbf{G}_i \, \mathbf{u}_i + \mathbf{v}_i, \quad (2)$$

where $\mathbf{r}_i = \frac{1}{\sqrt{K}} \sum_{q=0}^{K-1} e^{\frac{-2\pi \mathbf{j} q i}{K}} \mathbf{y}_q$, $\mathbf{u}_i = \frac{1}{\sqrt{K}} \sum_{q=0}^{K-1} e^{\frac{-2\pi \mathbf{j} q i}{K}} \widetilde{\mathbf{x}}_q$, $\mathbf{v}_i = \frac{1}{\sqrt{K}} \sum_{q=0}^{K-1} e^{\frac{-2\pi \mathbf{j} q i}{K}} \mathbf{w}_q$, $\mathbf{G}_i = \sum_{q=0}^{L-1} e^{\frac{-2\pi \mathbf{j} q i}{K}} \widetilde{\mathbf{H}}_q$, and $\mathbf{j} = \sqrt{-1}$. Stacking the K vectors $\mathbf{r}_i$, $i = 0, \cdots, K-1$, we write

$$\mathbf{r} = \underbrace{\mathbf{G}\mathbf{F}}_{\triangleq \mathbf{H}_{eff}} \mathbf{x}_{eff} + \mathbf{v}_{eff}, \quad (3)$$

where

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_{K-1} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & & & \mathbf{0} \\ & \mathbf{G}_1 & & \\ & \mathbf{0} & \ddots & \\ & & & \mathbf{G}_{K-1} \end{bmatrix},$$

$$\mathbf{x}_{eff} = \begin{bmatrix} \widetilde{\mathbf{x}}_0 \\ \widetilde{\mathbf{x}}_1 \\ \vdots \\ \widetilde{\mathbf{x}}_{K-1} \end{bmatrix}, \quad \mathbf{v}_{eff} = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{K-1} \end{bmatrix},$$

$$\mathbf{F} = \frac{1}{\sqrt{K}} \begin{bmatrix} \rho_{0,0}\mathbf{I}_{N_t} & \rho_{1,0}\mathbf{I}_{N_t} & \cdots & \rho_{K-1,0}\mathbf{I}_{N_t} \\ \rho_{0,1}\mathbf{I}_{N_t} & \rho_{1,2}\mathbf{I}_{N_t} & \cdots & \rho_{K-1,1}\mathbf{I}_{N_t} \\ \vdots & \vdots & \cdots & \vdots \\ \rho_{0,K-1}\mathbf{I}_{N_t} & \rho_{1,K-1}\mathbf{I}_{N_t} & \cdots & \rho_{K-1,K-1}\mathbf{I}_{N_t} \end{bmatrix}$$
$$= \frac{1}{\sqrt{K}} \mathbf{D}_K \otimes \mathbf{I}_{N_t},$$

where $\rho_{q,i} = e^{\frac{-2\pi \mathbf{j} q i}{K}}$, $\mathbf{D}_K$ is the K-point DFT matrix and $\otimes$ denotes the Kronecker product. Separating real and imaginary parts of the various vectors/matrices as

$$\mathbf{H}_{eff} = \mathbf{H}_{eff,I} + \mathbf{j}\,\mathbf{H}_{eff,Q}, \quad \mathbf{r} = \mathbf{r}_I + \mathbf{j}\,\mathbf{r}_Q,$$

$$\mathbf{x}_{eff} = \mathbf{x}_{eff,I} + \mathbf{j}\,\mathbf{x}_{eff,Q}, \quad \mathbf{v}_{eff} = \mathbf{v}_{eff,I} + \mathbf{j}\,\mathbf{v}_{eff,Q},$$

and defining

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_{eff,I} & -\mathbf{H}_{eff,Q} \\ \mathbf{H}_{eff,Q} & \mathbf{H}_{eff,I} \end{pmatrix}, \quad \mathbf{r} = [\mathbf{r}_I^T \;\; \mathbf{r}_Q^T]^T,$$

$$\mathbf{x} = [\mathbf{x}_{eff,I}^T \;\; \mathbf{x}_{eff,Q}^T]^T, \quad \mathbf{v} = [\mathbf{v}_{eff,I}^T \;\; \mathbf{v}_{eff,Q}^T]^T,$$

the received signal vector can be written in real form as

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{v}, \quad (4)$$

where $\mathbf{r} \in \mathbb{R}^{2N_rK}$, $\mathbf{H} \in \mathbb{R}^{2N_rK \times 2N_tK}$, and $\mathbf{v} \in \mathbb{R}^{2N_rK}$. The entries of the $\mathbf{x}$ vector will be from a real alphabet $\mathbb{A}_R = \{a_j, j = 1, 2, \cdots, M\}$, where $a_j = 2j - 1 - M$. The ML solution is obtained by minimizing the ML cost

$$\phi(\mathbf{x}) = \|\mathbf{r} - \mathbf{H}\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{H}^T\mathbf{H}\mathbf{x} - 2\mathbf{x}^T\mathbf{H}^T\mathbf{r} \quad (5)$$

over all $\mathbf{x} \in \mathbb{A}_R^{2N_tK}$. Though optimum, the computational complexity of the ML solution increases exponentially with N_tK. This necessitates the use of suboptimum solutions. It has been found that many modern meta-heuristic algorithms give near-ML performance at a much reduced complexity [19]. In the next section, we propose one such heuristic based on reactive tabu search. Let $S = 2N_tK$ and $R = 2N_rK$.

III. Proposed RTS Equalizer Algorithm

The RTS algorithm is an iterative local neighborhood search algorithm. It starts with an initial solution vector, defines a neighborhood around it (i.e., defines a set of neighboring vectors based on a neighborhood criteria), and moves to the best vector among the neighboring vectors (even if the best neighboring vector is worse, in terms of likelihood, than the current solution vector; this allows the algorithm to escape from local minima). This process is continued for a certain number of iterations, after which the algorithm is terminated and the best among all the solution vectors in all the iterations is declared as the final solution vector. In defining the neighborhood of the solution vector in a given iteration, the algorithm attempts to avoid cycling by making the moves to solution vectors of the past few iterations as tabu (i.e., prohibits these moves), which ensures efficient search of the solution space. The number of these past iterations is parametrized as the 'tabu period.' The search is referred to as fixed tabu search if the tabu period is kept constant. If the tabu period is dynamically changed (e.g., increase the tabu period if more repetitions of the solution vectors are observed in the search path), then the search is called reactive tabu search. We consider reactive tabu search in this paper because of its robustness (choice of a good fixed tabu period can be tedious).

Neighborhood Definition: Let M denote the cardinality of $\mathbb{A}_R$. Let $a_q \in \mathbb{A}_R$, $q = 1, \cdots, M$. Define a set $\mathcal{N}(a_q)$ as a fixed subset of $\mathbb{A}_R \backslash a_q$, which we refer to as the *symbol neighborhood* of a_q. We choose the cardinality of this set to be the same for all a_q, $q = 1, \cdots, M$; i.e., we take $|\mathcal{N}(a_q)| = N$, $\forall q$. Note that the maximum and minimum values of N are $M-1$ and 1, respectively. We choose the symbol neighborhood based on Euclidean distance, i.e., for a given symbol, those N symbols which are the nearest will form its neighborhood; the nearest symbol will be the first neighbor, the next nearest symbol will be the second neighbor, and so on. For e.g., $\mathbb{A}_R = \{-3,-1,1,3\}$ for 4-PAM, and choosing N to be 2, $\mathcal{N}(-3) = \{-1,1\}$, $\mathcal{N}(-1) = \{-3,1\}$, $\mathcal{N}(1) = \{-1,3\}$, $\mathcal{N}(3) = \{1,-1\}$ are possible symbol neighborhoods. If $N = 3$, $\mathcal{N}(-3) = \{-1,1,3\}$, $\mathcal{N}(-1) = \{-3,1,3\}$, $\mathcal{N}(1) = \{-1,3,-3\}$, $\mathcal{N}(3) = \{1,-1,-3\}$ are possible symbol neighborhoods. Let $w_v(a_q)$, $v = 1,..,N$ denote the vth element in $\mathcal{N}(a_q)$; i.e., we say $w_v(a_q)$ is the vth symbol neighbor of a_q.

Let $\mathbf{x}^{(m)} = [x_1^{(m)}\, x_2^{(m)} \cdots x_S^{(m)}]$ denote the data vector belonging to the solution space, in the mth iteration, where $x_i^{(m)} = a_q$, $q \in \{1,\cdots,M\}$. We refer to the vector

$$\mathbf{z}^{(m)}(u,v) \;=\; \big[z_1^{(m)}(u,v)\;\; z_2^{(m)}(u,v) \cdots z_S^{(m)}(u,v)\big], \quad (6)$$

as the (u,v)th *vector neighbor* $\big($or simply the (u,v)th neighbor$\big)$ of $\mathbf{x}^{(m)}$, $u = 1,\cdots,S$, $v = 1,\cdots,N$, if $i)$ $\mathbf{x}^{(m)}$ differs from $\mathbf{z}^{(m)}(u,v)$ in the uth coordinate, and $ii)$ the uth element of $\mathbf{z}^{(m)}(u,v)$ is the vth symbol neighbor of $x_u^{(m)}$. That is,

$$z_i^{(m)}(u,v) \;=\; \begin{cases} x_i^{(m)} & \text{for } i \neq u \\ w_v(x_u^{(m)}) & \text{for } i = u. \end{cases} \quad (7)$$

So we will have SN vectors which differ from a given vector in the solution space in only one coordinate. These SN vectors form the neighborhood of the given vector. We note that neighborhood definition based on *bit-flipping* [16] is a special case of the above neighborhood definition for $M = 2$, $N = 1$. An operation on $\mathbf{x}^{(m)}$ which gives $\mathbf{x}^{(m+1)}$ belonging to the vector neighborhood of $\mathbf{x}^{(m)}$ is called a *move*. The algorithm is said to execute a move (u,v) if $\mathbf{x}^{(m+1)} = \mathbf{z}^{(m)}(u,v)$. We note that the number of candidates to be considered for a move in any one iteration is SN. Also, the overall number of 'distinct' moves possible is SMN, which is the cardinality of the union of all moves from all M^S possible solution vectors. The tabu value of a move, which is a non-negative integer, means that the move cannot be considered for that many number of subsequent iterations, unless certain conditions are satisfied.

Tabu Matrix: A *tabu_matrix* of size $SM \times N$ is the matrix whose entries denote the tabu values of moves. The (r,s)th entry of the *tabu_matrix* corresponds to the move (u,v) from $\mathbf{x}^{(m)}$ when $u = \lfloor \frac{r-1}{M} \rfloor + 1$, $v = s$ and $x_u^{(m)} = a_q$, where $q = mod(r-1, M) + 1$.

Algorithm: Let $\mathbf{g}^{(m)}$ be the vector which has the least ML cost found till the mth iteration of the algorithm. Let l_{rep} be the average length (in number of iterations) between two successive occurrences of the same solution vector (repetitions), at the end of an iteration. Tabu period, P, a dynamic non-negative integer parameter, is defined. If a move is marked as tabu in an iteration, it will remain as tabu for P subsequent iterations. A binary flag, $lflag \in \{0,1\}$, is used to indicate whether the algorithm has reached a local minima in a given iteration or not; this flag is used in the evaluation of the stopping criterion of the algorithm. The algorithm starts with an initial solution vector $\mathbf{x}^{(0)}$, which, for e.g., could be the MMSE or MF output vector. Set $\mathbf{g}^{(0)} = \mathbf{x}^{(0)}$, $l_{rep} = 0$, and $P = P_0$. All the entries of the *tabu_matrix* are set to zero. Define $\mathbf{r}_{mf} \stackrel{\triangle}{=} \mathbf{H}^T\mathbf{r}$, and $\mathbf{R} \stackrel{\triangle}{=} \mathbf{H}^T\mathbf{H}$. Compute $\mathbf{r}_{mf}$ and $\mathbf{R}$. The following steps 1) to 3) are performed in each iteration. Consider mth iteration in the algorithm, $m \geq 0$.

Step 1): Initialize $lflag = 0$. Define $\mathbf{f}^{(m)} \stackrel{\triangle}{=} \mathbf{R}\mathbf{x}^{(m)} - \mathbf{r}_{mf}$. Let $\mathbf{e}^{(m)}(u,v) = \mathbf{z}^{(m)}(u,v) - \mathbf{x}^{(m)}$. The ML costs of the SN neighbors of $\mathbf{x}^{(m)}$, namely, $\mathbf{z}^{(m)}(u,v)$, $u = 1,\cdots,S$, $v = 1,\cdots,N$, are computed as

$$\begin{aligned}
\phi(\mathbf{z}^{(m)}(u,v)) \;&=\; \left(\mathbf{x}^{(m)} + \mathbf{e}^{(m)}(u,v)\right)^T \mathbf{R} \left(\mathbf{x}^{(m)} + \mathbf{e}^{(m)}(u,v)\right) \\
&\quad -2\left(\left(\mathbf{x}^{(m)} + \mathbf{e}^{(m)}(u,v)\right)^T \mathbf{r}_{mf}\right) \\
&= \phi(\mathbf{x}^{(m)}) + 2\left(\left(\mathbf{e}^{(m)}(u,v)\right)^T \mathbf{R}\,\mathbf{x}^{(m)}\right) \\
&\quad + \left(\mathbf{e}^{(m)}(u,v)\right)^T \mathbf{R}\,\mathbf{e}^{(m)}(u,v) - 2\left(\left(\mathbf{e}^{(m)}(u,v)\right)^T \mathbf{r}_{mf}\right) \\
&= \phi(\mathbf{x}^{(m)}) + \underbrace{2\left(e_u^{(m)}(u,v)\, f_u^{(m)}\right) + \left(e_u^{(m)}(u,v)\right)^2 \mathbf{R}_{u,u}}_{\stackrel{\triangle}{=} C\left(e_u^{(m)}(u,v)\right)}, \quad (8)
\end{aligned}$$

where $e_u^{(m)}(u,v)$ is the uth element of $\mathbf{e}^{(m)}(u,v)$, $f_u^{(m)}$ is uth element of $\mathbf{f}^{(m)}$, and $\mathbf{R}_{u,u}$ is the (u,u)th element of $\mathbf{R}$. $\phi(\mathbf{x}^{(m)})$ on the RHS in (8) can be dropped since it will not affect the cost minimization. Let

$$(u_1,v_1) \;=\; \underset{u,\,v}{\arg\min}\; C\big(e_u^{(m)}(u,v)\big). \quad (9)$$

The move (u_1,v_1) is accepted if any one of the following two conditions is satisfied:

$$\begin{aligned}
&i) \quad \phi(\mathbf{z}^{(m)}(u_1,v_1)) < \phi(\mathbf{g}^{(m)}) \\
&ii) \quad tabu_matrix\left((u_1-1)M + q, v_1\right) = 0 \quad (10)
\end{aligned}$$

where $q : x_{u_1}^{(m)} = a_q \in \mathbb{A}_R$. If move (u_1,v_1) is accepted, make

$$\mathbf{x}^{(m+1)} \;=\; \mathbf{x}^{(m)} + \mathbf{e}^{(m)}(u_1,v_1). \quad (11)$$

If move (u_1,v_1) is not accepted (i.e., neither of conditions $i)$ and $ii)$ is satisfied), find (u_2,v_2) such that

$$(u_2,v_2) \;=\; \underset{u,v\,:\,u \neq u_1, v \neq v_1}{\arg\min}\; C\big(e_u^{(m)}(u,v)\big), \quad (12)$$

and check for acceptance of the (u_2,v_2) move. If this also cannot be accepted, repeat the procedure for (u_3,v_3), and so on. If all the SN moves are tabu, then all the *tabu_matrix* entries are decremented by the minimum value in the *tabu_matrix*; this goes on till one of the moves becomes permissible. Let (u',v') be the index of the neighbor with the minimum cost for which the move is permitted. The variables q', q'', v'' are implicitly defined by $x_{u'}^{(m)} = a_{q'} = w_{v''}(x_{u'}^{(m+1)})$, and $x_{u'}^{(m+1)} = a_{q''}$, where $a_{q'}, a_{q''} \in \mathbb{A}_R$. It is noted that in this *Step 1* of the algorithm, essentially the best permissible vector neighbor is chosen as the solution vector for the next iteration.

Step 2): The new solution vector obtained from *Step 1* is checked for repetition. For the system model in (4), repetition can be checked by comparing the ML costs of the solutions in the previous iterations. If there is a repetition, the length of the repetition from the previous occurrence is found, the average length, l_{rep}, is updated, and the tabu period P is modified as $P = P + 1$. If the number of iterations elapsed since the last change of the value of P exceeds βl_{rep}, for a fixed $\beta > 0$,

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

make $P = P - 1$. The minimum value of P, however, will be 1. Note that this step of making $P = P - 1$, if executed, also qualifies as the one which changed P. After a move (u', v') is accepted, if $\phi(\mathbf{x}^{(m+1)}) < \phi(\mathbf{g}^{(m)})$, make

$$\begin{aligned} \textit{tabu\_matrix}\,((u'-1)M + q', v') &= 0; \\ \textit{tabu\_matrix}\,((u'-1)M + q'', v'') &= 0; \\ \mathbf{g}^{(m+1)} = \mathbf{x}^{(m+1)}; & \end{aligned}$$

else

$$\begin{aligned} \textit{tabu\_matrix}\,((u'-1)M + q', v') &= P + 1; \\ \textit{tabu\_matrix}\,((u'-1)M + q'', v'') &= P + 1; \\ lflag = 1, \quad \mathbf{g}^{(m+1)} = \mathbf{g}^{(m)}. & \end{aligned}$$

It is noted that this *Step 2* of the algorithm implements the 'reactive' part in the search, by dynamically changing P.

Step 3): Update the entries of the *tabu_matrix* as

$$\textit{tabu\_matrix}\,(r, s) \;=\; \max\{\textit{tabu\_matrix}\,(r, s) - 1, 0\}, \quad (13)$$

for $r = 1, \cdots, SM$, $s = 1, \cdots, N$. $\mathbf{f}^{(m)}$ is updated as

$$\mathbf{f}^{(m+1)} \;=\; \mathbf{f}^{(m)} + e_{u'}^{(m)}(u', v')\mathbf{R}_{u'}, \quad (14)$$

where $\mathbf{R}_{u'}$ is the u'th column of $\mathbf{R}$. The algorithm terminates in *Step 3* if the following stopping criterion is satisfied, else it goes back to *Step 1*.

Stopping criterion: The algorithm can be stopped based on a fixed number of iterations. Though convergence can be slow at low SNRs, it can be fast at moderate to high SNRs. So rather than fixing a large number of iterations to stop the algorithm irrespective of the SNR, we use an efficient stopping criterion which makes use of the knowledge of the best ML cost in a given iteration, as follows. Since the ML criterion is to minimize $\|\mathbf{Hx} - \mathbf{r}\|^2$, the minimum value of the objective function $\mathbf{x}^T\mathbf{H}^T\mathbf{Hx} - 2\mathbf{x}^T\mathbf{H}^T\mathbf{r}$, is always greater than $-\mathbf{r}^T\mathbf{r}$. We stop the algorithm when the least ML cost achieved in an iteration is within certain range of the global minimum, which is $-\mathbf{r}^T\mathbf{r}$. We stop the algorithm in the mth iteration, only if $lflag = 1$ and certain conditions are satisfied as given below. The algorithm is stopped if the condition

$$\frac{|\phi(\mathbf{g}^{(m)}) - (-\mathbf{r}^T\mathbf{r})|}{|-\mathbf{r}^T\mathbf{r}|} \;<\; \alpha_1 \quad (15)$$

is met with at least *min_iter* iterations being completed to make sure the search algorithm has 'settled.' The bound is gradually relaxed as the number of iterations increase and the algorithm is terminated when

$$\frac{|\phi(\mathbf{g}^{(m)}) - (-\mathbf{r}^T\mathbf{r})|}{|-\mathbf{r}^T\mathbf{r}|} \;<\; m\alpha_2. \quad (16)$$

In (15) and (16), α_1 and α_2 are positive constants. In addition, we terminate the algorithm whenever the number of repetitions of solutions exceeds *max_rep*. Also, the maximum number of iterations is set to *max_iter*.

Initial vector using FD-MMSE Equalizer: The detected symbol vector obtained using frequency domain (FD) MMSE equalization can be used as the initial vector to the RTS algorithm. The FD-MMSE equalizer on the ith frequency employs MMSE nulling as

$$\widehat{\mathbf{u}}_i \;=\; \Big(\mathbf{G}_i^H\mathbf{G}_i + \frac{N_0}{E_s}\mathbf{I}_{N_t}\Big)^{-1}\mathbf{G}_i^H\mathbf{r}_i, \quad 0 \leq i \leq K-1, \quad (17)$$

where E_s is the average energy of a transmitted symbol. The $\widehat{\mathbf{u}}_i$'s are transformed back to time domain using K-point IDFT, to obtain an estimate of the transmitted symbol vector, as

$$\widehat{\widehat{\mathbf{x}}}_q \;=\; \frac{1}{\sqrt{K}}\sum_{i=0}^{K-1} e^{\frac{2\pi \mathbf{j} q i}{K}}\widehat{\mathbf{u}}_i, \quad 0 \leq q \leq K-1, \quad (18)$$

which are used to form the initial vector to the RTS algorithm.

A. RTS algorithm versus LAS algorithm in [17]

It is noted that the LAS algorithm presented in [17] is also a local neighborhood search based algorithm, where the basic definition of neighborhood is the same as in RTS. However, LAS differs from RTS in the following aspects: i) while the definition of neighborhood is static in LAS for all iterations, in RTS, in addition to the basic neighborhood definition, there is also a dynamic aspect to the neighborhood definition by way of prohibiting certain vectors from being included in the neighbor list (implemented through repetition checks/tabu period), and ii) while LAS gets trapped in the local minima that it first encounters and declares this minima to be the final solution vector, RTS can potentially find better minimas because of the escape strategy embedded in the algorithm (by way of allowing to pick and move to the best neighbor even if that neighbor has a lesser likelihood than the current solution vector).

It is further noted that a general version of LAS (termed as *multistage LAS – MLAS* [17]) executes a different escape mechanism when it encounters a local minima, by changing the neighborhood definition: it considers vectors which differ in two or more coordinates (as opposed to only one in the basic neighborhood definition) as neighbors. On escaping from a local minima, the algorithm reverts back to the basic neighborhood definition till the next local minima is encountered and stops when no escape from a local minima is possible. Since the performance gain of MLAS compared to LAS is found to be small, in this paper we limit our comparison of RTS with only LAS. Our simulation results in the following section show that RTS performs better than LAS.

IV. Simulation Results

We evaluated the BER performance of the proposed RTS equalizer in a 4×4 MIMO V-BLAST system with 4-QAM modulation (the RTS algorithm, however, is applicable to general QAM), as a function of average E_b/N_0 per receive antenna, through simulations. We have assumed uniform power delay profile (i.e., all the L paths are assumed to be of equal energy). We evaluated the performance for various number of delay paths, L, and frame sizes, K, keeping L/K constant. As seen in Figs. 2 and 3, the RTS equalizer performs increasingly closer to the optimum performance for increasing number of dimensions (i.e., increasing L, K). The FD-MMSE equalizer output is taken to be the initial vector. The following RTS parameters are used in the simulations:

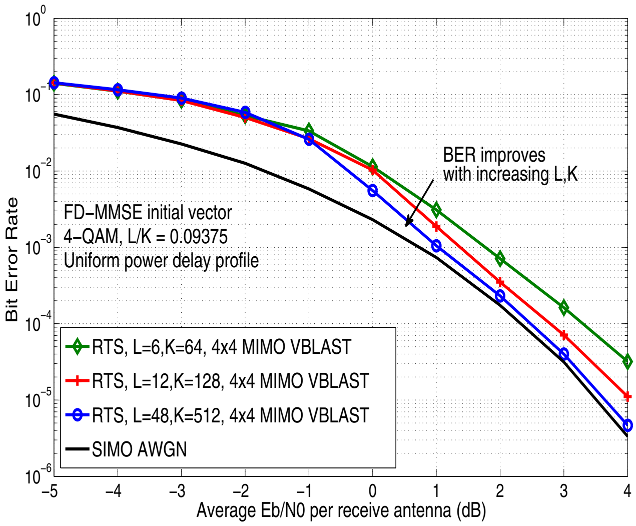

Fig. 2. Uncoded BER performance of the RTS equalizer in a 4×4 MIMO V-BLAST system with 4-QAM for different number of delay paths (L) and frame sizes (K), keeping L/K constant. Uniform power delay profile.

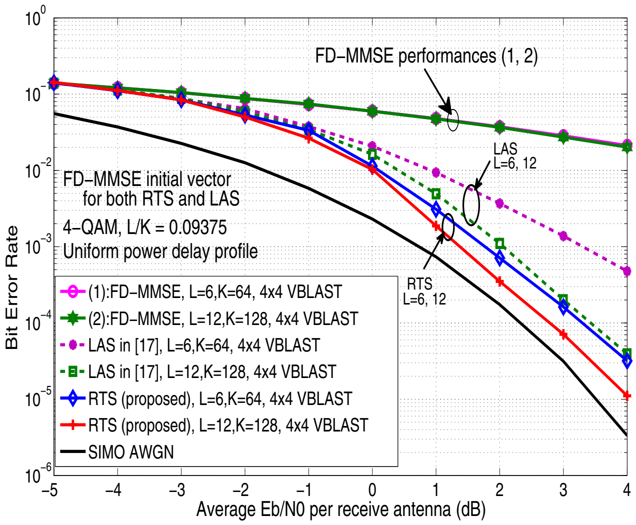

Fig. 3. Comparison of the BER performance of the proposed RTS equalizer with those of LAS equalizer and and FD-MMSE equalizer in a 4×4 MIMO V-BLAST system with 4-QAM for different number of delay paths (L) and frame sizes (K), keeping L/K constant. Uniform power delay profile.

$P_0 = 2; \beta = 1; \alpha_1 = 0.03; max\_rep = 75; min\_iter = 30$. For $K = 64$ and 128, $max\_iter = 300$ and $\alpha_2 = 0.00075$. For $K = 512$, $max\_iter = 500$ and $\alpha_2 = 0.0004$.

In Fig. 2, we plot the uncoded BER of the RTS equalizer for $(L = 6, K = 64)$, $(L = 12, K = 128)$, and $(L = 48, K = 512)$, $L/K = 0.09375$. We have also plotted the performance of SIMO AWGN (for the same N_r) which is a good lower bound on the best detector performance. It can be seen that the performance of RTS equalizer improves as L, K are increased. For e.g., the performance gap between the RTS equalizer and the SIMO AWGN performance is only about 0.4 dB for $L = 48$ MPCs at an uncoded BER of 10^{-3}. Even with only $L = 12$ MPCs, the RTS equalizer performance is only about 0.8 dB away from SIMO AWGN performance at 10^{-3} BER.

Next, in Fig. 3, we present a BER performance comparison of the equalizers based on RTS and LAS [17] with FD-MMSE initial vector for both. The BER performance of the FD-MMSE equalizer (without any search) is also plotted for comparison; it can be seen that it performs significantly poor. However, the subsequent search operations carried out in RTS and LAS result in significantly improved performance for increasing L, K. Both RTS and LAS show large dimension effect (i.e., BER improves for increasing L, K, keeping L/K constant). However, for a given L, RTS performs better than LAS. For example, at a BER of 10^{-3}, the performance of LAS is worse by about 1.5 dB compared to that of RTS for $L = 6, K = 64$. For $L = 12, K = 128$, the performance gap is about 0.8 dB in favor of RTS. This improvement can be attributed to the escape mechanism in RTS from local minimas. Also, this better performance is achieved with only a small increase in complexity (shown in Fig. 4). For even larger dimensions ($L = 48$, $K = 512$), we have found that RTS and LAS perform almost the same. We note that the BER of the factor graph based equalizer in [15] showed significant error floors for $L = 12$.

Complexity of RTS versus LAS: The complexity of the RTS algorithm comprises of 3 main parts: $i)$ FD-MMSE initial vector computation, $ii)$ computation of $\mathbf{H}^T\mathbf{H}$, and $iii)$ the search operation. Complexity parts $i)$ and $ii)$ are the same for both RTS and LAS, and only the search complexity part $iii)$ differs. Figure 4 shows the RTS and LAS complexity plots as a function of the frame size, K, for 4×4 V-BLAST, 4-QAM, average E_b/N_0=0 dB, $L/K = 0.09375$, and uniform delay profile. The per-symbol complexity of FD-MMSE computation is $O(KN_t + N_t^2)$. The per-symbol complexity of $\mathbf{H}^T\mathbf{H}$ computation is $O(K^2N_t)$. The per-symbol search complexities for RTS and LAS are obtained by simulations, and are plotted in Fig. 4. It can be seen that $a)$ the search complexity of RTS is slightly higher than that of LAS for small K, $b)$ for large K, search complexities of RTS and LAS are almost the same, and c) the per-symbol search complexity is only $O(K)$ for both RTS and LAS for large K (i.e., search complexity plots run parallel to c_2K line). It can be also seen from Fig. 4 that the *overall* per-symbol complexities of RTS and LAS are almost the same; this is because, for small N_t (e.g., $N_t = 4$), the $O(K^2N_t)$ complexity of $\mathbf{H}^T\mathbf{H}$ is the dominant one, which is same in both RTS and LAS. In summary, the proposed RTS performs better than (for small/moderate L, K) or the same as (for large L, K) LAS with about the same complexity.

V. Conclusions

We presented a reactive tabu search (RTS) based equalization algorithm that achieved near-ML bit error performance at low complexities in severely delay-spread UWB MIMO-ISI channels. The proposed RTS algorithm can be employed for low-complexity equalization in UWB wireless systems/standards that are characterized by large number of multipath components. We have assumed perfect channel knowledge of all the paths. Joint channel estimation and equalization in severely delay-spread UWB MIMO channels can be investigated as further extension to this work.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

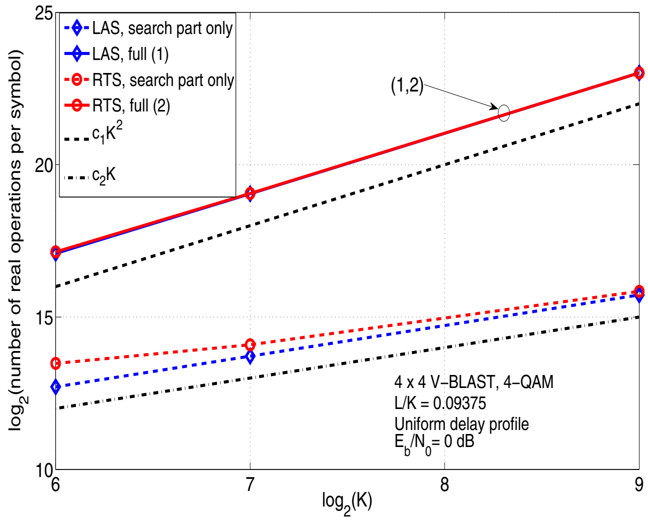

Fig. 4. Complexity comparison between the proposed RTS algorithm and the LAS algorithm. 4×4 V-BLAST, 4-QAM, $L/K = 0.09375$.

REFERENCES

[1] X. Shen, M. Guizani, R. C. Qiu, and T. Le-Ngoc, *Ultra-wideband Wireless Communications and Networks,* John Wiley & Sons, 2006.

[2] A. F. Molisch, J. R. Foerster, M. Pendergrass, "Channel models for ultrawideband personal area networks," *IEEE Wireless Commun.*, vol. 10, no. 6, pp. 1421, December 2003.

[3] A. F. Molisch, "Ultrawideband propagation channels - Theory, measurement, and modeling," *IEEE Trans. Veh. Tech.,* vol. 54, no. 5, pp. 1528-1545, September 2005.

[4] J. Karedal, S. Wyne, P. Almers, F. Tufvesson, and A. F. Molisch, "Statistical analysis of the UWB channel in an industrial environment," *Proc. IEEE VTC'2004-Fall,*, pp. 81-85, September 2004.

[5] R. Saadane and A. M. Hayar, "DRB1.3 third report on UWB channel models," http://www.eurecom.fr/util/publidownload.fr.htm?id=2112, Newcom, November 2006.

[6] J. G. Proakis, *Digital Communications*, 4th Ed., Mc-Graw Hill, 2001.

[7] F. Glover, "Tabu Search - Part I," *ORSA Journal of Computing,* vol. 1, no. 3, Summer 1989, pp. 190-206.

[8] F. Glover, "Tabu Search - Part II," *ORSA Journal of Computing,* vol. 2, no. 1, Winter 1990, pp. 4-32.

[9] F. Glover and M. Laguna, "Tabu Search - Modern Heuristic Techniques for Combinatorial Problems," Colin R. Reeves Ed., 70-150, Blackwell Scientific Publications, Oxford, 1993.

[10] R. Battiti and G. Tecchiolli, "The reactive tabu search," *ORSA Journal on Computing*, no. 2, pp. 126-140, 1994.

[11] Y. Huang and J. A. Ritcey, "Improved 16-QAM constellation labeling for BI-STCM-ID with the Alamouti scheme," *IEEE Commun. Letters,* vol. 9, no. 2, pp. 157-159, February 2005.

[12] P. H. Tan and L. K. Rasmussen, "Multiuser detection in CDMA - A comparison of relaxations, exact, and heuristic search methods," *IEEE Trans. Wireless Commun.,* vol. 3, no. 5, pp. 1802-1809, September 2004.

[13] H. Zhao, H. Long, and W. Wang, "Tabu search detection for MIMO systems," *Proc. IEEE PIMRC'2007*, Athens, September 2007.

[14] N. Srinidhi, S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "Low-complexity near-ML decoding of large non-orthogonal STBCs using reactive tabu search," *Proc. IEEE ISIT'2009*, Seoul, June 2009.

[15] T. Wo, P. A. Hoeher, "A simple iterative Gaussian detector for severely delay-spread MIMO channels," *Proc. IEEE ICC'2007*, June 2007.

[16] K. Vishnu Vardhan, S. K. Mohammed, A. Chockalingam, B. S. Rajan, "A low-complexity detector for large MIMO systems and multicarrier CDMA systems," *IEEE JSAC Spl. Iss. on Multiuser Detection, for Adv. Commun. Systems and Networks*, vol. 26, no. 3, pp. 473-485, April 2008.

[17] S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "A low-complexity near-ML performance achieving algorithm for large MIMO detection," *Proc. IEEE ISIT'2008*, Toronto, July 2008.

[18] T. Koike, "Bit-flipping equalizer and ML search space analysis in ultrawideband MIMO channels," *Proc. IEEE GLOBECOM'2008*, November-December 2008.

[19] V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and G. D. Smith, Editors, *Modern Heuristic Search Methods*, John Wiley & Sons, 1996.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.