in Proc. IEEE SPAWC-12
Cesme, Turkey, June 2012

LOW-COMPLEXITY DETECTION FOR LARGE MIMO SYSTEMS USING PARTIAL ML DETECTION AND GENETIC PROGRAMMING

Pavol Svac, Florian Meyer, Erwin Riegler, and Franz Hlawatsch

Institute of Telecommunications, Vienna University of Technology, Austria (florian.meyer@tuwien.ac.at)

ABSTRACT

We propose a low-complexity detector for multiple-input multiple-output (MIMO) systems using BPSK or QAM constellations. The detector operates at the bit level and is especially advantageous for large MIMO systems. It consists of three stages performing *partial ML detection*, *generation of soft values*, and *soft-input genetic optimization*. For the last stage, we present a genetic programming algorithm that uses the soft values computed by the second stage. Simulation results demonstrate that for large systems, our detector can outperform state-of-the-art methods, and its complexity scales roughly cubically with the system dimension.

1. INTRODUCTION

The generic multiple-input/multiple-output (MIMO) system model,

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}\,, \tag{1}$$

is relevant to important communication scenarios including multi-antenna wireless systems [1], orthogonal frequency-division multiplexing systems [2], and code-division multiple access systems [2]. In (1), $\mathbf{s} \in \mathcal{S}^{N_t}$ is the transmitted symbol vector, whose elements are from some signal constellation $\mathcal{S}$; $\mathbf{y} \in \mathbb{C}^{N_r}$ is the received vector; $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the channel matrix; and $\mathbf{n} \in \mathbb{C}^{N_r}$ is a noise vector. Here, we consider the problem of detecting the transmit vector $\mathbf{s}$ given the receive vector $\mathbf{y}$. We assume that the channel matrix $\mathbf{H}$ is known and the elements of the noise vector $\mathbf{n}$ are independent and identically distributed (iid) circularly symmetric complex Gaussian, i.e., $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I}_{N_r})$, where $\mathbf{I}_{N_r}$ denotes the $N_r \times N_r$ identity matrix.

For equally likely transmit vectors $\mathbf{s} \in \mathcal{S}^{N_t}$, the optimum detector is the maximum-likelihood (ML) detector given by [1]

$$\hat{\mathbf{s}}_{\mathrm{ML}}(\mathbf{y}) \;=\; \underset{\mathbf{s}\in\mathcal{S}^{N_t}}{\arg\min} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2. \tag{2}$$

Unfortunately, ML detection is infeasible for large MIMO systems because its complexity grows exponentially with N_t; this also holds for the efficient sphere-decoding algorithm [3,4]. Suboptimum methods include detection based on linear equalization [5], decision-feedback equalization (or nulling-and-canceling) [5–7], lattice reduction [8], semidefinite relaxation [9], and heuristic optimization methods such as genetic algorithms (see [10] and references therein).

Recently, large MIMO systems with several tens of antennas have attracted increased attention due to their high capacity [11, 12]. Suboptimum detection methods specially designed for large MIMO systems include local search algorithms such as likelihood ascent search [11, 13] and reactive tabu search [12]. Furthermore, a factor graph based belief propagation algorithm using a Gaussian approximation of the interference has been proposed for large MIMO systems [14].

This work was supported by the Austrian Science Fund (FWF) under Award S10603 (Statistical Inference) within the National Research Network SISE and by the WWTF under Award ICT10-066 (NOWIRE).

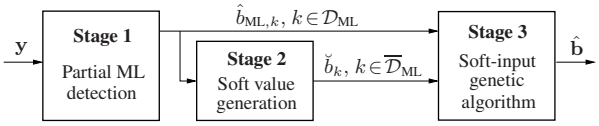

Fig. 1. Block diagram of the proposed MIMO detector.

In this paper, we propose a suboptimum detector for large MIMO systems using a BPSK or QAM constellation $\mathcal{S}$. The detector operates at the bit level (as defined in [15]) and consists of three stages as shown in Fig. 1. The first stage, *partial ML detection*, calculates some bits of the ML solution (2) by means of the iterative algorithm described in [16]. In the second stage, *soft values* for the undetected bits are generated. In the third stage, the undetected bits are detected by means of a *soft-input genetic algorithm*, which is a novel soft-input version of a genetic algorithm that was presented in [17] for solving large binary quadratic programming problems. Our main contributions are (i) the overall detector structure combining a partial ML detection stage with suboptimum detection based on soft values, (ii) the generation of soft values using previously detected bits, and (iii) an adaptation of a genetic algorithm to soft inputs and the use of the resulting algorithm for MIMO detection.

For large MIMO systems, the proposed MIMO detector is demonstrated through simulation to outperform detectors based on nulling-and-canceling, semidefinite relaxation, and likelihood ascent search. The computational complexity scales roughly cubically with the system dimension and constellation size.

This paper is organized as follows. In Section 2, we review the bit-level partial ML detection method of [16] (Stage 1), and we describe the generation of soft values for the undetected bits (Stage 2). In Section 3, the soft-input genetic algorithm (Stage 3) is developed. In Section 4, we assess the performance of the proposed MIMO detector through simulation results.

2. PARTIAL ML DETECTION AND GENERATION OF SOFT VALUES

For any QAM constellation $\mathcal{S}$, where $|\mathcal{S}| = 2^B$ with an even integer $B = \log_2 |\mathcal{S}|$, there is a vector $\mathbf{v} \in \mathbb{C}^B$ such that every symbol $s \in \mathcal{S}$ can be represented in a unique way as [15]

$$s = \mathbf{v}^T \tilde{\mathbf{b}}(s)\,, \tag{3}$$

with a bit vector $\tilde{\mathbf{b}}(s) \in \{-1, 1\}^B$. For example, $\mathbf{v} = (1 \quad j)^T$ for $|\mathcal{S}| = 4$ and $\mathbf{v} = (2 \quad 1 \quad 2j \quad j)^T$ for $|\mathcal{S}| = 16$. It should be noted that for $|\mathcal{S}| \geq 16$, the binary representation defined by (3) is not a Gray mapping. A bit-level representation of the MIMO system (1) is then given by [15]

$$\mathbf{y} = \mathbf{A}\mathbf{b} + \mathbf{n}\,, \tag{4}$$

where $\mathbf{A} \triangleq \mathbf{H} \otimes \mathbf{v}^T \in \mathbb{C}^{N_r \times BN_t}$ is an equivalent channel matrix (here, $\otimes$ denotes the Kronecker product), $\mathbf{b} \equiv \mathbf{b}(\mathbf{s}) \triangleq \big(\tilde{\mathbf{b}}^T(s_1) \cdots \tilde{\mathbf{b}}^T(s_{N_t})\big)^T \in \{-1,1\}^{BN_t}$ is the binary representation of the transmit symbol vector $\mathbf{s}$, and the noise vector $\mathbf{n}$ is identical to that in (1). The ML detector (2) can be equivalently formulated as

$$\hat{\mathbf{b}}_{\text{ML}}(\mathbf{y}) \;=\; \underset{\mathbf{b}\in\{-1,1\}^{BN_t}}{\arg\min} \|\mathbf{y} - \mathbf{A}\mathbf{b}\|^2 . \tag{5}$$

We note that, even though BPSK is not a QAM constellation, it is a trivial special case of (4) and (5), with $B = 1$, $\mathbf{v} = (1)$, $\mathbf{b} = \mathbf{s}$, and $\mathbf{A} = \mathbf{H}$.

2.1. Partial ML Detection

The first stage of the proposed MIMO detector performs "partial ML detection" in the sense that it computes *some* elements $\hat{b}_{\text{ML},k}$ of $\hat{\mathbf{b}}_{\text{ML}}$ in (5). This is done by using the efficient algorithm proposed in [16], which will now be reviewed. In what follows, let $\mathbf{z} \triangleq \mathbf{A}^H\mathbf{y}$ and $\mathbf{G} \triangleq \mathbf{A}^H\mathbf{A}$; let $\mathcal{I} \triangleq \{1,\ldots,BN_t\}$ denote the index set of the elements of $\mathbf{b} = (b_1 \cdots b_{BN_t})^T$; and denote by b_k, z_k, and $G_{k,l}$, with $k, l \in \mathcal{I}$, the elements of $\mathbf{b}$, $\mathbf{z}$, and $\mathbf{G}$, respectively.

The partial ML detector is derived by expanding, with respect to a single bit b_k, the metric $\|\mathbf{y}-\mathbf{A}\mathbf{b}\|^2$ minimized by the ML detector (5). Let $\mathcal{I}_{\sim k} \triangleq \mathcal{I}\setminus\{k\} = \{1,\ldots,k-1,k+1,\ldots,BN_t\}$ and $\mathbf{b}_{\sim k} \triangleq (b_1 \cdots b_{k-1}\; b_{k+1} \cdots b_{BN_t})^T$. Then (5) can be written as [16]

$$\hat{\mathbf{b}}_{\text{ML}}(\mathbf{y}) \;=\; \underset{\mathbf{b}\in\{-1,1\}^{BN_t}}{\arg\max} \big\{ b_k\, \psi_k(\mathbf{b}_{\sim k}) + \rho_k(\mathbf{b}_{\sim k}) \big\}\,, \tag{6}$$

with

$$\psi_k(\mathbf{b}_{\sim k}) \;\triangleq\; 2\left(\Re\{z_k\} - \sum_{l\in\mathcal{I}_{\sim k}} \Re\{G_{k,l}\}\, b_l \right), \tag{7}$$

$$\rho_k(\mathbf{b}_{\sim k}) \;\triangleq\; 2 \sum_{k'\in\mathcal{I}_{\sim k}} \Re\{z_{k'}\}\, b_{k'} \;-\; \sum_{k'\in\mathcal{I}_{\sim k}} \sum_{l\in\mathcal{I}_{\sim k}} b_{k'} G_{k',l}\, b_l\,.$$

Note that $\psi_k(\mathbf{b}_{\sim k})$ and $\rho_k(\mathbf{b}_{\sim k})$ do not involve b_k. Assuming that $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k}) \neq 0$, it follows from (6) that $\hat{b}_k = \hat{b}_{\text{ML},k}$ if and only if $\hat{b}_k = \operatorname{sgn}\big(\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})\big)$ (where $\operatorname{sgn}(x) = 1$ for $x \geq 0$ and -1 for $x < 0$). This means that any ML solution $\hat{\mathbf{b}}_{\text{ML}}$ satisfies

$$\hat{b}_{\text{ML},k} \;=\; \operatorname{sgn}\big(\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})\big)\,, \tag{8}$$

for all k such that $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k}) \neq 0$. Furthermore, it follows from (7) that

$$L_k(\mathcal{D}) \;\leq\; \psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k}) \;\leq\; U_k(\mathcal{D})\,, \tag{9}$$

where

$$L_k(\mathcal{D}) \;\triangleq\; 2\left(\Re\{z_k\} - \sum_{l\in\overline{\mathcal{D}}_{\sim k}} |\Re\{G_{k,l}\}| - \sum_{l\in\mathcal{D}_{\sim k}} \Re\{G_{k,l}\}\, \hat{b}_{\text{ML},l} \right),$$

$$U_k(\mathcal{D}) \;\triangleq\; 2\left(\Re\{z_k\} + \sum_{l\in\overline{\mathcal{D}}_{\sim k}} |\Re\{G_{k,l}\}| - \sum_{l\in\mathcal{D}_{\sim k}} \Re\{G_{k,l}\}\, \hat{b}_{\text{ML},l} \right).$$

Here, $\mathcal{D} \subseteq \mathcal{I}$ and $\overline{\mathcal{D}} = \mathcal{I}\setminus\mathcal{D}$ denote the sets of indices of those bits that are already ML detected and still undetected, respectively. In particular, if $L_k(\mathcal{D}) > 0$, then it follows from (9) that $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k}) > 0$ and thus (recall (8)) $\hat{b}_{\text{ML},k} \equiv \operatorname{sgn}\big(\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})\big) = 1$. Similarly, if $U_k(\mathcal{D}) < 0$, then $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k}) < 0$ and thus $\hat{b}_{\text{ML},k} = -1$. These facts suggest the following iterative detection scheme [16]:

- For $k \in \overline{\mathcal{D}}$ such that $L_k(\mathcal{D}) > 0$, set $\hat{b}_{\text{ML},k} = 1$ and update $\mathcal{D}$ according to $\mathcal{D}^{(\text{new})} = \mathcal{D} \cup \{k\}$ (it follows that $\overline{\mathcal{D}}^{(\text{new})} = \overline{\mathcal{D}}_{\sim k}$).
- For $k \in \overline{\mathcal{D}}$ such that $U_k(\mathcal{D}) < 0$, set $\hat{b}_{\text{ML},k} = -1$ and update $\mathcal{D}$ (and, thus, $\overline{\mathcal{D}}$) as stated previously.
- For all other $k \in \overline{\mathcal{D}}$, $\hat{b}_{\text{ML},k}$ cannot be determined in this manner; here, $\mathcal{D}$ is not changed.

This iterative procedure is initialized with $\mathcal{D} = \emptyset$ (thus, $\overline{\mathcal{D}} = \mathcal{I}$). It is terminated when no more bits can be detected. With $\mathcal{D}_{\text{ML}}$ denoting the index set of the detected ML bits after termination, we have $L_k(\mathcal{D}_{\text{ML}}) \leq 0$ and $U_k(\mathcal{D}_{\text{ML}}) \geq 0$ for all $k \in \overline{\mathcal{D}}_{\text{ML}}$, because otherwise a bit would have been detected. We note that after an update of $\mathcal{D}$, the new bounds $L_k(\mathcal{D}^{(\text{new})})$ and $U_k(\mathcal{D}^{(\text{new})})$ for $\mathcal{D}^{(\text{new})} = \mathcal{D} \cup \{k_0\}$ can be calculated recursively via the update relations

$$L_k(\mathcal{D}^{(\text{new})}) \;=\; L_k(\mathcal{D}) \;-\; 2\big(\Re\{G_{k,k_0}\}\hat{b}_{\text{ML},k_0} - |\Re\{G_{k,k_0}\}|\big)\,,$$

$$U_k(\mathcal{D}^{(\text{new})}) \;=\; U_k(\mathcal{D}) \;-\; 2\big(\Re\{G_{k,k_0}\}\hat{b}_{\text{ML},k_0} + |\Re\{G_{k,k_0}\}|\big)\,,$$

for all $k \in \overline{\mathcal{D}}^{(\text{new})}$.

2.2. Generation of Soft Values

We next consider suboptimum detection of the so-far undetected bits b_k, $k \in \overline{\mathcal{D}}_{\text{ML}}$. First, *soft values* $\breve{b}_k$ are calculated in Stage 2; these will be used in Stage 3 by the genetic algorithm developed in Section 3.

For a given $k \in \overline{\mathcal{D}}_{\text{ML}}$, let $x_k \triangleq \psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})$. We recall from (8) that $\hat{b}_{\text{ML},k} = \operatorname{sgn}(x_k)$ for all k such that $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k}) \neq 0$. Because x_k is unknown except for the fact that $L_k(\mathcal{D}_{\text{ML}}) \leq x_k \leq U_k(\mathcal{D}_{\text{ML}})$ (see (9)), we model it as a random variable uniformly distributed on $[L_k(\mathcal{D}_{\text{ML}}), U_k(\mathcal{D}_{\text{ML}})]$. Then, we define the soft value $\breve{b}_k$ as the expected value of $\hat{b}_{\text{ML},k}$, i.e.,

$$\begin{aligned} \breve{b}_k \;&\triangleq\; \mathrm{E}\{\hat{b}_{\text{ML},k}\} = \mathrm{E}\{\operatorname{sgn}(x_k)\} \\ &=\; 1\cdot\Pr\{x_k > 0\} + (-1)\cdot\Pr\{x_k < 0\} \\ &=\; \frac{L_k(\mathcal{D}_{\text{ML}}) + U_k(\mathcal{D}_{\text{ML}})}{U_k(\mathcal{D}_{\text{ML}}) - L_k(\mathcal{D}_{\text{ML}})}\,, \qquad k \in \overline{\mathcal{D}}_{\text{ML}}\,, \end{aligned} \tag{10}$$

where the uniform distribution of x_k was used in the last step. According to (10), the soft values $\breve{b}_k$ can be easily calculated from $L_k(\mathcal{D}_{\text{ML}})$ and $U_k(\mathcal{D}_{\text{ML}})$. They are bounded as $-1 \leq \breve{b}_k \leq 1$. It should be noted that our soft values $\breve{b}_k$ are defined differently than the soft-bits (log-likelihood ratios) used in turbo processing schemes.

3. THE SOFT-INPUT GENETIC ALGORITHM

For suboptimum detection of the so-far undetected bits b_k, $k \in \overline{\mathcal{D}}_{\text{ML}}$ (Stage 3), we propose an iterative algorithm that we term the *soft-input genetic algorithm* (SGA). This is a soft-input version of the genetic algorithm presented in [17] for solving large binary quadratic programming problems. The SGA differs from [17] in its initialization (which uses the results of Stages 1 and 2), the local search algorithm, and the mutation operation.

A genetic algorithm iteratively improves a set of candidate solutions (CSs). It uses operations inspired by genetics, such as crossover, mutation, and selection, to generate new CSs by modifying and combining existing CSs [18]. Similar to [17], the SGA adds to the genetic operations a local search. The SGA is substantially different from genetic algorithms previously proposed for MIMO detec-

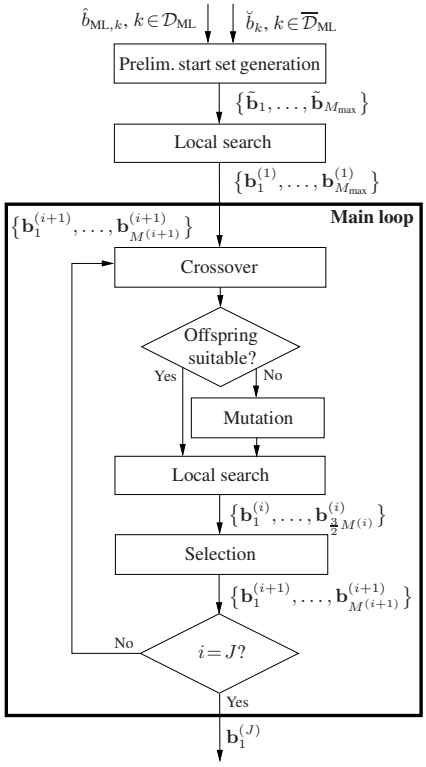

Fig. 2. Block diagram of the SGA with initialization.

tion [10, 19, 20] in that (i) it uses the soft values provided by Stage 2 and the partial ML detection results provided by Stage 1 for an improved initialization, and (ii) it includes a local search procedure, which makes the search for improved CSs more effective. Therefore, the SGA performs well even for very small population sizes. These reduced population sizes result in a low complexity and make the SGA suited to large MIMO systems with tens of antennas.

A block diagram of the SGA is shown in Fig. 2. The two blocks on top generate an initial start set $\{\mathbf{b}_1^{(1)},\ldots,\mathbf{b}_{M_{\max}}^{(1)}\}$ of CSs for the first iteration of the main loop in the SGA. First, a *preliminary* initial start set $\{\tilde{\mathbf{b}}_1,\ldots,\tilde{\mathbf{b}}_{M_{\max}}\}$ is generated, using the ML bits $\hat{b}_{\mathrm{ML},k}$, $k\in\mathcal{D}_{\mathrm{ML}}$ from Stage 1 and the soft values $\breve{b}_k$, $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$ from Stage 2. This preliminary set is then improved by a local search algorithm.

In iteration i of the main loop, the *crossover*, *mutation*, and *local search* steps use the locally optimized CSs, $\{\mathbf{b}_1^{(i)},\ldots,\mathbf{b}_{M^{(i)}}^{(i)}\}$, to calculate $M^{(i)}/2$ additional CSs. Here, $M^{(i)}$ is assumed even for simplicity, with $M^{(i)}\leq M_{\max}$. In the *selection* step, identical CSs in the extended set including the $M^{(i)}/2$ additional CSs are removed, and the best $M^{(i+1)}\leq M_{\max}$ CSs are used as the start set for the next iteration. Hence, the number of CSs in the start set of each iteration and therefore the complexity are limited by $M_{\max}$, whereas the quality of the CSs improves with progressing iterations. After a predetermined maximum number J of iterations, the best CS in the current set is used as the final result of the SGA.

We will now describe the individual SGA steps in more detail.

3.1. Generation of the Preliminary Initial Start Set

Each CS in the preliminary initial start set $\{\tilde{\mathbf{b}}_1,\ldots,\tilde{\mathbf{b}}_{M_{\max}}\}$ contains the ML bits $\hat{b}_{\mathrm{ML},k}$, $k\in\mathcal{D}_{\mathrm{ML}}$ detected in Stage 1. The remaining bits ($k\in\overline{\mathcal{D}}_{\mathrm{ML}}$) are derived from the soft values $\breve{b}_k$, $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$ calculated in Stage 2 by a modified version of the Chase algorithm [21], which yields high CS diversity in a simple and efficient way.

More specifically, the first CS $\tilde{\mathbf{b}}_1$ is generated by quantizing the soft values $\breve{b}_k$, $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$. Thus, the elements of $\tilde{\mathbf{b}}_1$ are given by $\tilde{b}_{1,k}=\hat{b}_{\mathrm{ML},k}$ for $k\in\mathcal{D}_{\mathrm{ML}}$ and $\tilde{b}_{1,k}=\mathrm{sgn}(\breve{b}_k)$ for $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$. The remaining CSs $\tilde{\mathbf{b}}_2,\ldots,\tilde{\mathbf{b}}_{M_{\max}}$ are generated by interpreting $|\breve{b}_k|$ as a reliability measure and flipping unreliable bits. Let us denote the indices $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$ by $k_1,k_2,\ldots,k_K$, with $K\triangleq|\overline{\mathcal{D}}_{\mathrm{ML}}|$ and ordered according to increasing reliability, i.e., $|\breve{b}_{k_1}|\leq|\breve{b}_{k_2}|\leq\ldots\leq|\breve{b}_{k_K}|$. Then $\tilde{\mathbf{b}}_2$ is formed by flipping the two most unreliable bits in $\tilde{\mathbf{b}}_1$, i.e., $\tilde{b}_{2,k}=-\tilde{b}_{1,k}$ for $k\in\{k_1,k_2\}$ and $\tilde{b}_{2,k}=\tilde{b}_{1,k}$ for $k\in\mathcal{I}_{\sim k_1,k_2}$. Similarly, $\tilde{\mathbf{b}}_3$ is formed by flipping the four most unreliable bits in $\tilde{\mathbf{b}}_1$, i.e., $\tilde{b}_{3,k}=-\tilde{b}_{1,k}$ for $k\in\{k_1,k_2,k_3,k_4\}$ and $\tilde{b}_{3,k}=\tilde{b}_{1,k}$ for $k\in\mathcal{I}_{\sim k_1,k_2,k_3,k_4}$. Continuing this way, for each new $\tilde{\mathbf{b}}_j$, two more bits—the most unreliable of those not flipped so far—are flipped. Finally, the last CS $\tilde{\mathbf{b}}_{M_{\max}}$ of the preliminary initial start set is given by $\tilde{b}_{M_{\max},k}=-\tilde{b}_{1,k}$ for $k\in\{k_1,\ldots,k_{2\,(M_{\max}-1)}\}$ and $\tilde{b}_{M_{\max},k}=\tilde{b}_{1,k}$ for $k\in\mathcal{I}_{\sim k_1,\ldots,k_{2(M_{\max}-1)}}$. Here, $M_{\max}$ is a design parameter that satisfies $2\,(M_{\max}-1)\leq K$.

3.2. Local Search

An iterative local search algorithm is used to convert the preliminary initial start set $\{\tilde{\mathbf{b}}_1,\ldots,\tilde{\mathbf{b}}_{M_{\max}}\}$ into the initial start set $\{\mathbf{b}_1^{(1)},\ldots,\mathbf{b}_{M_{\max}}^{(1)}\}$, which serves as the input to the main loop. This algorithm is also used in the main loop after the crossover and mutation operations (see Fig. 2 and Section 3.3). To keep the complexity low, we adopt the simple *1-opt* algorithm [22]; however, more powerful (and more complex) algorithms [17, 22] can be used as well.

In each iteration of the 1-opt algorithm, one bit is flipped in each CS. Consider an existing CS $\mathbf{b}=(b_l)$, with $b_l\in\{-1,1\}$, and a new CS $\mathbf{b}^{(k)}$ that is derived from $\mathbf{b}$ by flipping the bit b_k for some $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$, i.e., $b_k^{(k)}=-b_k$ and $b_l^{(k)}=b_l$ for $l\in\mathcal{I}_{\sim k}$. Here, the optimum $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$ is obtained by minimizing

$$g_k\triangleq f(\mathbf{b}^{(k)})-f(\mathbf{b}),\quad\text{with }f(\mathbf{b})\triangleq\|\mathbf{y}-\mathbf{A}\mathbf{b}\|^2.$$

Note that $f(\mathbf{b})$ is the metric minimized by the ML detector in (5). One easily obtains the expression

$$g_k=-4b_k\left(\Re\{z_k\}-\sum_{l\in\mathcal{I}_{\sim k}}\Re\{G_{k,l}\}\,b_l\right).$$

For the CS $\mathbf{b}$ considered, an iteration of the 1-opt local search algorithm can now be described as follows:

1. By exhaustive search, find the index $k\in\overline{\mathcal{D}}_{\mathrm{ML}}$ with the smallest g_k, i.e., $k_{\mathrm{opt}}\triangleq\arg\min_{k\in\overline{\mathcal{D}}_{\mathrm{ML}}}g_k$.
2. Flip the corresponding bit of $\mathbf{b}$, i.e., form the new CS $\mathbf{b}^{(k_{\mathrm{opt}})}$ with elements $b_{k_{\mathrm{opt}}}^{(k_{\mathrm{opt}})}=-b_{k_{\mathrm{opt}}}$ and $b_l^{(k_{\mathrm{opt}})}=b_l$ for $l\in\mathcal{I}_{\sim k_{\mathrm{opt}}}$. We note that the $g_k^{(\mathrm{new})}$ for the next iteration can be easily obtained by updating the g_k according to

$$g_k^{(\mathrm{new})}=\begin{cases}-g_k\,, & k=k_{\mathrm{opt}}\,,\\ g_k+8\,b_k\,b_{k_{\mathrm{opt}}}\,\Re\{G_{k,k_{\mathrm{opt}}}\}\,, & k\neq k_{\mathrm{opt}}\,.\end{cases}$$

This iterative process is terminated when all $g_k^{(\mathrm{new})}$ are nonnegative, which indicates that no further decrease of $f(\cdot)$ can be obtained by flipping a single bit and, thus, a local minimum of $f(\cdot)$ has been reached. The local search procedure is executed for each CS individually.

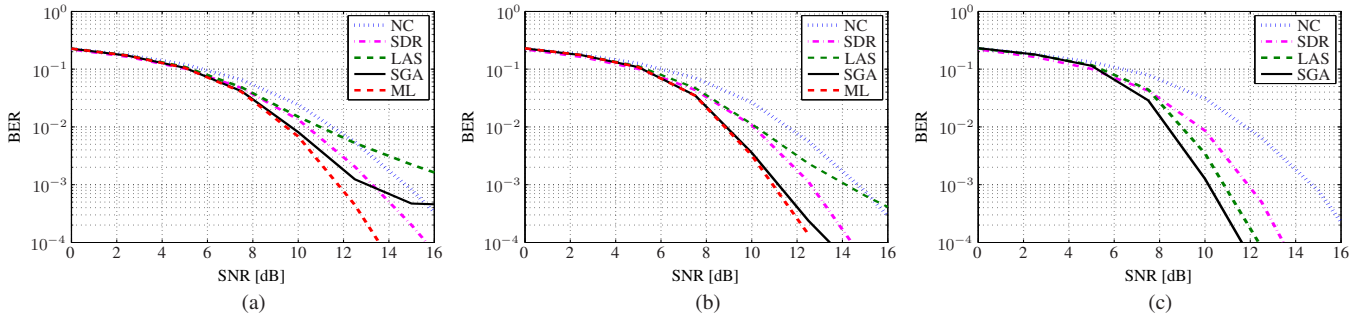

Fig. 3. BER-versus-SNR performance of the SGA detector and various state-of-the-art detectors for three spatial-multiplexing MIMO systems using 4QAM. The system dimension is $N_t = N_r = 8$ in (a), $N_t = N_r = 16$ in (b), and $N_t = N_r = 64$ in (c).

3.3. Crossover, Mutation, Selection

The first stage in the main loop of the SGA is the crossover stage (see Fig. 2). We will use the *uniform crossover* algorithm [18]. First, the current CS set $\{\mathbf{b}_1^{(i)}, \ldots, \mathbf{b}_{M^{(i)}}^{(i)}\}$ is randomly organized into pairs of CSs. (If $M^{(i)}$ is odd, one of the CSs appears in two pairs.) Each CS pair $\mathbf{b}_j^{(i)}, \mathbf{b}_{j'}^{(i)}$ produces an *offspring CS* $\mathbf{b}'$, which inherits those bits from the parent CSs that are equal in the parent CSs (note that these bits include all ML bits $\hat{b}_{\mathrm{ML},k}$, $k \in \mathcal{D}_{\mathrm{ML}}$), while the remaining bits are chosen randomly. The current set of CSs is then extended by the offspring CSs. The size of the extended set is $M_{\mathrm{ext}}^{(i)} = \lceil \frac{3}{2} M^{(i)} \rceil$, where $\lceil x \rceil$ denotes the smallest integer not smaller than x.

The crossover stage is followed by the *mutation* stage. Consider an offspring $\mathbf{b}'$ for which the Hamming distance of the parents satisfies $d(\mathbf{b}_j^{(i)}, \mathbf{b}_{j'}^{(i)}) \leq 2$. Because this implies that $d(\mathbf{b}', \mathbf{b}_j^{(i)}) \leq 2$ and $d(\mathbf{b}', \mathbf{b}_{j'}^{(i)}) \leq 2$, it is likely that the bit-flipping modification of $\mathbf{b}'$ performed by the subsequent local search procedure (see Fig. 2) results in $\mathbf{b}_j^{(i)}$ or $\mathbf{b}_{j'}^{(i)}$. To avoid this situation, one additional bit of $\mathbf{b}'$ at a randomly chosen position $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ is flipped ("mutation"). Subsequently, the new CSs created by the crossover and mutation stages are optimized by another local search stage.

The final *selection* stage reduces the set of $M_{\mathrm{ext}}^{(i)} = \lceil \frac{3}{2} M^{(i)} \rceil$ CSs. First, identical CSs in the set are removed. Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_Q\}$ with $Q \leq M_{\mathrm{ext}}^{(i)}$ denote the resulting CS set. If $Q \leq M_{\max}$, this set is used as the start set $\{\mathbf{b}_1^{(i+1)}, \ldots, \mathbf{b}_{M^{(i+1)}}^{(i+1)}\}$ for the next iteration (hence, $M^{(i+1)} = Q$). If $Q > M_{\max}$, the start set for the next iteration is chosen as the $M_{\max}$ CSs $\mathbf{b}_j$ with smallest $f(\mathbf{b}_j)$ values (hence, $M^{(i+1)} = M_{\max}$). Thus, $M^{(i+1)} \leq M_{\max}$ is always satisfied. A similar selection approach has been used in [17].

After the predetermined maximum number J of main loop iterations, the CS set $\{\mathbf{b}_1^{(J+1)}, \ldots, \mathbf{b}_{M^{(J+1)}}^{(J+1)}\}$ is obtained. The best CS in this set is used as the final result of the SGA. Alternatively, if a coded MIMO system using a soft-input channel decoder is considered, a modified "soft-output" version of the SGA detector can be established. In fact, using the max-log approximation as described in [20], one can compute from the CS set $\{\mathbf{b}_1^{(J+1)}, \ldots, \mathbf{b}_{M^{(J+1)}}^{(J+1)}\}$ soft information for the channel decoder.

4. SIMULATION RESULTS

We will demonstrate the bit error rate (BER) performance and computational complexity of the proposed SGA-based detector for a spatial-multiplexing multiantenna system [1, 6] with an iid Gaussian MIMO channel. We compare our detector with ML detection (2) using the Schnorr-Euchner sphere decoder [23, 24], the nulling-and-canceling (NC) detector [7] using the efficient implementation described in [25], a detector based on semidefinite relaxation (SDR) with rank-one approximation [9], and a three-stage linear ascent search (LAS) algorithm, which is a detector specially designed for large MIMO systems [13]. We did not simulate existing genetic algorithms for MIMO detection, such as [19, 20], since they assume large population sizes and small MIMO systems and are therefore not suited to our simulation setting. The NC detector requires an estimate of the noise variance σ_n^2; however, the true value of σ_n^2 was used in our simulations. In the SGA, the number $M_{\max}$ of CSs in the preliminary inital start set was chosen as a function of the number $|\overline{\mathcal{D}}_{\mathrm{ML}}|$ of undetected bits after partial ML detection (Stage 1) according to

$$M_{\max} = \left\lceil 0.8 \, \frac{|\overline{\mathcal{D}}_{\mathrm{ML}}|}{2} \right\rceil + 1 . \tag{11}$$

This can be shown to imply that $\tilde{\mathbf{b}}_{M_{\max}}$ differs from $\tilde{\mathbf{b}}_1$ (cf. Section 3.1) in at least 80% of the undetected bits, $\tilde{b}_k$ for $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$. The SGA was termined after $J = 18$ iterations.

In Fig. 3, the BER-versus-SNR performance of the various detectors is shown for MIMO systems of dimension 8×8, 16×16, and 64×64, using a 4QAM constellation. It is seen that the proposed SGA-based detector (briefly termed "SGA" in the following) outperforms the other suboptimum detectors for the 8×8 system below an SNR of about 13 dB and for the 16×16 and 64×64 systems for all displayed SNR values; its performance advantage over NC and SDR increases with growing system dimension. Furthermore, for the 16×16 system, the performance of SGA is close to that of the ML detector. (The ML detector is not included in Fig. 3(c) since it is too complex to simulate for a 64×64 system.) For the 8×8 system, SGA exhibits an error floor at high SNR. The SNR range of the error floor depends on the number of CSs used in the initialization procedure, and thus, according to (11), on the system size. For the 16×16 and 64×64 systems, the error floor occurs at SNR values higher than 16 dB, which are not shown in Fig. 3.

Table 1 presents estimates of the computational complexity (kflop count) of the different detectors averaged over all simulated SNR values. These estimates were obtained using the Lightspeed toolbox [26] for MATLAB. (The kflop count for the SDR detector is not included because our implementation of SDR uses an external toolbox that cannot be accessed by the Lightspeed routines.) Note that the kflop estimates should be interpreted with caution as they are implementation-dependent. We distinguish between the complexity of the operations performed when the channel matrix $\mathbf{H}$ changes (termed "preparation complexity") and the complexity of the operations performed once for each received vector $\mathbf{y}$ (termed "vector

PREPARATION COMPLEXITY (kflops per block preparation)				
$N_t = N_r$	NC	LAS	SGA	ML
8	17	18	4	3
16	126	146	32	22
32	964	1156	262	175
64	7528	9210	2097	1398

VECTOR COMPLEXITY (kflops per received vector)				
$N_t = N_r$	NC	LAS	SGA	ML
8	8	28	41	23
16	67	233	341	7603
32	552	1914	2844	–
64	4479	15601	25038	–

Table 1. Computational complexity (in kflops) of the SGA detector and various state-of-the-art detectors for spatial-multiplexing MIMO systems using 4QAM, with different system dimensions $N_t = N_r$.

complexity"). One can observe that for SGA, doubling the system dimension $N \triangleq N_t = N_r$ results in a (roughly) 8-fold increase in the vector complexity; this suggests that the vector complexity scales roughly cubically with the system dimension. We note that a similar scaling behavior is exhibited by NC [7] and LAS [13], which are known to scale as $\mathcal{O}(N^3)$, while the scaling behavior of SDR-based detection is $\mathcal{O}(N^{9/2})$ [9].

It is furthermore seen that the vector complexity of SGA is higher than that of NC and LAS; this is the price paid for the better performance of SGA. (The vector complexity of SDR can be expected to be higher than that of SGA.) However, the preparation complexity of SGA is significantly lower than that of NC and LAS. Finally, the vector complexity of SGA is seen to be dramatically lower than that of ML detection (sphere decoding) in the 16×16 system. (The vector complexity of the sphere decoder for the 32×32 and 64×64 systems is not shown in Table 1 because of the excessive computational cost.)

5. CONCLUSION

We presented a low-complexity bit-level detector for large MIMO systems employing BPSK or QAM constellations. This detector combines efficient partial ML detection with suboptimum detection based on soft values. The suboptimum detection stage uses a novel soft-input genetic optimization algorithm that includes a local search procedure. Simulation results showed that the performance of the proposed detector can be close to that of the ML detector, and for large MIMO systems, it can be better than that of state-of-the-art suboptimum methods. The complexity scales roughly cubically with the system dimension. Thus, our detector is particularly attractive for large MIMO systems where ML detection (sphere decoding) is infeasible. For coded systems using a soft-input decoder, an extension of the proposed detector to soft-output detection is possible.

6. REFERENCES

[1] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications*. New York, NY: Cambridge University Press, 2007.

[2] K. Fazel and S. Kaiser, *Multi-Carrier and Spread Spectrum Systems: From OFDM and MC-CDMA to LTE and WiMAX*. Chichester, UK: Wiley, 2nd ed., 2008.

[3] U. Fincke and M. Phost, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comp.*, vol. 44, pp. 463–471, Apr. 1985.

[4] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, pp. 1474–1484, Apr. 2005.

[5] G. K. Kaleh, "Channel equalization for block transmission systems," *IEEE J. Sel. Areas Comm.*, vol. 13, pp. 110–121, Jan. 1995.

[6] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proc. URSI Int. Symp. Signals, Syst., Electron.*, Pisa, Italy, pp. 295–300, Sep. 1998.

[7] B. Hassibi, "A fast square-root implementation for BLAST," in *Proc. Asilomar Conf. Sig., Syst., Comput.*, Pacific Grove, CA, pp. 1255–1259, Nov. 2000.

[8] D. Wübben, D. Seethaler, J. Jaldén, and G. Matz, "Lattice reduction," *IEEE Signal Process. Mag.*, vol. 28, pp. 70–91, May 2011.

[9] Z.-Q. Luo, W.-K. Ma, A.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, pp. 20–34, May 2010.

[10] M. Jiang and L. Hanzo, "Multiuser MIMO-OFDM for next-generation wireless systems," *Proc. IEEE*, vol. 95, pp. 1430–1469, Jul. 2007.

[11] S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "A low-complexity near-ML performance achieving algorithm for large MIMO detection," in *Proc. IEEE ISIT '08*, Toronto, Canada, pp. 2012–2016, Jul. 2008.

[12] S. K. Mohammed, A. Zaki, A. Chockalingam, and B. S. Rajan, "High-rate space time coded large-MIMO systems: Low-complexity detection and channel estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 3, pp. 958–974, Dec. 2009.

[13] T. Datta, N. Srinidhi, A. Chockalingam, and B. S. Rajan, "Random-restart reactive tabu search algorithm for detection in large-MIMO systems," *IEEE Comm. Lett.*, vol. 14, pp. 1107–1109, Dec. 2010.

[14] P. Som, T. Datta, A. Chockalingam, and B. S. Rajan, "Improved large-MIMO detection based on damped belief propagation," in *Proc. IEEE ITW '10*, Dublin, Ireland, Jan. 2010.

[15] J. Choi, "Iterative receivers with bit-level cancellation and detection for MIMO-BICM systems," *IEEE Signal Process. Lett.*, vol. 53, pp. 4568–4577, Dec. 2005.

[16] P. Ödling, H. B. Eriksson, and P. O. Börjesson, "Making MLSD decisions by thresholding the matched filter output," *IEEE Trans. Comm.*, vol. 48, pp. 324–332, Feb. 2000.

[17] K. Katayama, M. Tani, and H. Narihisa, "Solving large binary quadratic programming problems by effective genetic local search algorithm," in *Proc. GECCO-2000*, Las Vegas, NV, pp. 643–650, Jul. 2000.

[18] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham, *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Boca Raton, FL: Chapman & Hall/CRC, 2009.

[19] S. Bashir, A. A. Khan, M. Naeem, and S. I. Shah, "An application of GA for symbol detection in MIMO communication systems," in *Proc. ICNC '07*, Haikou, China, pp. 404–410, Aug. 2007.

[20] M. Jiang, J. Akhtman, and L. Hanzo, "Soft-information assisted near-optimum nonlinear detection for BLAST-type space division multiplexing OFDM systems," *IEEE Trans. Wireless Comm.*, vol. 6, pp. 1230–1234, Apr. 2007.

[21] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, pp. 170–182, Jan. 1972.

[22] P. Merz and B. Freisleben, "Greedy and local search heuristics for unconstrained binary quadratic programming," *J. Heuristics*, vol. 8, pp. 197–213, Mar. 2002.

[23] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programm.*, vol. 66, pp. 181–199, Jan. 1994.

[24] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, pp. 2201–2214, Aug. 2002.

[25] J. Benesty, Y. Huang, and J. Chen, "A fast recursive algorithm for optimum sequential signal detection in a BLAST system," *IEEE Trans. Signal Process.*, vol. 51, pp. 1722–1730, Jul. 2003.

[26] T. Minka, "The Lightspeed Matlab toolbox, version 2.5," 2011. Available online: http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/.