IEEE Trans. Signal Process., vol. 61, no. 18, Sept. 2013, pp. 4573–4586.

Soft-Heuristic Detectors for Large MIMO Systems

Pavol Švač, *Member, IEEE*, Florian Meyer, *Student Member, IEEE*, Erwin Riegler, *Member, IEEE*, and Franz Hlawatsch, *Fellow, IEEE*

***Abstract*—We propose low-complexity detectors for large MIMO systems with BPSK or QAM constellations. These detectors work at the bit level and consist of three stages. In the first stage, maximum likelihood decisions on certain bits are made in an efficient way. In the second stage, soft values for the remaining bits are calculated. In the third stage, these remaining bits are detected by means of a heuristic programming method for high-dimensional optimization that uses the soft values ("soft-heuristic" algorithm). We propose two soft-heuristic algorithms with different performance and complexity. We also consider a feedback of the results of the third stage for computing improved soft values in the second stage. Simulation results demonstrate that, for large MIMO systems, our detectors can outperform state-of-the-art detectors based on nulling and canceling, semidefinite relaxation, and likelihood ascent search.**

***Index Terms*—Multiple-input multiple-output systems, large MIMO systems, MIMO detection, spatial multiplexing, OFDM, ICI mitigation, heuristic programming, genetic algorithm.**

I. Introduction

Multiple-input/multiple-output (MIMO) systems for wireless communications have received considerable interest [1]. A MIMO system with input dimension N_t and output dimension N_r can be described by the input-output relation

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

where $\mathbf{s} \in \mathcal{S}^{N_t}$ is the transmit symbol vector (here, $\mathcal{S}$ denotes a finite symbol alphabet), $\mathbf{y} \in \mathbb{C}^{N_r}$ 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. The MIMO model (1) is relevant to multi-antenna wireless systems [1], orthogonal frequency-division multiplexing (OFDM) systems [2], and code-division multiple access (CDMA) systems [2]. Here, we consider the detection of $\mathbf{s}$ from $\mathbf{y}$ under the frequently used assumptions that the channel matrix $\mathbf{H}$ is known and the noise $\mathbf{n}$ is 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 σ_n^2 is the noise variance and $\mathbf{I}_{N_r}$ is the $N_r \times N_r$ identity matrix.

A. State of the Art

The result of maximum-likelihood (ML) detection, which minimizes the error probability for equally likely transmit vectors $\mathbf{s} \in \mathcal{S}^{N_t}$, is given by [1]

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

ML detection is infeasible for larger MIMO systems because its computational complexity grows exponentially with N_t. This is also true for efficient implementations of ML detection using the sphere-decoding algorithm [3].

Among the suboptimum detection methods, linear equalization methods [4] often perform poorly because each symbol is quantized individually. Detection using decision-feedback equalization, also known as nulling-and-canceling (NC), outperforms linear equalization but is still inferior to ML detection [4]. NC implementations with reliability-based symbol ordering include V-BLAST [5], [6] and dynamic NC [7]. Detectors based on lattice reduction have polynomial average complexity and tend to outperform equalization-based detection [8], [9]. Detectors based on semidefinite relaxation (SDR) [10] exhibit excellent performance but are significantly more complex than equalization-based detectors. The "subspace marginalization with interference suppression" (SUMIS) soft-output detector [11] has a low and fixed (deterministic) complexity. The suboptimum multistage detectors proposed in [12] can achieve near-optimum performance with a complexity much lower than that of sphere decoding. A survey of MIMO detection using heuristic optimization (or programming) methods, such as genetic algorithms, short-term or reactive tabu search, simulated annealing, particle swarm optimization, and 1-opt local search, is given in [13]. In particular, several adaptations of genetic algorithms to MIMO detection have been proposed (see [14] and references therein).

Recently, large MIMO systems with several tens of antennas have attracted increased attention due to their high capacity. Suboptimum detection methods for large MIMO systems include local search algorithms such as likelihood ascent search (LAS) [15], [16] and reactive tabu search [17], as well as a belief propagation algorithm [18].

B. Contribution

Extending our work in [19], we present low-complexity detectors for large MIMO systems using a BPSK or QAM symbol alphabet $\mathcal{S}$. The proposed MIMO detectors operate at the bit level and consist of three stages as depicted in Fig. 1. The first stage performs *partial ML detection*. Let the bit vector $\hat{\mathbf{b}}_{\mathrm{ML}} = (\hat{b}_{\mathrm{ML},k})$ denote the ML solution at bit level that corresponds to $\hat{\mathbf{s}}_{\mathrm{ML}}$ as described in [20]. In the first stage, certain bits $\hat{b}_{\mathrm{ML},k}$ are calculated by means of the iterative algorithm presented in [21]. We reformulate that algorithm in terms of lower and upper bounds that also play an important role in the following stages. (We note that in

P. Švač was with the Institute of Telecommunications, Vienna University of Technology, Vienna, Austria. He is now with SkyToll, 84104 Bratislava, Slovakia (email: pavol.svac@skytoll.sk). F. Meyer, E. Riegler, and F. Hlawatsch are with the Institute of Telecommunications, Vienna University of Technology, A-1040 Vienna, Austria (e-mail: {fmeyer,eriegler,fhlawats}@nt.tuwien.ac.at). 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). Parts of this work were presented at IEEE SPAWC 2012, Çeşme, Turkey, June 2012.

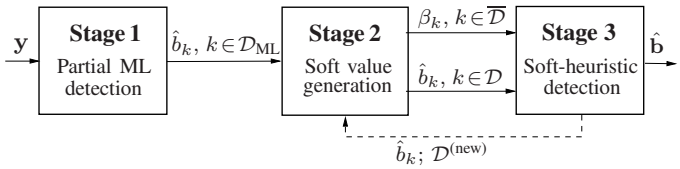


Fig. 1. Block diagram of the proposed MIMO detector. $\mathcal{D}$ and $\overline{\mathcal{D}}$ denote the sets of indices k of, respectively, the detected and undetected bits b_k at a given iteration.

contrast to [21], where a single-input single-output system with intersymbol interference was considered and the undetected bits were subsequently detected using a linear or decision-feedback equalizer, here we consider a MIMO system and replace the equalizer by a novel bit-level detector consisting of the second and third stages shown in Fig. 1.)

In the second stage, *soft values* β_k for the undetected bits are calculated from the lower and upper bounds.

In the third stage, the undetected bits b_k are detected by means of an iterative "soft-heuristic" optimization algorithm that uses the ML bits $\hat{b}_{\mathrm{ML},k}$ and soft values β_k produced by the first two stages. We propose two soft-heuristic algorithms with different performance and complexity. Both algorithms are based on principles used to solve large-scale optimization problems and are therefore especially suitable for large MIMO systems. The *sequential soft-heuristic algorithm* is a soft-input version of the greedy optimization algorithm presented in [22], however using an improved (nongreedy) order of decisions inspired by the Nelder-Mead algorithm [23]. The *genetic soft-heuristic algorithm* [19] is a soft-input and otherwise modified version of the genetic algorithm presented in [24]. It is substantially different from genetic algorithms previously proposed for MIMO detection [14], [25], [26] in that it uses the results of the first two stages for an improved initialization and includes a local search procedure that produces improved candidate solutions even for very small population sizes. The reduced population sizes result in a low complexity and make the algorithm suited to large MIMO systems. In the sequential soft-heuristic algorithm, the bits detected by the third stage are fed back to the second stage in order to obtain improved soft values. A similar feedback can also be used with the genetic soft-heuristic algorithm.

The proposed MIMO detectors are shown via simulation to achieve near-optimum bit error rate (BER) performance for large MIMO systems. In spatial-multiplexing multiantenna systems, they outperform detectors based on NC, SDR, SUMIS, and LAS, with growing advantages over NC, SDR, and SUMIS for larger systems and a strongly reduced complexity compared to ML detection (i.e., sphere decoding). For intercarrier interference mitigation in OFDM systems, they significantly outperform minimum mean-square error (MMSE) equalization based detection and achieve effectively ML performance just as NC, SDR, SUMIS, and LAS; moreover, by exploiting the diagonal dominance of the channel matrix, they are much less complex than NC, SDR, SUMIS, and LAS.

This paper is organized as follows. In Section II, we review the partial ML detection method of [21] (Stage 1) and describe the generation of soft values (Stage 2). In Sections III and IV, two soft-heuristic algorithms for Stage 3 are developed. In Section V, the performance of the proposed detectors is assessed experimentally in comparison to ML detection and state-of-the-art suboptimum detection.

II. Partial ML Detection and Generation of Soft Values

For a QAM symbol alphabet $\mathcal{S}$, where $|\mathcal{S}|=2^B$ with an even $B=\log_2|\mathcal{S}|$, there is a unique vector $\mathbf{v}=(v_1\cdots v_B)^T\in\mathbb{C}^B$ such that every symbol $s\in\mathcal{S}$ can be written as [20]

$$s \;=\; \sum_{m=1}^{B} v_m \breve{b}_m(s) \;=\; \mathbf{v}^T\breve{\mathbf{b}}(s)\,, \tag{3}$$

with a bit vector $\breve{\mathbf{b}}(s)=\big(\breve{b}_1(s)\cdots\breve{b}_B(s)\big)^T\in\{-1,1\}^B$ that provides a unique binary representation of the symbol s. The complex vector $\mathbf{v}$ only depends on $|\mathcal{S}|$: e.g., $\mathbf{v}=(1\quad j)^T$ for $|\mathcal{S}|=4$, $\mathbf{v}=(2\quad 1\quad 2j\quad j)^T$ for $|\mathcal{S}|=16$, and $\mathbf{v}=(4\quad 2\quad 1\quad 4j\quad 2j\quad j)^T$ for $|\mathcal{S}|=64$. For $|\mathcal{S}|\geq 16$, the binary representation defined by (3) is not a Gray mapping. Although BPSK is not a special case of QAM, it is nevertheless a (trivial) special case of (3), with $B=1$, $\mathbf{v}=(1)$, and $\breve{b}_1(s)=s$.

Let $s_p=(\mathbf{s})_p$ denote the pth element of $\mathbf{s}$. For QAM or BPSK, using (3) for each s_p, a binary representation of the MIMO system in (1) is obtained as [20]

$$\mathbf{y} \;=\; \mathbf{A}\mathbf{b}+\mathbf{n}\,.$$

Here, $\mathbf{A}\triangleq\mathbf{H}\otimes\mathbf{v}^T\in\mathbb{C}^{N_r\times BN_t}$ ($\otimes$ denotes the Kronecker product) is an equivalent channel matrix and $\mathbf{b}=\mathbf{b}(\mathbf{s})\triangleq\big(\breve{\mathbf{b}}^T(s_1)\cdots\breve{\mathbf{b}}^T(s_{N_t})\big)^T\in\{-1,1\}^{BN_t}$ is the binary representation of the transmit symbol vector $\mathbf{s}$. For BPSK, $\mathbf{b}=\mathbf{s}$ and $\mathbf{A}=\mathbf{H}$. The ML detection rule (2) can now be equivalently formulated at the bit level as

$$\hat{\mathbf{b}}_{\mathrm{ML}}(\mathbf{y}) \;=\; \underset{\mathbf{b}\in\{-1,1\}^{BN_t}}{\arg\min}\ \|\mathbf{y}-\mathbf{A}\mathbf{b}\|^2. \tag{4}$$

A. Partial ML Detection

The first stage of the proposed MIMO detector computes *some* elements $\hat{b}_{\mathrm{ML},k}$ of the ML detection result $\hat{\mathbf{b}}_{\mathrm{ML}}(\mathbf{y})$ in (4) in an efficient manner. This is done by means of the algorithm proposed in [21], which will now be reviewed. In what follows, let $\mathbf{z}\triangleq\mathbf{A}^H\mathbf{y}$ and $\mathbf{G}\triangleq\mathbf{A}^H\mathbf{A}$. Furthermore, let $\mathcal{I}\triangleq\{1,\ldots,BN_t\}$ denote the index set of the elements of $\mathbf{b}=\big(\breve{\mathbf{b}}^T(s_1)\cdots\breve{\mathbf{b}}^T(s_{N_t})\big)^T=(b_1\cdots b_{BN_t})^T$, 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. As explained in the following, we expand the ML metric $\|\mathbf{y}-\mathbf{A}\mathbf{b}\|^2$ with respect to a specific bit b_k, with $k\in\mathcal{I}$ arbitrary but fixed [21]. Let $\mathcal{I}_{\sim k}\triangleq\mathcal{I}\backslash\{k\}=\{1,\ldots,k-1,k+1,\ldots,BN_t\}$ and $\mathbf{b}_{\sim k}\triangleq(b_1\cdots b_{k-1}\ b_{k+1}\cdots b_{BN_t})^T$. We have

$$\|\mathbf{y}-\mathbf{A}\mathbf{b}\|^2 = \|\mathbf{y}\|^2 - f(\mathbf{b})\,,$$

with

$$\begin{aligned} f(\mathbf{b}) \;&\triangleq\; 2\,\Re\{\mathbf{z}^H\mathbf{b}\}-\mathbf{b}^T\mathbf{G}\mathbf{b} \\ &=\; 2b_k\Re\{z_k\} - G_{k,k}\underbrace{b_k^2}_{1} - \sum_{l\in\mathcal{I}_{\sim k}} b_k G_{k,l}\, b_l \end{aligned}$$

$$- \sum_{k' \in \mathcal{I}_{\sim k}} b_{k'} G_{k',k} b_k + 2 \sum_{k' \in \mathcal{I}_{\sim k}} b_{k'} \Re\{z_{k'}\} - \sum_{k' \in \mathcal{I}_{\sim k}} \sum_{l \in \mathcal{I}_{\sim k}} b_{k'} G_{k',l} b_l \,. \quad (5)$$

The ML detection rule (4) can then be rewritten as

$$\hat{\mathbf{b}}_{\mathrm{ML}}(\mathbf{y}) = \underset{\mathbf{b} \in \{-1,1\}^{BN_t}}{\arg\max} f(\mathbf{b}) \quad (6)$$

$$= \underset{\mathbf{b} \in \{-1,1\}^{BN_t}}{\arg\max} \big\{ b_k \psi_k(\mathbf{b}_{\sim k}) + \rho_k(\mathbf{b}_{\sim k}) \big\} \,, \quad (7)$$

with

$$\psi_k(\mathbf{b}_{\sim k}) \triangleq 2 \left(\Re\{z_k\} - \sum_{l \in \mathcal{I}_{\sim k}} \Re\{G_{k,l}\} b_l \right) \quad (8)$$

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

Because $\psi_k(\mathbf{b}_{\sim k})$ and $\rho_k(\mathbf{b}_{\sim k})$ do not involve b_k, (7) shows how the function maximized by $\hat{\mathbf{b}}_{\mathrm{ML}}$ depends on b_k. In particular, assuming that $\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k}) \neq 0$, it follows from the presence of $b_k \psi_k(\mathbf{b}_{\sim k})$ in (7) that $\hat{b}_k = \hat{b}_{\mathrm{ML},k}$ if and only if $\hat{b}_k = \mathrm{sgn}\big(\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k})\big)$. Thus, the ML solution $\hat{\mathbf{b}}_{\mathrm{ML}}$ satisfies

$$\hat{b}_{\mathrm{ML},k} = \mathrm{sgn}\big(\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k})\big) \,, \quad (9)$$

for all k such that $\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k}) \neq 0$. (If $\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k}) = 0$, breaking the tie either way still leads to an ML decision, i.e., the ML solution is not unique.)

Of course, $\hat{\mathbf{b}}_{\mathrm{ML},\sim k}$ is unknown and thus (9) cannot be directly used for determining $\hat{b}_{\mathrm{ML},k}$. However, it follows from (8) that $\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k})$ can be bounded according to

$$L_k(\mathcal{D}) \leq \psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k}) \leq U_k(\mathcal{D}) \,, \quad (10)$$

where

$$L_k(\mathcal{D}) \triangleq 2 \left(\Re\{z_k\} - \sum_{l \in \overline{\mathcal{D}}_{\sim k}} |\Re\{G_{k,l}\}| - \sum_{l \in \mathcal{D}_{\sim k}} \Re\{G_{k,l}\} \hat{b}_{\mathrm{ML},l} \right) \quad (11)$$

$$U_k(\mathcal{D}) \triangleq 2 \left(\Re\{z_k\} + \sum_{l \in \overline{\mathcal{D}}_{\sim k}} |\Re\{G_{k,l}\}| - \sum_{l \in \mathcal{D}_{\sim k}} \Re\{G_{k,l}\} \hat{b}_{\mathrm{ML},l} \right) . \quad (12)$$

Here, $\mathcal{D} \subseteq \mathcal{I}$ and $\overline{\mathcal{D}} = \mathcal{I} \setminus \mathcal{D}$ denote the sets of indices of the already detected and still undetected bits, respectively. In particular, if $L_k(\mathcal{D}) \geq 0$, it follows from (10) that $\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k}) \geq 0$ and thus (recall (9)) $\hat{b}_{\mathrm{ML},k} = \mathrm{sgn}\big(\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k})\big) = 1$. Similarly, if $U_k(\mathcal{D}) \leq 0$, then $\psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k}) \leq 0$ and thus $\hat{b}_{\mathrm{ML},k} = -1$. This suggests the following iterative detection scheme [21]. In each iteration, consider all $k \in \overline{\mathcal{D}}$ (where $\overline{\mathcal{D}} = \mathcal{I} \setminus \mathcal{D}$ may be updated repeatedly during the iteration as explained presently), and take the following actions:

- For $k \in \overline{\mathcal{D}}$ such that $L_k(\mathcal{D}) \geq 0$, set $\hat{b}_{\mathrm{ML},k} = 1$ and update the index set $\mathcal{D}$ according to $\mathcal{D}^{(\mathrm{new})} = \mathcal{D} \cup \{k\}$ (it follows that $\overline{\mathcal{D}}^{(\mathrm{new})} = \overline{\mathcal{D}}_{\sim k}$).
- For $k \in \overline{\mathcal{D}}$ such that $U_k(\mathcal{D}) \leq 0$, set $\hat{b}_{\mathrm{ML},k} = -1$ and update $\mathcal{D}$ (and, thus, $\overline{\mathcal{D}}$) as stated previously.
- For all other $k \in \overline{\mathcal{D}}$, $\hat{b}_{\mathrm{ML},k}$ 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 if no new bits can be detected. Let $\mathcal{D}_{\mathrm{ML}}$ denote the index set of the detected ML bits after termination, i.e., of all $\hat{b}_{\mathrm{ML},k}$ detected in the partial ML detection stage. The corresponding bounds $L_k(\mathcal{D}_{\mathrm{ML}})$ and $U_k(\mathcal{D}_{\mathrm{ML}})$ satisfy

$$L_k(\mathcal{D}_{\mathrm{ML}}) < 0 \;\text{ and }\; U_k(\mathcal{D}_{\mathrm{ML}}) > 0 \,, \quad \text{for all } k \in \overline{\mathcal{D}}_{\mathrm{ML}} \,, \quad (13)$$

because otherwise a bit would have been detected.

Note that by updating $\mathcal{D}$ after each bit detection, as proposed in [21], the already detected bits are used to produce successively tighter bounds $L_k(\mathcal{D})$ and $U_k(\mathcal{D})$ within a given iteration step. After detection of a bit $\hat{b}_{\mathrm{ML},k_0}$ at position k_0, the new bounds $L_k(\mathcal{D}^{(\mathrm{new})})$ and $U_k(\mathcal{D}^{(\mathrm{new})})$ for $\mathcal{D}^{(\mathrm{new})} = \mathcal{D} \cup \{k_0\}$ (equivalently, $\overline{\mathcal{D}}^{(\mathrm{new})} = \overline{\mathcal{D}}_{\sim k_0}$) can be calculated recursively by means of the update relations (cf. (11), (12))

$$L_k(\mathcal{D}^{(\mathrm{new})}) = L_k(\mathcal{D}) - 2\big(\Re\{G_{k,k_0}\} \hat{b}_{\mathrm{ML},k_0} - |\Re\{G_{k,k_0}\}|\big) \quad (14)$$

$$U_k(\mathcal{D}^{(\mathrm{new})}) = U_k(\mathcal{D}) - 2\big(\Re\{G_{k,k_0}\} \hat{b}_{\mathrm{ML},k_0} + |\Re\{G_{k,k_0}\}|\big) \,, \quad (15)$$

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

The bits detected by the above algorithm equal the corresponding elements of the ML detection result $\hat{\mathbf{b}}_{\mathrm{ML}}(\mathbf{y})$; of course, they are not necessarily correct. It is *a priori* unknown which of the BN_t bits b_k can be detected, i.e., the index set $\mathcal{D}_{\mathrm{ML}}$ is *a priori* unknown. However, some general observations can be made. From (11) and (12), it can be concluded that the bounds $L_k(\mathcal{D})$ and $U_k(\mathcal{D})$ are close to each other if the real parts of the off-diagonal elements of $\mathbf{G}$ in the kth row, $\Re\{G_{k,l}\}$, $l \neq k$, are small. If additionally $\Re\{z_k\}$ is not close to 0, it is very likely that either $L_k(\mathcal{D}) \geq 0$ or $U_k(\mathcal{D}) \leq 0$ and thus an ML decision on b_k can be made. In particular, if $\mathbf{H}$ has orthogonal columns and BPSK modulation is used ($\mathbf{A} = \mathbf{H}$), the matrix $\mathbf{G} = \mathbf{A}^H \mathbf{A}$ is diagonal. Here, $L_k(\mathcal{D}) = U_k(\mathcal{D}) = 2\Re\{z_k\}$, and thus an ML decision on *all* b_k, $k = 1, \ldots, BN_t$ can be made in a single iteration. We also note that $|\overline{\mathcal{D}}_{\mathrm{ML}}|$ grows with the total number of bits, $N_t B = N_t \log_2 |\mathcal{S}|$.

As previously observed in [21], the probability of a detection tends to decrease with increasing signal-to-noise ratio (SNR). In fact, as the noise power decreases, the probability that $L_k(\mathcal{D}) \geq 0$ or $U_k(\mathcal{D}) \leq 0$ decreases, and thus the number of decisions decreases [21]. (However, the reliability of the decisions increases with the SNR.)

B. Generation of Soft Values

For detection of the bits b_k, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ that were not detected by the partial ML detection stage (Stage 1), we first generate *soft values* $\beta_k \in \mathbb{R}$ (Stage 2). The soft values will constitute an input to Stage 3. For a given $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$, let $x_k \triangleq \psi_k(\hat{\mathbf{b}}_{\mathrm{ML},\sim k})$ for brevity. We recall from (9) that $\hat{b}_{\mathrm{ML},k} = \mathrm{sgn}(x_k)$ for all k such that $x_k \neq 0$. Because x_k is unknown except for the fact that $L_k(\mathcal{D}_{\mathrm{ML}}) \leq x_k \leq U_k(\mathcal{D}_{\mathrm{ML}})$ (see (10)), we model x_k as a random variable that is uniformly distributed on $\big[L_k(\mathcal{D}_{\mathrm{ML}}), U_k(\mathcal{D}_{\mathrm{ML}})\big]$. Note that this interval includes 0

because of (13). We now define the soft value β_k as the expected value of $\hat{b}_{\mathrm{ML},k}$, i.e.,

$$\beta_k \triangleq \mathrm{E}\{\hat{b}_{\mathrm{ML},k}\} = \mathrm{E}\{\mathrm{sgn}(x_k)\}\,, \qquad k \in \overline{\mathcal{D}}_{\mathrm{ML}}\,.$$

The "soft decision" $\beta_k = \mathrm{E}\{\mathrm{sgn}(x_k)\}$, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ can be viewed as the counterpart of the hard decision $\hat{b}_{\mathrm{ML},k} = \mathrm{sgn}(x_k)$ that was made for $k \in \mathcal{D}_{\mathrm{ML}}$ in Stage 1. The soft values β_k can be easily calculated from the bounds $L_k(\mathcal{D}_{\mathrm{ML}})$ and $U_k(\mathcal{D}_{\mathrm{ML}})$: using the uniform distribution of x_k, we obtain

$$\begin{aligned}\beta_k &= 1 \cdot \Pr(x_k > 0) + (-1) \cdot \Pr(x_k < 0) \\ &= \frac{L_k(\mathcal{D}_{\mathrm{ML}}) + U_k(\mathcal{D}_{\mathrm{ML}})}{U_k(\mathcal{D}_{\mathrm{ML}}) - L_k(\mathcal{D}_{\mathrm{ML}})}\,, \qquad k \in \overline{\mathcal{D}}_{\mathrm{ML}}\,. \end{aligned} \tag{16}$$

Note that $-1 < \beta_k < 1$. The bounds and, thus, the soft values in (16) are also valid if no ML bits are detected in Stage 1, but the tightness of the bounds and the quality of the soft values improve if more bits are detected.

III. The Sequential Soft-Heuristic Algorithm

The task of Stage 3 is to determine the bits b_k, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ not detected in Stage 1. In [21], a linear or decision-feedback equalizer is used for this task. Here, for improved performance in large MIMO systems, we propose two alternative soft-input heuristic algorithms that make use of the soft values β_k, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ computed in Stage 2. The *sequential soft-heuristic algorithm* (SSA) described in this section is a soft-input version of the greedy algorithm presented in [22]. As in [22], a solution vector is generated in a bit-sequential (recursive) manner by detecting one b_k, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ in each recursion step; the corresponding decision is never reconsidered. However, the SSA employs a different initialization that takes into account the results of Stages 1 and 2. Furthermore, it uses an improved (nongreedy) order of decisions inspired by the Nelder-Mead algorithm [23]. Finally, it performs a continuous update of the soft values via a feedback from Stage 3 to Stage 2.

A. Initialization

The greedy algorithm of [22] (adapted to our bit alphabet $\{-1, 1\}$) uses the zero vector as the initial input vector. In contrast, the initial input vector $\tilde{\mathbf{b}}$ used by the SSA is composed of the ML bits $\hat{b}_{\mathrm{ML},k}$, $k \in \mathcal{D}_{\mathrm{ML}}$ detected in Stage 1 and the soft values β_k, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ calculated in Stage 2, i.e.,

$$\tilde{b}_k \equiv (\tilde{\mathbf{b}})_k = \begin{cases} \hat{b}_{\mathrm{ML},k}\,, & k \in \mathcal{D}_{\mathrm{ML}} \\ \beta_k\,, & k \in \overline{\mathcal{D}}_{\mathrm{ML}}\,. \end{cases} \tag{17}$$

B. Statement of the SSA

Let $\mathcal{D} \supseteq \mathcal{D}_{\mathrm{ML}}$ denote the index set of all bits $\hat{b}_k$ detected so far, which consist of the ML bits $\hat{b}_{\mathrm{ML},k}$ (index set $\mathcal{D}_{\mathrm{ML}}$) and the suboptimum detection results obtained so far in the present Stage 3 (index set $\mathcal{D} \setminus \mathcal{D}_{\mathrm{ML}}$). In each recursion step, the SSA detects one of the as yet undetected bits b_k, $k \in \overline{\mathcal{D}}$. The iterated vector $\overline{\mathbf{b}}$ that provides the input to the recursion step considered is given as (cf. (17))

$$\overline{b}_k = \begin{cases} \hat{b}_k\,, & k \in \mathcal{D} \\ \beta_k\,, & k \in \overline{\mathcal{D}}\,, \end{cases} \tag{18}$$

where the $\hat{b}_k$, $k \in \mathcal{D}$ are the bits detected so far and the β_k are soft values. The SSA now produces a modified vector $\overline{\mathbf{b}}^{(k)}$ in which the soft value β_k contained in $\overline{\mathbf{b}}$ at some $k \in \overline{\mathcal{D}}$ is replaced by a hard bit $\hat{b}_k \in \{-1, 1\}$:

$$\overline{b}_l^{(k)} \equiv (\overline{\mathbf{b}}^{(k)})_l = \begin{cases} \overline{b}_l\,, & l \in \mathcal{D} \text{ or } l \in \overline{\mathcal{D}}_{\sim k} \\ \hat{b}_k\,, & l = k\,. \end{cases} \tag{19}$$

Finally, the index sets are updated according to $\mathcal{D}^{(\mathrm{new})} = \mathcal{D} \cup \{k\}$ and $\overline{\mathcal{D}}^{(\mathrm{new})} = \overline{\mathcal{D}}_{\sim k}$.

It remains to determine the "best" index $k \in \overline{\mathcal{D}}$ and the "best" bit value $\hat{b}_k \in \{-1, 1\}$. Motivated by (6), the greedy strategy [22] chooses the k and $\hat{b}_k$ yielding the largest increase in $f(\cdot)$. The SSA takes a different approach that is inspired by the Nelder-Mead optimization algorithm [23]. First, the k and $\hat{b}_k$ producing the maximum *decrease* of $f(\cdot)$ are determined; then, this "worst decision" is inverted by setting the bit $\hat{b}_k$ to the respective other value. As will be shown in Section V-E, this strategy yields a significantly better performance than the greedy strategy. An intuitive explanation might be that avoiding these worst decisions reduces error propagation.

For a formal statement, we define the *gain function*

$$g_k(\hat{b}_k) \triangleq f\big(\overline{\mathbf{b}}^{(k)}\big) - f(\overline{\mathbf{b}})\,, \qquad k \in \overline{\mathcal{D}}\,, \tag{20}$$

which characterizes the increase in $f(\cdot)$ obtained by replacing $\overline{\mathbf{b}}$ with $\overline{\mathbf{b}}^{(k)}$. Inserting (18) and (19) into (20), with $f(\mathbf{b})$ as given by (5), and using the fact that $\hat{b}_k^2 = 1$ yields

$$\begin{aligned} g_k(\hat{b}_k) = 2(\hat{b}_k - \beta_k)\Bigg(&\Re\{z_k\} - \sum_{l \in \mathcal{D}} \Re\{G_{k,l}\}\hat{b}_l \\ &- \sum_{l \in \overline{\mathcal{D}}_{\sim k}} \Re\{G_{k,l}\}\beta_l \Bigg) - \big(\beta_k^2 - 1\big) G_{k,k}\,, \quad k \in \overline{\mathcal{D}}\,. \end{aligned} \tag{21}$$

A recursion step of the SSA can now be stated as follows.

1) Compute the index $k \in \overline{\mathcal{D}}$ corresponding to the smallest gain:

$$k_{\mathrm{opt}} = \arg\min_{k \in \overline{\mathcal{D}}} \hat{g}_k\,, \qquad \text{with } \hat{g}_k \triangleq \min\big\{g_k(1), g_k(-1)\big\}\,.$$

2) At k_{opt}, choose the bit value with the larger gain, i.e.,

$$\hat{b}_{k_{\mathrm{opt}}} = \begin{cases} 1\,, & g_{k_{\mathrm{opt}}}(1) \geq g_{k_{\mathrm{opt}}}(-1) \\ -1\,, & g_{k_{\mathrm{opt}}}(1) < g_{k_{\mathrm{opt}}}(-1)\,. \end{cases} \tag{22}$$

3) Update the index sets $\mathcal{D}$ and $\overline{\mathcal{D}}$ according to $\mathcal{D}^{(\mathrm{new})} = \mathcal{D} \cup \{k_{\mathrm{opt}}\}$ and $\overline{\mathcal{D}}^{(\mathrm{new})} = \overline{\mathcal{D}}_{\sim k_{\mathrm{opt}}}$.

4) For all $k \in \overline{\mathcal{D}}^{(\mathrm{new})}$, update the bounds $L_k(\mathcal{D})$ and $U_k(\mathcal{D})$ using the update relations (14) and (15), respectively:

$$\begin{aligned} L_k(\mathcal{D}^{(\mathrm{new})}) &= L_k(\mathcal{D}) - 2\big(\Re\{G_{k,k_{\mathrm{opt}}}\}\hat{b}_{k_{\mathrm{opt}}} - |\Re\{G_{k,k_{\mathrm{opt}}}\}|\big) \\ U_k(\mathcal{D}^{(\mathrm{new})}) &= U_k(\mathcal{D}) - 2\big(\Re\{G_{k,k_{\mathrm{opt}}}\}\hat{b}_{k_{\mathrm{opt}}} + |\Re\{G_{k,k_{\mathrm{opt}}}\}|\big)\,. \end{aligned}$$

5) Recalculate soft values $\beta_k^{(\mathrm{new})}$, $k \in \overline{\mathcal{D}}^{(\mathrm{new})}$ from the updated bounds $L_k(\mathcal{D}^{(\mathrm{new})})$ and $U_k(\mathcal{D}^{(\mathrm{new})})$ as follows. If $L_k(\mathcal{D}^{(\mathrm{new})}) \geq 0$, set $\beta_k^{(\mathrm{new})} = 1$; if $U_k(\mathcal{D}^{(\mathrm{new})}) \leq 0$, set $\beta_k^{(\mathrm{new})} = -1$ (this is motivated by the partial ML detection algorithm); otherwise obtain $\beta_k^{(\mathrm{new})}$ from (16),

i.e., $\beta_k^{(\text{new})} = \big[L_k(\mathcal{D}^{(\text{new})}) + U_k(\mathcal{D}^{(\text{new})})\big] / \big[U_k(\mathcal{D}^{(\text{new})}) - L_k(\mathcal{D}^{(\text{new})})\big]$. The fact that after each recursion step (i.e., new decision), the soft values β_k for the as yet undetected bits b_k, $k \in \overline{\mathcal{D}}^{(\text{new})}$ are improved using the updated bounds $L_k(\mathcal{D}^{(\text{new})})$ and $U_k(\mathcal{D}^{(\text{new})})$ can be viewed as a feedback from Stage 3 to Stage 2.

6) Recalculate gains $g_k^{(\text{new})}(1)$ and $g_k^{(\text{new})}(-1)$ for all $k \in \overline{\mathcal{D}}^{(\text{new})}$ according to expression (21), i.e.,

$$g_k^{(\text{new})}(\hat{b}_k) = 2(\hat{b}_k - \beta_k^{(\text{new})}) \Bigg(\Re\{z_k\} - \sum_{l \in \mathcal{D}^{(\text{new})}} \Re\{G_{k,l}\} \hat{b}_l - \sum_{l \in \overline{\mathcal{D}}_{\sim k}^{(\text{new})}} \Re\{G_{k,l}\} \beta_l^{(\text{new})} \Bigg) - \big(\beta_k^{(\text{new})2} - 1\big) G_{k,k}\,, \quad k \in \overline{\mathcal{D}}^{(\text{new})}.$$

This recursion stops when all bits have been detected ($\mathcal{D}^{(\text{new})} = \mathcal{I}, \overline{\mathcal{D}}^{(\text{new})} = \emptyset$).

The SSA detector can be converted into a *soft-output* detector via the approach described in [15]. The resulting soft decisions (of the log-likelihood type) allow the SSA detector to be combined with soft-input channel decoding.

C. Interpretation

An interesting interpretation can be obtained by writing (21) as $g_k(\pm 1) = \pm\big(1 \mp \beta_k\big) A_k + B_k$, with

$$A_k \triangleq 2 \left(\Re\{z_k\} - \sum_{l \in \mathcal{D}} \Re\{G_{k,l}\} \hat{b}_l - \sum_{l \in \overline{\mathcal{D}}_{\sim k}} \Re\{G_{k,l}\} \beta_l \right) \quad (23)$$

and $B_k \triangleq -\big(\beta_k^2 - 1\big) G_{k,k}$. The bit decision (22) can then be written as $\hat{b}_{k_{\text{opt}}} = \text{sgn}(A_{k_{\text{opt}}})$. This is reminiscent of the ML bit decision $\hat{b}_{\text{ML},k} = \text{sgn}\big(\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})\big)$ in (9); however, $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})$ is replaced by $A_{k_{\text{opt}}}$. Comparing the expression of $A_{k_{\text{opt}}}$ in (23) with that of $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})$ (see (8)), we see that the only difference is the fact that some of the ML bits $\hat{b}_{\text{ML},l}$ used in $\psi_k(\hat{\mathbf{b}}_{\text{ML},\sim k})$ are replaced by the suboptimum bits $\hat{b}_l$ or by the soft values β_l.

IV. The Genetic Soft-Heuristic Algorithm

The *genetic soft-heuristic algorithm* (GSA) is an alternative to the SSA with better performance but higher complexity. It is a soft-input version of the genetic optimization algorithm presented in [24], and differs from that algorithm in its initialization (which uses the results of Stages 1 and 2), the local search algorithm, and the mutation operation. Also, it contains a novel diversification operation that uses soft values. Similar to [24], it adds to the genetic operations (crossover, mutation, selection, and diversification [27]) a local search.

A block diagram of the GSA with initialization is shown in Fig. 2. (The individual blocks will be explained in more detail later in this section.) The initialization procedure generates an initial start set $\big\{\mathbf{b}_1^{(1)}, \ldots, \mathbf{b}_{M_{\text{max}}}^{(1)}\big\}$ of candidate solutions (CSs) for the first iteration of the GSA, using the ML bits $\hat{b}_{\text{ML},k}$, $k \in \mathcal{D}_{\text{ML}}$ from Stage 1 and the soft values β_k, $k \in \overline{\mathcal{D}}_{\text{ML}}$ from Stage 2. This is done in two steps: first, a *preliminary* initial start set $\big\{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{M_{\text{max}}}\big\}$ is generated; next, this preliminary set is improved by a local search algorithm.

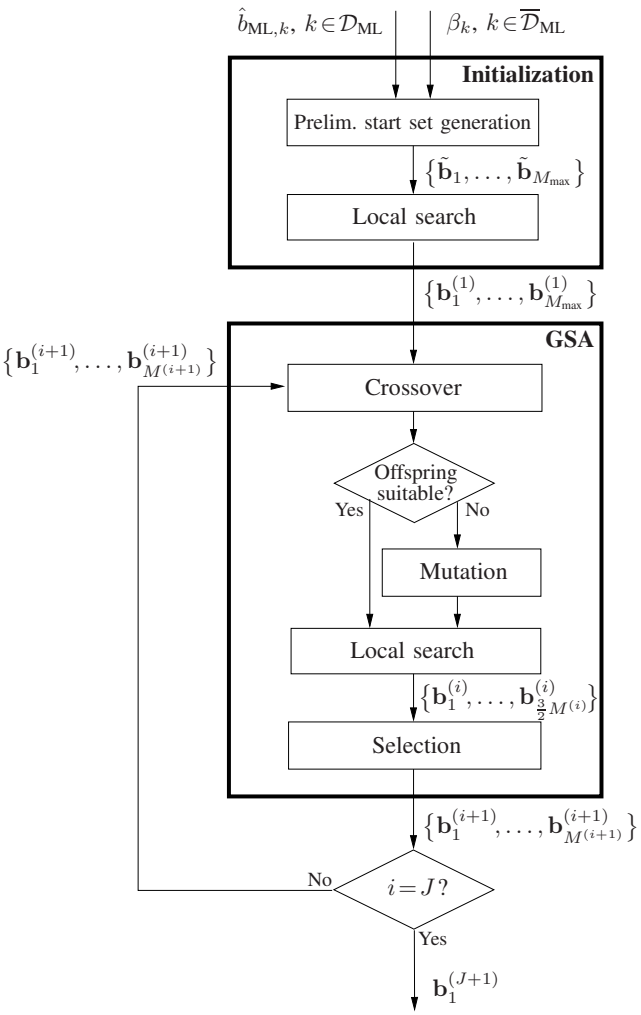


Fig. 2. Block diagram of the GSA with initialization.

In iteration i of the GSA, the *crossover*, *mutation*, and *local search* steps use the locally optimized CSs, $\big\{\mathbf{b}_1^{(i)}, \ldots, \mathbf{b}_{M^{(i)}}^{(i)}\big\}$, to calculate $M^{(i)}/2$ new CSs. Here, $M^{(i)}$ is assumed even for simplicity, with $M^{(i)} \leq M_{\text{max}}$. In the *selection* step, identical CSs in the extended set consisting of the $M^{(i)}$ previous CSs and the $M^{(i)}/2$ additional CSs are removed, and the best $M^{(i+1)} \leq M_{\text{max}}$ CSs—i.e., those with the largest $f(\cdot)$ in (5)—are used as the start set for the next iteration. Hence, the number of CSs in each start set and, therefore, the complexity of each iteration 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 CS set is used as the final result of the GSA. Here, J represents a tradeoff between performance and computing time. However, beyond a certain point, the performance cannot be improved further by increasing J [27]. We will now describe the GSA in more detail.

A. Generation of the Preliminary Initial Start Set

Each CS in the preliminary initial start set $\big\{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{M_{\text{max}}}\big\}$ contains the ML bits $\hat{b}_{\text{ML},k}$, $k \in \mathcal{D}_{\text{ML}}$ detected in Stage 1. The

remaining bits (for $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$) are derived from the soft values β_k, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ calculated in Stage 2 by means of the following modified version of the Chase algorithm [28], which yields high CS diversity in an efficient way. The first CS $\tilde{\mathbf{b}}_1$ of the preliminary initial start set is generated by quantizing the soft values β_k, $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$. Thus, $\tilde{\mathbf{b}}_1$ is given by

$$\tilde{b}_{1,k} \;=\; \begin{cases} \hat{b}_{\mathrm{ML},k}\,, & k \in \mathcal{D}_{\mathrm{ML}} \\ \mathrm{sgn}(\beta_k)\,, & k \in \overline{\mathcal{D}}_{\mathrm{ML}}\,. \end{cases} \tag{24}$$

The remaining CSs $\tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_{M_{\max}}$ are generated by interpreting the absolute values of the soft values β_k as reliability measures and flipping unreliable bits. More precisely, let us denote the indices $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ by $k_1, k_2, \ldots, k_K$, with $K \triangleq \big|\overline{\mathcal{D}}_{\mathrm{ML}}\big|$ and with ordering according to increasing reliability, i.e., $|\beta_{k_1}| \le |\beta_{k_2}| \le \ldots \le |\beta_{k_K}|$. Then $\tilde{\mathbf{b}}_2$ is formed by flipping the two most unreliable bits in $\tilde{\mathbf{b}}_1$:

$$\tilde{b}_{2,k} \;=\; \begin{cases} -\tilde{b}_{1,k}\,, & k \in \{k_1, k_2\} \\ \tilde{b}_{1,k}\,, & k \in \mathcal{I}_{\sim k_1,k_2}\,. \end{cases} \tag{25}$$

Similarly, $\tilde{\mathbf{b}}_3$ is formed by flipping the four most unreliable bits in $\tilde{\mathbf{b}}_1$. Continuing this way, for each new $\tilde{\mathbf{b}}_j$, two more bits—the most unreliable bits of those not flipped so far—are flipped. The elements of the last CS $\tilde{\mathbf{b}}_{M_{\max}}$ of the preliminary initial start set are thus given by

$$\tilde{b}_{M_{\max},k} \;=\; \begin{cases} -\tilde{b}_{1,k}\,, & k \in \{k_1, k_2, \ldots, k_{2\,(M_{\max}-1)}\} \\ \tilde{b}_{1,k}\,, & k \in \mathcal{I}_{\sim k_1,k_2,\ldots,k_{2(M_{\max}-1)}}\,. \end{cases} \tag{26}$$

Here, $M_{\max}$ is a design parameter that satisfies $2\,(M_{\max}-1) \le K$ and is determined empirically.

B. 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 locally optimized initial start set $\{\mathbf{b}_1^{(1)}, \ldots, \mathbf{b}_{M_{\max}}^{(1)}\}$, which serves as the input to the GSA. (The same algorithm is also used to improve the new CSs created by the crossover and mutation operations, see Fig. 2 and Section IV-C.) The local search procedure is executed for every CS individually.

To keep the complexity low, we use the simple *1-opt* algorithm [22]; however, more powerful—and more complex—algorithms like *k-opt* and *randomized k-opt* local search [22], [24] can also be used. In each iteration step, for each existing CS, the 1-opt algorithm attempts to find a better CS in which one bit is flipped. 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.,

$$b_l^{(k)} = \begin{cases} -b_k\,, & l = k \\ b_l\,, & l \in \mathcal{I}_{\sim k}\,, \end{cases} \qquad k \in \overline{\mathcal{D}}_{\mathrm{ML}}\,.$$

Here, the optimum $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ is obtained by maximizing $g_k \triangleq f(\mathbf{b}^{(k)}) - f(\mathbf{b})$. Using (5) and the fact that $b_l^2 = 1$ yields

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

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

1) By an exhaustive search, find the index $k \in \overline{\mathcal{D}}_{\mathrm{ML}}$ with the largest gain, i.e., $k_{\mathrm{opt}} \triangleq \arg\max_{k \in \overline{\mathcal{D}}_{\mathrm{ML}}} g_k$.
2) Flip the corresponding bit in $\mathbf{b}$, i.e., form the new CS $\mathbf{b}^{(k_{\mathrm{opt}})}$ with elements

$$b_l^{(k_{\mathrm{opt}})} = \begin{cases} -b_{k_{\mathrm{opt}}}\,, & l = k_{\mathrm{opt}} \\ b_l\,, & l \in \mathcal{I}_{\sim k_{\mathrm{opt}}}\,. \end{cases}$$

The gain $g_k^{(\mathrm{new})}$ for the next iteration can be easily obtained by updating the g_k according to

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

This iterative process is terminated when all $g_k^{(\mathrm{new})}$ are non-positive, which indicates that no further increase of $f(\cdot)$ can be achieved by flipping a single bit.

C. Crossover, Mutation, Selection

The first GSA stage (see Fig. 2) is the crossover operation. According to the *uniform crossover* algorithm [27], the current CS set $\{\mathbf{b}_1^{(i)}, \ldots, \mathbf{b}_{M^{(i)}}^{(i)}\}$, of size $M^{(i)}$, 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* $\mathbf{b}'$ that inherits those bits 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 CS set is then extended by the offspring CSs. The size of the extended set is given by $M_{\mathrm{ext}}^{(i)} = \lceil (3/2)\, M^{(i)} \rceil$, where $\lceil x \rceil$ denotes the smallest integer not smaller than x.

In the subsequent mutation stage, a given offspring CS $\mathbf{b}'$ generated by crossover is considered unsuitable if the Hamming distance of the parents $\mathbf{b}_j^{(i)}, \mathbf{b}_{j'}^{(i)}$ satisfies $d\big(\mathbf{b}_j^{(i)}, \mathbf{b}_{j'}^{(i)}\big) \le 2$. Indeed, this implies $d\big(\mathbf{b}', \mathbf{b}_j^{(i)}\big) \le 2$ and $d\big(\mathbf{b}', \mathbf{b}_{j'}^{(i)}\big) \le 2$; thus, it is very likely that the bit-flipping modification of $\mathbf{b}'$ by means of 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. (We note that in [24], the minimum Hamming distance $d\big(\mathbf{b}_j^{(i)}, \mathbf{b}_{j'}^{(i)}\big)$ required in order to consider the offspring CS as suitable is much larger than the value of 3 chosen here; also, the number of bits flipped in each mutation step is much larger than 1. However, in our experiments, we obtained better results with our choice of mutation parameters.) Subsequently, all new (additional) CSs created by the crossover and mutation stages are optimized by another local search stage (see Fig. 2).

Finally, the selection stage reduces the extended set of $M_{\mathrm{ext}}^{(i)} = \lceil (3/2)\, M^{(i)} \rceil$ CSs obtained by the local search stage. First, identical CSs are removed. Let $\{\mathbf{b}'_1, \ldots, \mathbf{b}'_Q\}$ with $Q \le M_{\mathrm{ext}}^{(i)}$ denote the resulting CS set. If $Q \le M_{\max}$, 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 largest $f(\mathbf{b}'_j)$ values (hence, $M^{(i+1)} = M_{\max}$). Thus, $M^{(i+1)} = Q$ if $Q \le M_{\max}$ and $M^{(i+1)} = M_{\max}$ if $Q > M_{\max}$; note that

$M^{(i+1)} \leq M_{\max}$ is always satisfied. A similar approach was used in [24].

The final result of the GSA is taken to be the best CS in the CS set $\{\mathbf{b}_1^{(J+1)}, \ldots, \mathbf{b}_{M^{(J+1)}}^{(J+1)}\}$ obtained after the predetermined maximum number J of iterations, i.e., $\mathbf{b}_1^{(J+1)}$. Alternatively, a soft-output version of the GSA detector can be established: using the max-log approximation as described in [26], one can compute from the CS set $\{\mathbf{b}_1^{(J+1)}, \ldots, \mathbf{b}_{M^{(J+1)}}^{(J+1)}\}$ soft information for use with a soft-input channel decoder.

D. Diversification

A performance improvement can be achieved by an optional diversification stage. As shown in Fig. 3, this adds an outer loop to the GSA. Let $\nu \in \{1, 2, \ldots\}$ and the superscript $^{[\nu]}$ denote the iteration index for this outer loop. Furthermore, for $\nu \geq 2$, let $\{\mathbf{b}_1^{[\nu-1]}, \ldots, \mathbf{b}_{M^{(J+1)}}^{[\nu-1]}\}$ denote the CS set obtained at the $(\nu-1)$st outer iteration after termination of the (inner) GSA loop, i.e., at the output of the GSA's selection stage. The diversification stage calculates from $\{\mathbf{b}_1^{[\nu-1]}, \ldots, \mathbf{b}_{M^{(J+1)}}^{[\nu-1]}\}$ new soft values

$$\beta_k^{[\nu]} = \frac{1}{M^{(J+1)}} \sum_{m=1}^{M^{(J+1)}} \left(\mathbf{b}_m^{[\nu-1]}\right)_k , \qquad k \in \overline{\mathcal{D}}_{\text{ML}} .$$

These soft values are then used by the initialization stage of the GSA to calculate a new preliminary initial start set $\{\tilde{\mathbf{b}}_1^{[\nu]}, \ldots, \tilde{\mathbf{b}}_{M_{\max}}^{[\nu]}\}$ for the next (νth) outer iteration. The initialization stage is modified in that the first CS of this new preliminary initial start set is chosen as the best CS obtained from the previous (i.e. $(\nu-1)$st) outer iteration, i.e., the CS from $\{\mathbf{b}_1^{[\nu-1]}, \ldots, \mathbf{b}_{M^{(J+1)}}^{[\nu-1]}\}$ with largest $f(\cdot)$ value. Assuming for concreteness that this best CS is $\mathbf{b}_1^{[\nu-1]}$, we thus have $\tilde{\mathbf{b}}_1^{[\nu]} = \mathbf{b}_1^{[\nu-1]}$. The remaining CSs are constructed by means of the scheme described in Section IV-A, using the new soft values $\beta_k^{[\nu]}$. Let us again denote the indices $k \in \overline{\mathcal{D}}_{\text{ML}}$ by $k_1, k_2, \ldots, k_K$, with $K = \left|\overline{\mathcal{D}}_{\text{ML}}\right|$ and ordered such that $\left|\beta_{k_1}^{[\nu]}\right| \leq \left|\beta_{k_2}^{[\nu]}\right| \leq \ldots \leq \left|\beta_{k_K}^{[\nu]}\right|$. The second CS $\tilde{\mathbf{b}}_2^{[\nu]}$ is then given by (cf. (24))

$$\tilde{b}_{2,k}^{[\nu]} = \begin{cases} \hat{b}_{\text{ML},k} , & k \in \mathcal{D}_{\text{ML}} \\ \operatorname{sgn}\left(\beta_k^{[\nu]}\right) , & k \in \overline{\mathcal{D}}_{\text{ML}} ; \end{cases}$$

the third CS $\tilde{\mathbf{b}}_3^{[\nu]}$ is given by (cf. (25))

$$\tilde{b}_{3,k}^{[\nu]} = \begin{cases} -\tilde{b}_{2,k}^{[\nu]} , & k \in \{k_1, k_2\} \\ \tilde{b}_{2,k}^{[\nu]} , & k \in \mathcal{I}_{\sim k_1, k_2} ; \end{cases}$$

and so on, until the last CS (cf. (26))

$$\tilde{b}_{M_{\max},k}^{[\nu]} = \begin{cases} -\tilde{b}_{2,k}^{[\nu]} , & k \in \{k_1, \ldots, k_{2(M_{\max}-2)}\} \\ \tilde{b}_{2,k}^{[\nu]} , & k \in \mathcal{I}_{\sim k_1, \ldots, k_{2(M_{\max}-2)}} . \end{cases}$$

This outer loop iteration process is initialized at $\nu = 1$ with the preliminary initial start set of Section IV-A, based on the original soft values β_k, $k \in \overline{\mathcal{D}}_{\text{ML}}$. The process is terminated after a predetermined maximum number V of iterations. The best CS at that point, $\mathbf{b}_1^{[V+1]}$, is used as the final result of the extended GSA. Alternatively, soft information for a soft-input channel decoder is computed. The performance improvements achieved by diversification will be demonstrated in Section V.

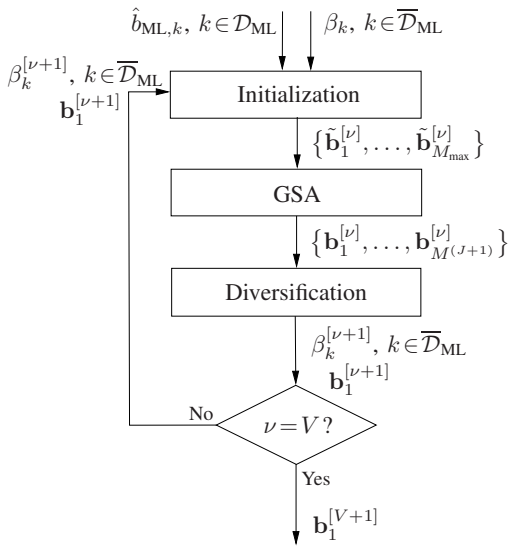


Fig. 3. Block diagram of the extended GSA including a diversification stage and an outer loop.

V. Simulation Results

We present simulation results demonstrating the uncoded BER and computational complexity of the proposed detectors. The MATLAB routines of the proposed detectors are available online at http://www.nt.tuwien.ac.at/about-us/staff/florian-meyer/.

A. Simulation Scenarios and Parameters

Two scenarios are considered: a spatial-multiplexing multiantenna system [1], [5] and an OFDM system with intercarrier interference (ICI) [2], [29]. In the spatial-multiplexing scenario, the channel matrix $\mathbf{H}$ has iid Gaussian entries. In the OFDM/ICI scenario, the MIMO system corresponds to the transmission of a single OFDM symbol consisting of N_t subcarriers over a doubly selective single-antenna channel, with ICI due to the channel's time variation [29]. Thus, the dimension of the MIMO system is $N_t \times N_t$, and the main task of the MIMO detector is a mitigation of the detrimental effects of ICI. The doubly selective fading channel is characterized by a Gaussian wide-sense stationary uncorrelated scattering (WSSUS) model with uniform delay and Doppler profiles (brick-shaped scattering function) [30]. The maximum delay (channel length) is $\tau_{\max} = 9$, the cyclic prefix length is $L_{\text{CP}} = 8$, and the maximum Doppler frequency is 16% of the subcarrier spacing. Because $L_{\text{CP}} = \tau_{\max} - 1$, intersymbol interference is avoided [29]. For each transmit symbol vector $\mathbf{s}$, a new channel realization was randomly generated using the method presented in [31]. The MIMO channel matrix $\mathbf{H}$ depends on the impulse response of the doubly selective fading channel as well as the (rectangular) transmit and receive pulses as described in [32]. The entries of $\mathbf{H}$ are not independent nor identically distributed; they exhibit a strong diagonal dominance and an approximate band structure [29], which leads to an approximate band structure of $\mathbf{A}$.

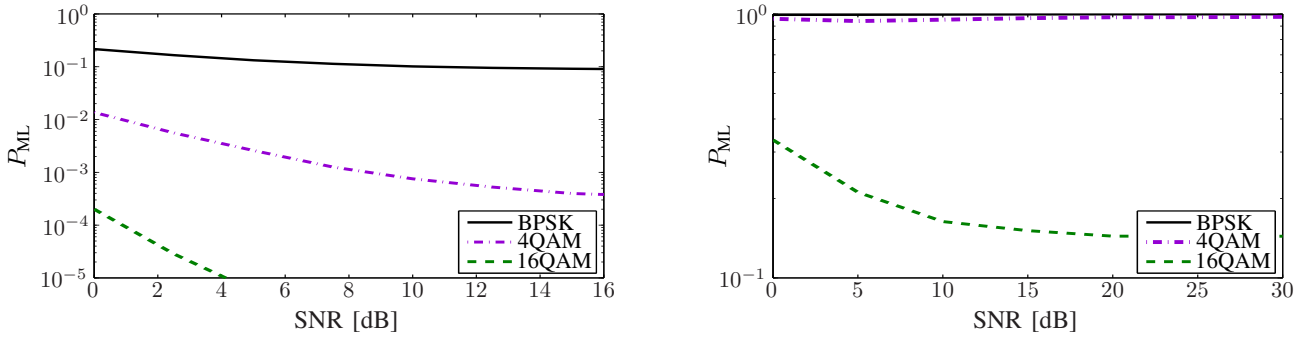


Fig. 4. Detection probability of partial ML detection for (a) a 16×16 spatial-multiplexing system and (b) a 64×64 OFDM/ICI system.

We compare the proposed detectors—hereafter briefly termed "SSA" and "GSA"—with ML detection (2) using the Schnorr-Euchner sphere decoder [33], [34]; the MMSE detector [4]; the NC detector with MMSE nulling vectors and V-BLAST ordering [6] using the efficient implementation described in [35]; an SDR-based detector with rank-one approximation [10]; a three-stage LAS detector [15]; and the SUMIS detector [11]. We did not simulate existing genetic algorithms for MIMO detection, such as [25], [26], since they assume large populations and are therefore infeasible for large MIMO systems. MMSE detection, NC, and SUMIS require an estimate of the noise variance σ_n^2; however, the true value of σ_n^2 was used in our simulations. For the tuning parameter of SUMIS [11], we used $n_s = 4$ for systems of size 8×8 and 16×16 and $n_s = 1$ (the approximation for large MIMO systems [11]) for systems of size 32×32 and 64×64. The BER was measured based on the transmission of 10^6 bits.

In the GSA, the number of CSs in the preliminary initial start set was chosen as a function of the number $|\overline{\mathcal{D}}_{\text{ML}}|$ of undetected bits after partial ML detection according to

$$M_{\max} = \left\lceil 0.8 \frac{|\overline{\mathcal{D}}_{\text{ML}}|}{2} \right\rceil + 1 . \tag{27}$$

This can be shown to imply that $\tilde{\mathbf{b}}_{M_{\max}}$ differs from $\tilde{\mathbf{b}}_1$ in at least 80% of the undetected bits, b_k for $k \in \overline{\mathcal{D}}_{\text{ML}}$. The number of GSA iterations was chosen as $J = 18$ for the spatial-multiplexing case and $J = 9$ for OFDM/ICI. Hereafter, GSA1 and GSA3 denote the GSA using $V = 1$ (no diversification) and $V = 3$ (two diversification steps), respectively.

B. Performance of the Partial ML Detector

First, we study the effectiveness of the partial ML detection stage (Stage 1). For a 16×16 spatial-multiplexing system and a 64×64 OFDM/ICI system, we determined the empirical probability P_{ML} that a bit is detected—no matter if correctly or incorrectly—by the partial ML detection stage. Fig. 4(a) shows P_{ML} versus the SNR for the spatial-multiplexing system using BPSK, 4QAM, and 16QAM. It is seen that P_{ML} decreases dramatically with increasing constellation size; in particular, it is very small for 16QAM. Furthermore, as previously observed in [21] and mentioned in Section II-A, P_{ML} decreases with increasing SNR. Finally, further experiments (not shown) demonstrated a weak decrease of P_{ML} for increasing system size. Fig. 4(b) shows P_{ML} for the OFDM scenario. Here, the equivalent channel matrix $\mathbf{A}$ exhibits an approximately banded (quasi-banded) structure; as observed in Section II-A, this is favorable for the ML detection stage. In fact, P_{ML} is seen to be much higher than in the spatial-multiplexing scenario: it is approximately 1 for BPSK and 4QAM and above 0.1 for 16QAM, for all considered SNRs. This shows that the effectiveness of the partial ML detection stage is very different in the two scenarios: in the OFDM scenario (quasi-banded structure), the partial ML detection stage is able to detect almost all or many bits; in the spatial-multiplexing scenario, it is just a preprocessing stage that improves the performance only slightly.

C. BER Performance in Spatial-Multiplexing Systems

Next, we assess the BER-versus-SNR performance of the proposed detectors (SSA, GSA1, GSA3) and the state-of-the-art detectors (NC, SDR, LAS, SUMIS, ML) in a spatial-multiplexing system. MMSE-based detection was not considered here because of its poor performance in large spatial-multiplexing systems. Fig. 5(a)–(c) shows results obtained for BPSK and system dimensions 8×8, 16×16, and 64×64. It is seen that at low and medium SNRs, the performance of the proposed detectors (SSA and GSA) tends to be very similar, close to that of SDR, LAS, SUMIS, and ML detection, and better than that of NC. (The ML detector is not included in Fig. 5(c) since it is too complex to simulate for the 64×64 system.) We note that for all three system dimensions, we observed the BER performance of GSA3 (not shown) to be effectively equal to that of GSA1.

Fig. 5(d)–(f) shows results for 4QAM. For all system dimensions, SSA performs well only at low SNRs and exhibits an error floor at high SNRs. This can be explained by the fact (observed in Section V-B) that, for the spatial-multiplexing scenario using 4QAM, only few bits are detected by Stage 1 at higher SNRs. This results in a poor quality of the soft values, to which SSA is very sensitive. The performance of GSA is more satisfactory. In fact, GSA1 outperforms the other suboptimum detectors—including SDR and SUMIS—for the 8×8 system at low and medium SNRs and for the 16×16 and 64×64 systems at all displayed SNRs. For the 16×16

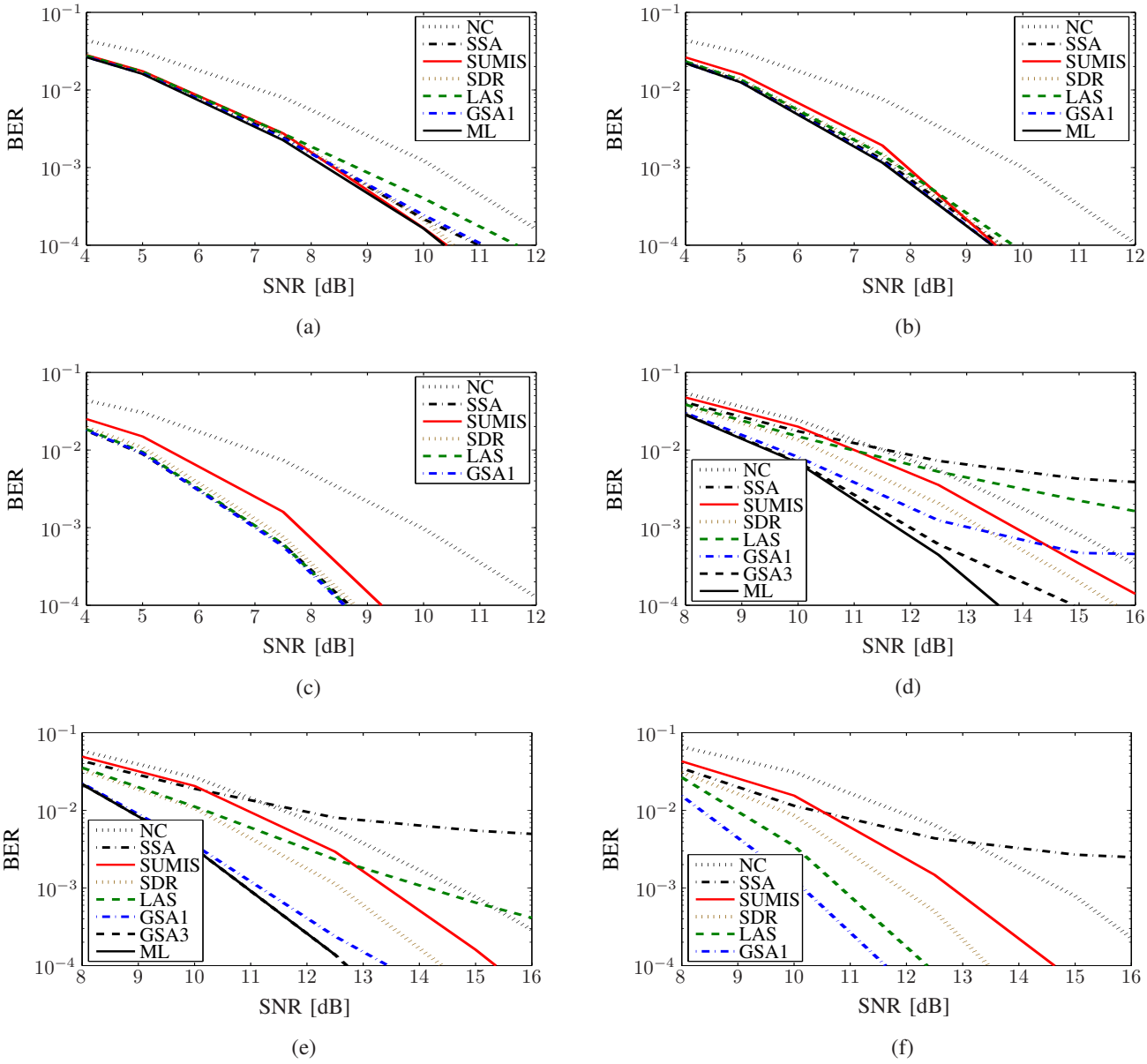


Fig. 5. BER of various detectors for spatial-multiplexing systems: (a)–(c) BPSK, dimension (a) 8×8, (b) 16×16, and (c) 64×64; (d)–(f) 4QAM, dimension (d) 8×8, (e) 16×16, and (f) 64×64.

system, GSA1 performs close to the ML detector at low and medium SNRs, while GSA3 performs close to the ML detector at all displayed SNRs. (Again, the ML detector was not simulated for the 64×64 system.) The performance advantage of GSA over NC, SDR and SUMIS increases with growing system dimension. GSA3 tends to outperform GSA1 at high SNRs; this effect is strongest for small system dimension and disappears in the 64×64 system, where GSA1 and GSA3 (not shown) perform equally well. Hence, diversification is most helpful for small systems. For the 8×8 system, GSA exhibits an error floor at high SNRs. The SNR range of the error floor depends on the number of CSs used in the initialization procedure, and thus, according to (27), also on $|\overline{\mathcal{D}}_{\mathrm{ML}}|$ (which increases with $N_t B = N_t \log_2 |\mathcal{S}|$). For the 16×16 and 64×64 systems, the error floor occurs at SNRs higher than 16 dB, which are not shown in Fig. 5.

For spatial-multiplexing systems using QAM constellations larger than 4QAM, the proposed detectors perform worse than NC. This is due to the poorer quality of the soft values for increasing constellation size.

Further experiments (not shown) demonstrated that the performance of a complexity-limited sphere decoder [34] is poor in large MIMO systems if the abort condition of the sphere decoder is chosen in such a way that the resulting complexity is comparable with that of the proposed detectors.

D. BER Performance in OFDM/ICI Systems

For the OFDM/ICI scenario, Fig. 6 shows the BER-versus-SNR performance obtained with BPSK and 4QAM in a 64×64 system. (We did not consider smaller systems because these are less relevant in the OFDM/ICI scenario.) It is seen that the

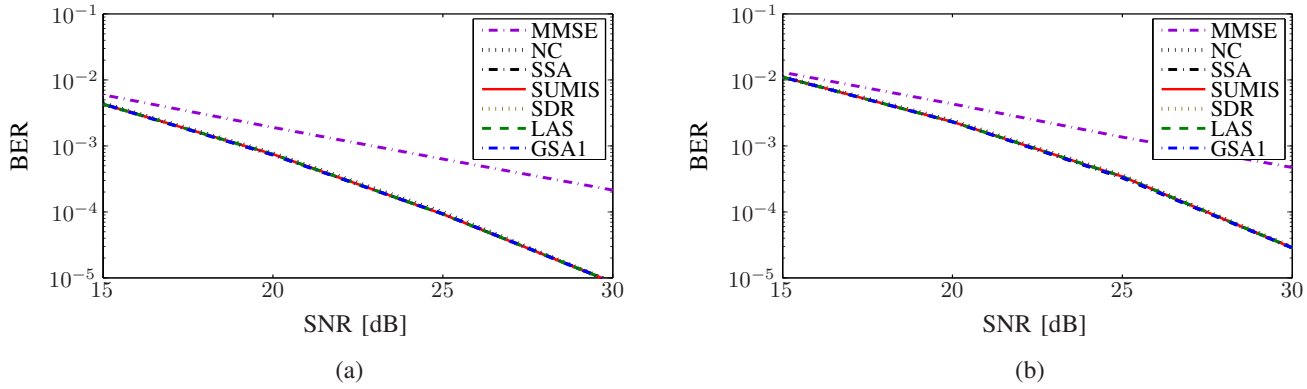


Fig. 6. BER of various detectors for OFDM/ICI systems of dimension 64×64 using (a) BPSK and (b) 4QAM.

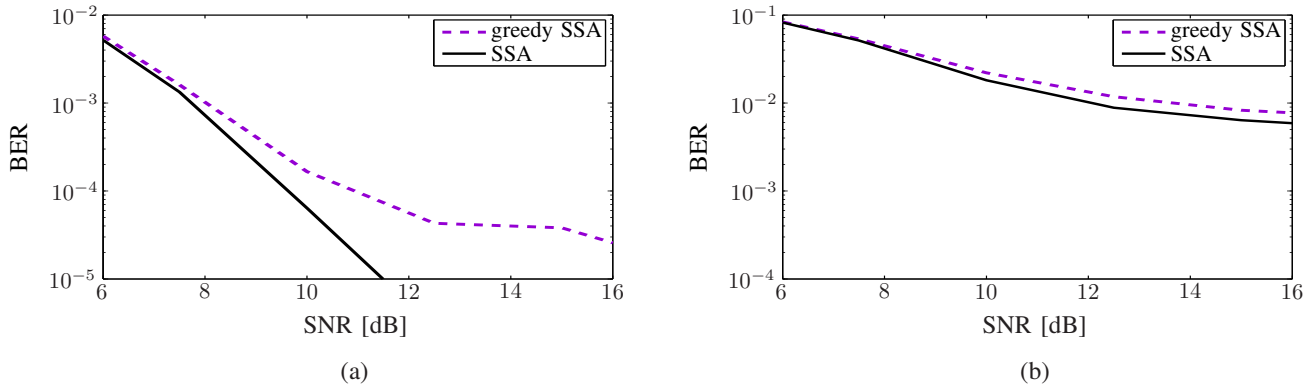


Fig. 7. BER of SSA and of SSA with greedy ordering for a 16×16 spatial-multiplexing system using (a) BPSK and (b) 4QAM.

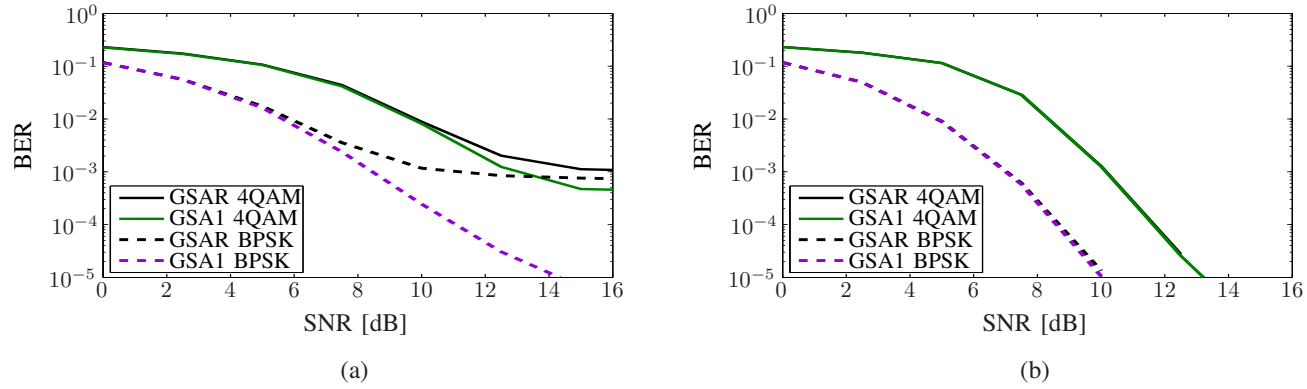


Fig. 8. BER of GSA1 and GSAR for spatial-multiplexing systems using BPSK and 4QAM, of dimension (a) 8×8 and (b) 64×64.

BER curves of all detectors coincide, with the exception of MMSE, which at higher SNRs performs significantly worse than the other detectors. On the other hand, because of the high detection rate P_{ML} of Stage 1 (which is due to the quasi-banded structure of $\mathbf{A}$, cf. Section V-B), the complexity of SSA and GSA1 is significantly smaller than that of LAS and SUMIS, and also much smaller than in the spatial-multiplexing case (cf. Section V-F). (For other suboptimum detectors in the OFDM/ICI scenario that also exploit the quasi-banded structure, see [29], [36] and references therein.) The BER performance of GSA3 (not shown) was observed to be effectively equal to that of GSA1.

E. Further Experiments

To verify the advantage of the Nelder-Mead ordering employed by SSA over the greedy ordering of [22], we compare in Fig. 7 the BER-versus-SNR performance of SSA and an SSA version using the greedy ordering. A 16×16 spatial-multiplexing system using BPSK and 4QAM is considered. As can be seen, the Nelder-Mead ordering outperforms the greedy ordering for SNRs larger than about 8 dB.

To assess the importance of Stages 1 and 2 in GSA detection, Fig. 8 compares the BER-versus-SNR performance of GSA1 and a GSA1 detector in which Stages 1 and 2 are

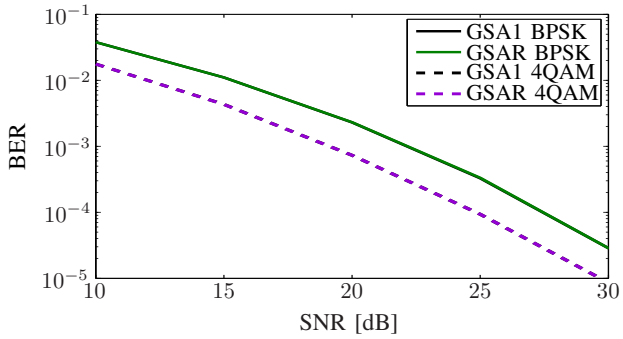


Fig. 9. BER of GSA1 and GSAR for 64×64 OFDM/ICI systems using BPSK and 4QAM. The GSA1 and GSAR curves coincide for each of the two modulation formats.

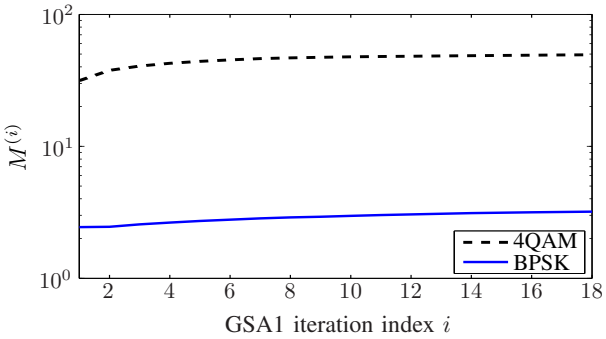


Fig. 10. Average number of candidate solutions $M^{(i)}$ at the beginning of GSA1 iteration i for 64×64 spatial-multiplexing systems using BPSK and 4QAM.

replaced by a random initialization. This latter detector, briefly termed GSAR, is initialized with $M_{\max} = \lceil 0.8 \frac{BN_t}{2} \rceil + 1$ randomly chosen bit vectors of length BN_t in which all bits are drawn iid with 0 and 1 equally likely; these vectors replace the output of the initialization stage of GSA1 (upper box in Fig. 2). Spatial-multiplexing systems of dimension 8×8 and 64×64 using BPSK and 4QAM are considered. It is seen that GSAR performs worse than GSA1 for the 8×8 system but almost equally well as GSA1 for the 64×64 system. Thus, for spatial-multiplexing systems, Stages 1 and 2 in GSA1 are most important for small system dimensions. Fig. 9 shows that for 64×64 OFDM/ICI systems, the performance of GSA1 and GSAR is identical. However, as will be seen in Section V-F and Section V-G, the complexity of GSA1 is smaller than that of GSAR. This is due to the "smart initialization" by Stages 1 and 2, which is very effective for OFDM/ICI systems because of the high quality of the soft values produced by Stage 2 (cf. Section V-B).

Finally, for GSA1, Fig. 10 shows the number of candidate solutions $M^{(i)}$ (averaged over different simulations with SNR values varied from 0 dB to 20 dB in steps of 5 dB) at the beginning of iteration i for a 64×64 spatial-multiplexing system using BPSK or 4QAM. For the BPSK system, the number of candidate solutions is rather small. This indicates that there are only few local maxima, and thus the "searching in parallel" approach of GSA1 cannot exploit its full potential. This agrees with Fig. 5(c), which shows that the performance advantage of GSA1 over SSA is very small. However, for the 4QAM system, the number of candidate solutions (each corresponding to a local maximum, cf. Section IV-B) is increased and, as can be seen in Fig. 5(f), GSA1 exhibits excellent performance.

F. Computational Complexity for Spatial-Multiplexing Systems

Table I presents estimates of the complexity (kflop count) of the different detectors for spatial-multiplexing systems of dimension $N_t = N_r \in \{8, 16, 32, 64\}$ using 4QAM. The complexities of the proposed detectors, of LAS, and of ML detection depend on the SNR. Therefore, we averaged the kflop counts over different simulations with SNR values varied from 0 dB to 20 dB in steps of 5 dB. The kflop count was obtained by means of the Lightspeed toolbox [37] 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 these kflop estimates should be interpreted with caution as they are implementation-dependent. GSA1, GSA3, and GSAR used $J = 18$ GSA iterations. The kflop counts for NC and SUMIS were calculated as described in [35, Section V] and [11, Section VII], respectively. In Table I, 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 for each received vector $\mathbf{y}$ (termed "vector complexity").

From Table I, it is seen that the preparation complexity of all proposed detectors (SSA, GSA1, GSA3) is equal, and it is smaller than that of the other suboptimum detectors (MMSE, NC, LAS, and SUMIS). The vector complexity of SSA is much higher than that of MMSE and NC, comparable to that of SUMIS, and much lower than that of LAS. The vector complexity of GSA1 is higher than that of MMSE, NC, SSA, LAS, and SUMIS; this is the price paid for the better performance of GSA1. (The vector complexity of SDR can be expected to be slightly lower than that of GSA1.) However, the preparation complexity of GSA1 is significantly lower than that of MMSE, NC, and LAS and slightly lower than that of SUMIS. Furthermore, the vector complexity of GSA1 and of GSA3 is significantly lower than that of ML detection (sphere decoding) for system dimension 16×16 or larger. (The vector complexity of the sphere decoder for the 32×32 and 64×64 systems is not shown in Table I because of the excessive computational cost.) As expected, the vector complexity of GSA3 is approximately three times that of GSA1.

The vector complexity of GSAR is higher than that of GSA1. This is caused by a slower convergence of GSAR due to its random initialization. Thus, the first two stages in GSA1, in addition to improving the BER performance as discussed in Section V-E, also result in a reduced complexity.

For GSA1 and GSA3, 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 LAS [15], which

PREPARATION COMPLEXITY (kflops per block preparation)									
$N_t = N_r$	MMSE	NC	LAS	SUMIS	SSA	GSA1	GSA3	GSAR	ML
8	15	24	18	12	4	4	4	4	3
16	110	190	146	66	32	32	32	32	22
32	834	1504	1156	270	262	262	262	262	175
64	6480	11958	9210	2130	2097	2097	2097	2097	1398

VECTOR COMPLEXITY (kflops per received vector)									
$N_t = N_r$	MMSE	NC	LAS	SUMIS	SSA	GSA1	GSA3	GSAR	ML
8	1	1	28	7	10	41	122	56	23
16	3	5	233	41	61	341	1008	460	7603
32	12	20	1914	295	377	2844	9274	3965	–
64	49	81	15601	2228	2557	25038	82897	30985	–

TABLE I
Computational complexity (in kflops) of various detectors for spatial-multiplexing systems using 4QAM, with different dimensions $N_t = N_r$.

VECTOR COMPLEXITY (kflops per received vector)									
$N_t = N_r$	MMSE	NC	LAS	SUMIS	SSA	GSA1	GSA3	GSAR	ML
8	1	1	28	7	2	2	4	10	13
16	3	5	222	41	9	9	16	70	3061
32	12	20	1800	295	34	42	944	564	–
64	49	81	14600	2228	135	302	8458	4947	–

TABLE II
Computational complexity (in kflops) of various detectors for OFDM/ICI systems using 4QAM, with different dimensions $N_t = N_r$. (The preparation complexity is not shown because it equals that shown in Table I for the respective dimension.)

is known to scale as $\mathcal{O}(N^3)$, whereas the scaling behavior of SDR-based detection is, in our case, $\mathcal{O}(N^{7/2})$ [10]. The preparation and vector complexities of NC [6] scale as $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$, respectively. The vector complexity of SUMIS scales as $\mathcal{O}(N^3)$ (assuming that the tuning parameter n_s is small in the sense that $|\mathcal{S}|^{n_s} \ll N^2$). For the SSA, doubling N results in only an approximately 6-fold increase in the vector complexity. Further experiments (not shown) suggest that the scaling behavior of all proposed detectors with respect to the constellation size is approximately cubic.

G. Computational Complexity for OFDM/ICI Systems

In Table II, we present complexity estimates for OFDM/ICI systems of dimension $N_t = N_r \in \{8, 16, 32, 64\}$, again using 4QAM. The preparation complexity is not shown because it is equal to that obtained for the corresponding spatial-multiplexing system of equal dimension (see Table I). Here, we used only $J = 9$ GSA iteration steps for GSA1, GSA3, and GSAR.

The vector complexity of SSA and GSA1 is seen to be much smaller than that of all other detectors except MMSE and NC. It is also much smaller than in the spatial-multiplexing case shown in Table I. In both cases, the main reason is the very effective and efficient partial ML detection performed in Stage 1. In fact, as previously observed in Section V-B, Stage 1 is able to make ML decisions for nearly all bits. This is a major difference from the spatial-multiplexing case. The vector complexity of GSA3 is smaller than that of LAS for all considered system dimensions, and smaller than that of SUMIS for dimensions $N = 8$ and $N = 16$. It is furthermore smaller than in the spatial-multiplexing case, for the reasons mentioned above. Even the vector complexity of the randomly initialized GSAR is considerably smaller than in the spatial-multiplexing case. The reason is that the diagonally dominant channel matrix $\mathbf{H}$ results in a simpler minimization problem, which yields a faster convergence of the GSA iteration. In contrast to the proposed detectors, the other suboptimum detectors (MMSE, NC, LAS, and SUMIS) do not exploit diagonal dominance of $\mathbf{H}$ for a reduction of the vector complexity. Thus, their vector complexity is as large or nearly as large as in the spatial-multiplexing case. Finally, the vector complexity of GSA1 and GSA3 is significantly lower than that of ML (sphere decoding).

VI. CONCLUSION

We presented low-complexity bit-level detectors for MIMO systems employing a BPSK or QAM constellation. The detectors combine efficient partial ML detection [21], generation of soft values, and a novel type of suboptimum detection based on heuristic optimization and soft values ("soft-heuristic optimization"). We proposed two alternative soft-heuristic algorithms, the sequential soft-heuristic algorithm (SSA) and the genetic soft-heuristic algorithm (GSA). Due to their architecture and their use of efficient techniques for high-dimensional optimization, the SSA and GSA are especially advantageous for large MIMO systems. Moreover, their ability to exploit diagonal dominance of the channel matrix for a complexity reduction makes them attractive for ICI mitigation in OFDM systems.

We evaluated the performance of the SSA and GSA for spatial-multiplexing multiantenna systems and OFDM/ICI systems. In spatial-multiplexing systems using BPSK, the SSA and GSA outperform MMSE detection and nulling-and-canceling (NC) and perform similar as semidefinite relaxation (SDR) based detection, likelihood ascent search (LAS) based detection, and the SUMIS detector. For 4QAM, the SSA fails to perform satisfactorily whereas the GSA outperforms MMSE, NC, SDR, LAS, and SUMIS at low-to-medium SNRs and, for larger systems, at all considered SNRs. The SSA is less complex than SDR and LAS and has a better scaling behavior, whereas the GSA has a higher complexity than the other suboptimum detectors.

In OFDM/ICI systems, the SSA and GSA significantly outperform MMSE detection. Similarly to NC, SDR, LAS, and SUMIS, they achieve effectively optimum (ML) performance for BPSK and 4QAM. Furthermore, they are significantly less complex than all other suboptimum detectors considered except MMSE and NC.

Possible directions for further research include the use of other high-dimensional optimization techniques within the proposed detector architecture and the development of soft-input/soft-output versions of the SSA and GSA for use in iterative (turbo-like) receivers.

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] 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.

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

[5] 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.

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

[7] D. Seethaler, H. Artes, and F. Hlawatsch, "Dynamic nulling-and-canceling for efficient near-ML decoding of MIMO systems," *IEEE Trans. Signal Process.*, vol. 54, pp. 4741–4752, Dec. 2006.

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

[9] J. Jaldén, D. Seethaler, and G. Matz, "Worst- and average-case complexity of LLL lattice reduction in MIMO wireless systems," in *Proc. IEEE ICASSP '08*, Las Vegas, NV, pp. 2685–2688, Apr. 2008.

[10] 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.

[11] M. Čirkić and E. G. Larsson, "SUMIS: A near-optimal soft-ouput MIMO detector at low and fixed complexity," 2013. arXiv:1207.3316 [cs.IT].

[12] D. W. Waters and J. R. Barry, "The Chase family of detection algorithms for multiple-input multiple-output channels," *IEEE Trans. Signal Process.*, vol. 56, pp. 739–747, Feb. 2008.

[13] T. Abrão, L. de Oliveira, F. Ciriaco, B. Angélico, P. Jeszensky, and F. Casadevall Palacio, "S/MIMO MC-CDMA heuristic multiuser detectors based on single-objective optimization," *Wireless Personal Comm.*, vol. 53, pp. 529–553, Jun. 2010.

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

[15] 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.

[16] 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.

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

[18] 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.

[19] P. Švač, F. Meyer, E. Riegler, and F. Hlawatsch, "Low-complexity detection for large MIMO systems using partial ML detection and genetic programming," in *Proc. IEEE SPAWC '12*, Çeşme, Turkey, pp. 585–589, Jun. 2012.

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

[21] 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.

[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] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer J.*, pp. 308–313, Jan. 1965.

[24] K. Katayama, M. Tani, and H. Narihisa, "Solving large binary quadratic programming problems by effective genetic local search algorithm," in *Proc. GECCO '00*, Las Vegas, NV, pp. 643–650, Jul. 2000.

[25] 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.

[26] 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.

[27] 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.

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

[29] L. Rugini, P. Banelli, and G. Leus, "OFDM communications over time-varying channels," in *Wireless Communications over Rapidly Time-Varying Channels* (F. Hlawatsch and G. Matz, eds.), ch. 7, pp. 285–336, Academic Press, 2011.

[30] G. Matz and F. Hlawatsch, "Fundamentals of time-varying communication channels," in *Wireless Communications over Rapidly Time-Varying Channels* (F. Hlawatsch and G. Matz, eds.), ch. 1, pp. 1–63, Academic Press, 2011.

[31] D. Schafhuber, G. Matz, and F. Hlawatsch, "Simulation of wideband mobile radio channels using subsampled ARMA models and multistage interpolation," in *Proc. SSP '01*, Singapore, pp. 571–574, Aug. 2001.

[32] W. Kozek and A. F. Molisch, "Nonorthogonal pulseshapes for multicarrier communications in doubly dispersive channels," *IEEE J. Sel. Areas Comm.*, vol. 16, pp. 1579–1589, Oct. 1998.

[33] 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.

[34] 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.

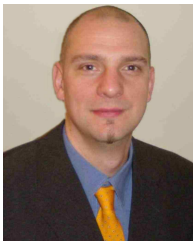
[35] J. Benesty, Y. Huang, and J. Chen, "A fast recursive algorithm for optimum sequential signal detection in a BLAST system," *IEEE Trans. Signal Process.*, vol. 51, pp. 1722–1730, Jul. 2003.

[36] P. Schniter, S.-J. Hwang, S. Das, and A. P. Kannu, "Equalization of time-varying channels," in *Wireless Communications over Rapidly Time-Varying Channels* (F. Hlawatsch and G. Matz, eds.), ch. 6, pp. 237–283, Academic Press, 2011.

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

Pavol Švač (M'03) received the Ing. (M. Eng.) degree in electrical engineering (electronics and telecommunications) and the Ph.D. degree in electrical engineering (telecommunications) from the Technical University of Košice, Slovakia in 2001 and 2008, respectively. From 2001 to 2008, he was with Siemens PSE, Slovakia, where he worked in the field of simulation and standardization of mobile communication systems. From 2008 to 2010, he was a Research Assistant with the Institute of Telecommunications, Vienna University of Technology, Vienna, Austria. Since 2010, he has been with SkyToll, Slovakia. His research interests include statistical signal processing and wireless communications.

Florian Meyer (S'12) received the Dipl.-Ing. (M.Sc.) degree in electrical engineering from Vienna University of Technology, Vienna, Austria in 2011. Since 2011, he has been a Research and Teaching Assistant with the Institute of Telecommunications, Vienna University of Technology, Vienna, Austria, where he is working toward the Ph.D. degree. His research interests include MIMO detection, localization and tracking, signal processing for wireless sensor networks, message passing algorithms, and finite set statistics.



Erwin Riegler (M'07) received the Dipl.-Ing. degree in Technical Physics (with distinction) in 2001 and the Dr. techn. degree in Technical Physics (with distinction) in 2004, both from Vienna University of Technology, Vienna, Austria. From 2005 to 2006, he was a post-doctoral researcher with the Institute for Analysis and Scientific Computing, Vienna University of Technology. From 2007 to 2010, he was a senior researcher with the Telecommunications Research Center Vienna (FTW), Vienna, Austria. Since 2010, he has been a senior researcher with the Institute of Telecommunications, Vienna University of Technology. He was a visiting researcher with the Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany (Sep. 2004 – Feb. 2005), the Communication Theory Group at ETH Zürich, Switzerland (Sep. 2010 – Feb. 2011 and Jun. 2012 – Nov. 2012), and the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, OH, USA (Mar. 2012). His research interests include noncoherent communications, machine learning, interference management, large system analysis, and transceiver design.

Franz Hlawatsch (S'85–M'88–SM'00–F'12) received the Diplom-Ingenieur, Dr. techn., and Univ.-Dozent (habilitation) degrees in electrical engineering/signal processing from Vienna University of Technology, Vienna, Austria in 1983, 1988, and 1996, respectively. Since 1983, he has been with the Institute of Telecommunications, Vienna University of Technology. During 1991–1992, as a recipient of an Erwin Schrödinger Fellowship, he was a visiting researcher with the Department of Electrical Engineering, University of Rhode Island, Kingston, RI, USA. In 1999, 2000, and 2001, he held one-month visiting professor positions with INP/ENSEEIHT, Toulouse, France and IRCCyN, Nantes, France. He (co)authored a book, two review papers that appeared in the IEEE SIGNAL PROCESSING MAGAZINE, about 200 refereed scientific papers and book chapters, and three patents. He coedited the books *Time-Frequency Analysis: Concepts and Methods* (London: ISTE/Wiley, 2008) and *Wireless Communications over Rapidly Time-Varying Channels* (New York: Academic, 2011). His research interests include signal processing for wireless communications and sensor networks, statistical signal processing, and compressive signal processing.

Prof. Hlawatsch was Technical Program Co-Chair of EUSIPCO 2004 and served on the technical committees of numerous IEEE conferences. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2003 to 2007 and for the IEEE TRANSACTIONS ON INFORMATION THEORY from 2008 to 2011. From 2004 to 2009, he was a member of the IEEE SPS Technical Committee on Signal Processing for Communications and Networking. He is coauthor of papers that won an IEEE Signal Processing Society Young Author Best Paper Award and a Best Student Paper Award at IEEE ICASSP 2011.