Abstract—Compute-and-forward (C&F) relaying usually requires channel state information (CSI) at the receivers so that an “optimal” scale factor can be computed for the purposes of decoding. In this paper, a blind C&F scheme—i.e., one not requiring CSI—is developed. Rather than attempting to compute the optimal scale factor, this new scheme seeks one (or more) “good” scalars, i.e., scalars which allow correct decoding despite possibly being sub-optimal. The region of all such good scalars is characterized. To find a good scalar, a computationally efficient scheme, involving error-detection and a hierarchically organized list, is proposed. Simulation results show that this blind C&F scheme achieves—for a class of lattices admitting an efficient trellis decoder—the same throughput as its CSI-enabled counterpart, at the expense of, approximately, a ten-fold increase in computational complexity in the high-throughput region.

I. INTRODUCTION

Compute-and-forward (C&F) is a new relaying strategy proposed by Nazer and Gastpar [1]. It enables intermediate nodes to recover linear combinations of concurrently transmitted messages. As shown in [1], for some AWGN wireless relay networks, C&F outperforms other relaying strategies (such as compress-and-forward and decode-and-forward) in certain SNR region when channel state information (CSI) is known at the receivers. If CSI is also known at the transmitters, the performance of C&F can be greatly improved, achieving the full degrees of freedom for certain network scenarios [2].

In C&F, an intermediate node receives $y = \sum_\ell h_\ell x_\ell + z$, where $h_\ell$ are channel gains, and $x_\ell$ are points in a multidimensional lattice. Based on the fact that any integer combination of lattice points is again a lattice point, the intermediate node selects integer coefficients $a_\ell$ and a scalar $\alpha$, and then attempts to decode the lattice point $\sum_\ell a_\ell x_\ell$ from the scaled signal

$$\alpha y = \sum_\ell a_\ell h_\ell x_\ell + \alpha z = \sum_\ell a_\ell x_\ell + \sum_{\ell}(a_\ell h_\ell - a_\ell) x_\ell + \alpha z. \quad (1)$$

The scalar $\alpha$ and integer coefficients $a_\ell$ are carefully chosen based on channel gains $h_\ell$ so that the effective noise $n$ is made (in some sense) small. Hence, the “optimal” scalar $\alpha$ and integer coefficients $a_\ell$ depend critically on CSI.

In this paper, we aim to eliminate the need of CSI in C&F. We consider the case when no CSI is available, hereafter called blind C&F. This is motivated by the fact that the requirement of CSI (even if only at the receivers) may dominate communication when the number of concurrent transmissions is large, since channel training is known to effectively reduce the degrees of freedom of a wireless link [3].

The basic idea of our approach to blind C&F is simple. Although the optimal scalar is nearly impossible to acquire without CSI, some “good” scalars (that allow correct decoding of linear combinations) can be obtained with a reasonable effort. In particular, when the lattices are asymptotically-good (in the sense of [4]), we are able to characterize the region of all such good scalars, showing that it is bounded, symmetric, and consisting of a union of disks. Based on these properties, we propose a generic blind C&F scheme that makes use of error-detection codes to find a good scalar.

To control the computational complexity of our blind C&F scheme, we propose two strategies that are complementary to each other. In particular, we show that, if a blind C&F scheme admits Viterbi decoding, then its complexity can be significantly reduced. For instance, the complexity can be made around 10 times the complexity of coherent C&F in high-throughput region, as observed from our simulation results.

II. COHERENT COMPUTE-AND-FORWARD

In this section, we briefly review coherent C&F, which serves as a natural benchmark for blind C&F. For ease of presentation, we focus on a single building block of coherent C&F, namely, a system of $L$ concurrent transmitters and a single receiver.

In such a system, each transmitter $\ell$ sends a length-$n$ complex vector $x_\ell \in \mathbb{C}^n$, which satisfies an average power constraint $E[\|x_\ell\|^2] \leq nP$. The receiver observes $y = \sum_{\ell=1}^L h_\ell x_\ell + z$, where $h_\ell \in \mathbb{C}$ are complex-valued channel gains and $z$ is i.i.d. circularly-symmetric complex Gaussian noise, i.e., $z \sim \mathcal{CN}(0, N_0 I_n)$. The goal of the receiver is to reliably recover a linear combination of the transmitted messages based on the received signal $y$ and the channel gains $h_\ell$, which are assumed to be perfectly known at the receiver.

Nazer and Gastpar [1] proposed an effective coding scheme for the above coherent C&F system. Their scheme makes use of asymptotically-good lattice partitions constructed by Erez and Zamir [4]. Since asymptotically-good lattice partitions require very long block lengths and almost unbounded complexity, several practical C&F schemes have been developed (e.g., [5]–[8]) to overcome this difficulty. Here, we present a generic coherent C&F scheme following the work of [5].

The work of D. Silva was partly supported by CNPq-Brazil.
We first introduce a class of complex lattices called $R$-lattices. Let $R$ be a discrete subring of $\mathbb{C}$ forming a principle ideal domain. Typical examples include Gaussian integers $\mathbb{Z}[i] = \{a + bi : a, b \in \mathbb{Z}\}$ and Eisenstein integers $\mathbb{Z}[\omega] = \{a + b\omega : a, b \in \mathbb{Z}, \omega = e^{2\pi i/3}\}$. For any $N$ given linearly independent basis vectors $g_1, \ldots, g_N$ in $\mathbb{C}^n$, the $R$-lattice $\Lambda$ (generated by them) is defined as the set of all $R$-linear combinations of these vectors, i.e., $\Lambda = \{\sum_i r_i g_i : r_i \in R\}$. We call $N$ the rank of $\Lambda$, and $n$ the dimension of $\Lambda$. In general, we have $N \leq n$. When $N = n$, the lattice $\Lambda$ is called a full-rank lattice. Throughout this paper, we focus on full-rank lattices.

An $R$-sublattice $\Lambda'$ of $\Lambda$ is a subset of $\Lambda$ which is itself an $R$-lattice. The set of all the cosets of $\Lambda'$ in $\Lambda$, denoted by $\Lambda/\Lambda' \triangleq \{\lambda + \Lambda' : \lambda \in \Lambda\}$, forms a partition of $\Lambda$, hereafter called an $R$-lattice partition. Without loss of generality, we may assume that there is a one-to-one linear map between a lattice partition $\Lambda/\Lambda'$ and a message space $W$ (see [5] for such a construction).

A nearest-lattice-point (NLP) decoder is a map $D_\Lambda : \mathbb{C}^n \rightarrow \Lambda$ that sends a point $x \in \mathbb{C}^n$ to a nearest lattice point in Euclidean distance, i.e.,

$$D_\Lambda(x) = \text{arg min}_{\lambda \in \Lambda} \|x - \lambda\|. \quad (2)$$

An element $x$ in a coset $\lambda + \Lambda'$ is called a coset leader if $\|x\| \leq \|y\|$ for all $y \in \lambda + \Lambda'$.

With the above setup, a generic coherent C&F scheme can be described as follows.

- Transmitter $\ell$ maps its message $w_\ell$ in $W$ to a coset $\lambda_\ell + \Lambda'$ in $\Lambda/\Lambda'$, and then transmits a coset leader $x_\ell$ for $\lambda_\ell + \Lambda'$.
- The receiver scales the received signal $y$ by $\alpha \in \mathbb{C}$, decodes the scaled signal $\alpha y$ to a nearest lattice point $D_\Lambda(\alpha y)$ in $\Lambda$, and then maps the coset $D_\Lambda(\alpha y) + \Lambda'$ to a message $\hat{u}$ in order to recover a linear combination.

The recovery $\hat{u}$ is said to be correct if $\hat{u} = \sum_\ell a_\ell x_\ell$ for some $a = (a_1, \ldots, a_L) \in R^L \setminus \{0\}$.

The scaling operation at the receiver plays an important role here. To understand this role, recall that (see Eq. (1))

$$\alpha y = \sum_\ell a_\ell x_\ell + n,$$

where $a_\ell \in R$ are coefficients and $n = \sum_\ell (\alpha h_\ell - a_\ell)x_\ell$ is the effective noise. In other words, the scaling operation induces a “virtual” point-to-point channel with channel input $\sum_\ell a_\ell x_\ell$ and channel noise $n$.

Note that if the channel input is decoded correctly, i.e., $D_\Lambda(\alpha y) = \sum_\ell a_\ell x_\ell$, then a message $\sum_\ell a_\ell w_\ell$ can be obtained from the coset $D_\Lambda(\alpha y) + \Lambda' = \sum_\ell a_\ell x_\ell + \Lambda'$ by using the one-to-one linear map constructed in [5]. In fact, the recovery of $\sum_\ell a_\ell w_\ell$ is correct if and only if $D_\Lambda(\alpha y) \in \sum_\ell a_\ell x_\ell + \Lambda'$ [5], which is a weaker condition than $D_\Lambda(\alpha y) = \sum_\ell a_\ell x_\ell$.

Note also that the effective noise $n$ can be “minimized” by choosing the scalar $\alpha$ and the coefficient vector $a$ cleverly. In fact, for given $(\alpha, a)$, the computation rate

$$R(\alpha, a) = \log_2 \left( \frac{\text{SNR}}{\|\alpha h - a\|^2 + |\alpha|^2} \right) \quad (3)$$

is achievable [1], where $\text{SNR} = P/N_0$, $h = (h_1, \ldots, h_L)$, and $\log_2 x = \max\{\log_2 x, 0\}$. Moreover, the optimal $(\alpha, a)$ that maximizes $R(\alpha, a)$ can be obtained by applying Lemma 1 and Theorem 4 in [1].

To summarize, in order to recover a best linear combination, the receiver first computes the optimal scalar $\alpha$, then maps the coset $D_\Lambda(\alpha y) + \Lambda'$ to a message $\hat{u}$.

### III. Blind Compute-and-Forward: General Framework

In this section, we develop a general framework for blind C&F, leading to a generic blind C&F scheme. The basic idea of our framework is simple: although the optimal scalar is nearly impossible to acquire without CSI, some “good” scalars can still be obtained by applying an error detection code. Formally, we introduce the following definitions.

**Definition 1:** A scalar $\alpha$ is said to be good, if $D_\Lambda(\alpha y) = \sum_{\ell=1}^L a_\ell x_\ell + \Lambda'$ for some $(a_1, \ldots, a_L) \in R^L \setminus \{0\}$ and is said to be bad otherwise.

Recall that the recovery of $\sum_\ell a_\ell w_\ell$ is correct if and only if $D_\Lambda(\alpha y) \in \sum_\ell a_\ell x_\ell + \Lambda'$. This justifies the above definition.

**Definition 2:** The good region of scalars, denoted $G_\alpha$, is the set of all good $\alpha$’s, i.e., $G_\alpha = \{\alpha \in \mathbb{C} : \alpha$ is good$\}$. Note that the good region depends on the channel gains. Thus, it is unknown to the receiver. Nevertheless, it is still beneficial to understand some basic properties of the good region, which will facilitate our design of blind C&F schemes.

**A. Properties of the good region $G_\alpha$**

It turns out that the good region $G_\alpha$ has a number of interesting properties, when the underlying lattice partitions $\Lambda/\Lambda'$ are asymptotically good (in the sense of [4]). Moreover, these properties still hold (or approximately hold) even for practical lattice partitions.

When the underlying lattice partitions are asymptotically good, Theorem 3 in [1] can be restated as: a scalar $\alpha$ is good if and only if the message rate $R_m$ is less than the computation rate $R(\alpha, a)$ (given in (3)) for some $a \in R^L \setminus \{0\}$. Based on this fact, we are able to show that the good region $G_\alpha$ is bounded, symmetric, and consisting of a union of disks.

**Proposition 1:** The good region $G_\alpha$ is bounded.

*Proof:* If $\alpha$ is good, then by definition

$$\log_2 \left( \frac{\text{SNR}}{\|\alpha h - a\|^2 + |\alpha|^2} \right) > R_m \quad \text{for some} \ a \in R^L \setminus \{0\}.$$ Note that

$$\log_2 \left( \frac{\text{SNR}}{|\alpha|^2} \right) \geq \log_2 \left( \frac{\text{SNR}}{\|\alpha h - a\|^2 + |\alpha|^2} \right).$$

It follows that

$$\log_2 \left( \frac{\text{SNR}}{|\alpha|^2} \right) > R_m.$$ Thus, every good $\alpha$ is bounded by $|\alpha|^2 < \text{SNR}/2R_m$. ■

**Proposition 2:** The good region $G_\alpha$ is symmetric with respect to rotations by some angle $\theta$. The angle $\theta$ is determined by $R$. 

Thus, every good $\alpha$ is bounded by $|\alpha|^2 < \text{SNR}/2R_m$. ■
Proof: It suffices to show that if $\alpha$ is good, so is $e^{i\theta}\alpha$ for some angle $\theta$. We need the following fact in abstract algebra.

Let $R$ be a discrete subring of $\mathbb{C}$ forming a principle ideal domain. Let $\mathcal{U}$ be the set of all the units in $R$. Then $\mathcal{U} = \{e^{2\pi ki/n} : k = 0, 1, \ldots, n-1\}$ for some positive integer $n$. That is, the units of $R$ are also the roots of unity.

Now choose a unit $u = e^{2\pi i/n}$. If $\alpha$ is good, then

$$\log_2 \left( \frac{\text{SNR}}{\text{SNR}\|\alpha h - a\|^2 + |\alpha|^2} \right) > R_m$$

for some $a \in R^L \setminus \{0\}$. Hence,

$$\log_2 \left( \frac{\text{SNR}}{\text{SNR}\|u\alpha h - ua\|^2 + |u\alpha|^2} \right) > R_m$$

and $u\alpha \in R^L \setminus \{0\}$.

Therefore, $u\alpha$ is good by definition.

Proposition 3: The good region $G_\delta$ consists of a union of disks. These disks are pairwise disjoint if the message rate $R_m \geq 2$.

Proof: Recall that $\alpha$ is good if and only if $R_m < R(\alpha, a)$ for some $a \in R^L \setminus \{0\}$, or equivalently,

$$\text{SNR}\|\alpha h - a\|^2 + |\alpha|^2 < \text{SNR}/2^{R_m}$$

for some $a \in R^L \setminus \{0\}$. Note that $\text{SNR}\|\alpha h - a\|^2 + |\alpha|^2$ can be viewed as the squared distance between the vector $(\alpha h_1\sqrt{\text{SNR}}, \ldots, \alpha h_L\sqrt{\text{SNR}}, \alpha)$ and the vector $(a_1\sqrt{\text{SNR}}, \ldots, a_L\sqrt{\text{SNR}}, 0)$. Hence, we have

$$\begin{align*}
\text{SNR}\|\alpha h - a\|^2 + |\alpha|^2 &= \text{SNR}(|\alpha^* h - a|^2 + |\alpha|^2) + |\alpha - \alpha^*(1 + \text{SNR}\|h\|^2),
\end{align*}$$

where $\alpha^*$ is the MMSE coefficient given by

$$\begin{align*}
\alpha^* &= \frac{\text{SNR}ah^H}{1 + \text{SNR}\|h\|^2}. \tag{4}
\end{align*}$$

Recall that $\text{SNR}\|\alpha^* h - a\|^2 + |\alpha|^2 = \text{SNR}/2^{R(\alpha^*, a)}$. Therefore, $\alpha$ is good if and only if

$$\begin{align*}
|\alpha - \alpha^*|^2 < \text{SNR} \left( \frac{1}{1 + \text{SNR}\|h\|^2} - \frac{1}{2^{R(\alpha^*, a)}} \right) \tag{5}
\end{align*}$$

for some $a \in R^L \setminus \{0\}$, or equivalently, $\alpha$ is in some disk of centre $\alpha^*$. This proves the first part. The proof of the second part is omitted due to space constraints.

Fig. 1(a) and 1(b) show some typical good regions for asymptotically-good Gaussian lattice partitions ($R = \mathbb{Z}[i]$) and Eisenstein lattice partitions ($R = \mathbb{Z}[[\omega]]$). Since $\mathbb{Z}[i]$ has four units $\{1, i, -1, -i\}$ and $\mathbb{Z}[[\omega]]$ has six units $\{\omega^k : k = 0, \ldots, 5\}$, the rotation angles in Fig. 1(a) and 1(b) are $90^\circ$ and $60^\circ$, respectively.

Fig. 2 depicts a typical good region for a naive lattice partition $\mathbb{Z}[i]^{400}/\mathbb{Z}[i]^{400}$. This lattice partition can be viewed as uncoded 9-QAM. Since this lattice partition is not asymptotically-good (in the sense of [4]), the disjoint areas are not quite disk-like. Nevertheless, the good region is still bounded and symmetric (with respect to rotations by $90^\circ$).

B. Generic blind C&F scheme

In Sec. III-A, we derived several properties of the good region. Here, we present a generic blind C&F scheme that aims to find a point in the good region, which is inspired by these properties.

First, we discuss how these properties can be used to facilitate the search of good points. Proposition 1 suggests that, to find a good point, it suffices to consider a bounded region. Proposition 2 shows that, to find a good point, it suffices to “ignore” some unnecessary areas. For instance, only the region in the first quadrant is worth investigating for Gaussian lattice partitions. Proposition 3 implies that, to find a good point, it suffices to “probe” a discrete set of points. The denser the points, the better the performance.

Next, let us assume that the discrete set of points is given. We now discuss how to “probe” a point by applying a linear detection code $\mathcal{C}$. First, the transmitters embed a linear detection code $\mathcal{C}$ into the message space $W$ so that each valid message $w_t$ (as well as any linear combinations) is a codeword in $C$. Second, the receiver performs a basic probing operation as described in Algorithm 1.

If a point $\alpha$ is good, then Algorithm 1 always declares $\alpha$ to be good, since $\hat{u}$ is a codeword in this case. On the other hand, if a point $\alpha$ is bad, then Algorithm 1 might make a mistake by declaring $\alpha$ to be good, due to an undetected error. The probability of this event can be made very small, when the code $\mathcal{C}$ has sufficiently many parity-checks.

Now we are ready to describe a generic blind C&F scheme. The input to the scheme is an ordered list containing a discrete set of points. The scheme probes the points in the list one by one until it finds a good point or until it reaches the end of...
the list. Thus, the output is either a (claimed) good point or nothing.

Note that the performance of the scheme is determined by the points in the list regardless of their order, whereas the computational cost of the scheme depends on the order of these points. In other words, two ordered lists containing exactly the same points achieve the same performance with possibly quite different computational complexity (as we will see in Sec. V).

IV. BLIND COMPUTE-AND-FORWARD: EFFICIENT ALGORITHMS

In this section, we propose two (complementary) strategies to control the computational complexity of the generic blind C&F scheme presented in Sec. III. The first strategy attempts to create some “smart” probing lists; the second strategy aims to reduce the complexity of the basic probing operation.

A. Hierarchically-organized list-building strategy

The choice of the probing list is crucial to attaining good performance with low complexity. For instance, the probing points can be made relatively sparse when the good region consists of many large disjoint areas. Based on this observation, we propose a heuristic method for creating the list.

First, we choose a well-shaped region \( \mathcal{R} \) to avoid unnecessary probing (see discussions in Sec. III-B). For Gaussian lattice partitions, we note that \( \mathcal{R} \) can be chosen heuristically as \([0, \log_{10}(\text{SNR})] \times [0, \log_{10}(\text{SNR})]\). For example, if \( \text{SNR} = 10\text{dB} \), then \( \mathcal{R} = [0, 1] \times [0, 1] \).

Then, we construct an \( m \)-level lattice-partition chain [9] \( \mathcal{L}_0/\mathcal{L}_1/\ldots/\mathcal{L}_m \) in \( \mathbb{C} \) (i.e., each \( \mathcal{L}_j \) is a one-dimensional complex lattice and \( \mathcal{L}_0 \supset \mathcal{L}_1 \supset \cdots \supset \mathcal{L}_m \)). Note that the lattice-partition chain, together with the region \( \mathcal{R} \), induces \( m+1 \) probing grids \( \{\mathcal{L}_j \cap \mathcal{R}\} \) satisfying \( \{\mathcal{L}_m \cap \mathcal{R}\} \subset \cdots \subset \{\mathcal{L}_0 \cap \mathcal{R}\} \) (see Fig. 3 for a concrete example). For Gaussian lattice partitions, we heuristically set \( \mathcal{L}_j = \frac{1}{10} \log_{10}(\text{SNR})(1+i)^j\mathbb{Z}[i] \), where \( j = 0, \ldots, 8 \).

With these grids, a list-building algorithm is described in Algorithm 2. The order in the list is designed such that the points in the sparse grids will appear before the points in the dense grids.

B. Early rejection strategy

In the basic probing operation, the receiver first decodes a lattice point \( D_\Lambda(\alpha y) \) and then performs error detection for \( \hat{u} \).

Here we propose to conduct an early detection well before complete decoding of \( \alpha y \). This can be achieved for certain lattice-decoder structures, e.g., a lattice decoder based on the Viterbi algorithm.

Recall that the Viterbi algorithm is an instance of maximum-likelihood decoding for convolutional codes, and it generates a list of candidates for each initial message segment along the trellis search, where the size of the list is equal to the number of states of the Viterbi algorithm. Note that, for certain initial message segment, if all current candidates fail to pass the parity-checks involved so far, then there is no need to continue the trellis search and the algorithm aborts. This suggests an early rejection strategy as summarized in Algorithm 3.

We note that there is an efficient implementation of Algorithm 3 in which each parity-check equation is verified at most \( M \) times, where \( M \) is the number of states of the Viterbi algorithm. Thus, the cost of verifying parity-check equations is much less than the cost of the trellis search. In other words, the computational cost of Algorithm 3 is dominated by the trellis search. If Algorithm 3 stops at some initial message segment \( (\hat{u}_1, \ldots, \hat{u}_j) \), then the cost is roughly \( j/k \) of the cost of a complete trellis search, where \( k \) is the length of the messages.

V. SIMULATION RESULTS

In this section, we illustrate the feasibility of blind C&F through simulations. The setup of our simulation is as follows. The lattice partition \( \Lambda/\Lambda' \) is constructed from a terminated convolutional code over \( \mathbb{Z}[i]/(3) \) following Example 7 in [5] with the parameters \( \mu = 400, \nu = 1 \). For illustrative purposes, the linear detection code \( C \) is set to a simple \( 20 \times 20 \) product code.
code consisting of a $[20,19]$ single parity-check code with itself over $\mathbb{Z}[i]/(3)$. Clearly, the rate of $C$ is $361/400 \approx 0.9$. The region $R$ is set to $[0, \log_{10}(SNR)] \times [0, \log_{10}(SNR)]$, and the lattice-partition chain is $L_j = \frac{1}{2} \log_{10}(SNR)(1 + i)^j \mathbb{Z}[i] (j = 0, \ldots , 8)$ as suggested in Sec. IV-A.

We consider a two-transmitter, single receiver configuration, which can be viewed as a building block of a more complicated and realistic network scenario. Communication occurs in rounds. In each round, the channel gains are assumed to follow independent Rayleigh fading. A round is said to be successful if the receiver correctly recovers a linear combination. The throughput is defined as the fraction of successful rounds in the simulation.

We have evaluated four blind C&F schemes through simulation by carrying out 1000 rounds. The first scheme is a baseline scheme where the probing list contains all the points in $L_0 \cap R$ ordered so that their $L_1$-norms are non-decreasing. The second and third schemes use the strategies outlined in Sec. IV-A and IV-B, respectively, while the fourth scheme uses both strategies simultaneously.

Note that the throughputs of each scheme is the same, as each scheme contains the same set of points. Table I compares the throughput of these blind schemes with the throughput of coherent C&F under various SNRs. It is observed that these blind C&F schemes are always able to approach the throughput of coherent C&F.

### TABLE I

**THROUGHPUT OF COHERENT AND BLIND C&F SCHEMES.**

<table>
<thead>
<tr>
<th>SNR</th>
<th>12dB</th>
<th>14dB</th>
<th>16dB</th>
<th>18dB</th>
<th>20dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coherent</td>
<td>57.0%</td>
<td>73.8%</td>
<td>84.8%</td>
<td>92.1%</td>
<td>97.3%</td>
</tr>
<tr>
<td>Blind</td>
<td>56.8%</td>
<td>73.5%</td>
<td>82.9%</td>
<td>90.2%</td>
<td>95.1%</td>
</tr>
</tbody>
</table>

Next we examine the complexity of these blind C&F schemes under various SNRs. Recall that the complexity of our blind C&F schemes is dominated by the trellis search. As such, the complexity is normalized relative to one complete trellis search. Table II compares the complexity of these blind schemes under various SNRs. It is observed that our proposed strategies significantly reduce the complexity, especially when the throughput is high.

### TABLE II

**COMPLEXITY OF FOUR BLIND C&F SCHEMES.**

<table>
<thead>
<tr>
<th>SNR</th>
<th>12dB</th>
<th>14dB</th>
<th>16dB</th>
<th>18dB</th>
<th>20dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>152.12</td>
<td>119.00</td>
<td>99.03</td>
<td>74.78</td>
<td>59.71</td>
</tr>
<tr>
<td>hierarchical grid</td>
<td>120.55</td>
<td>78.50</td>
<td>54.80</td>
<td>35.86</td>
<td>22.01</td>
</tr>
<tr>
<td>early rejection</td>
<td>36.55</td>
<td>28.44</td>
<td>23.04</td>
<td>18.74</td>
<td>15.50</td>
</tr>
<tr>
<td>combined</td>
<td>27.14</td>
<td>17.11</td>
<td>12.04</td>
<td>8.16</td>
<td>5.30</td>
</tr>
</tbody>
</table>

To summarize, our simulation results suggest that a blind C&F scheme that admits Viterbi decoding is able to approach the throughput of coherent C&F with around 10 times complexity in the high-throughput region.

### VI. CONCLUSION

In this paper, the problem of designing blind C&F schemes has been considered. A framework based on error-detection has been proposed, which eliminates the need for CSI in C&F. In particular, a generic blind C&F scheme has been developed, and several strategies have been suggested to make it computationally efficient. The effectiveness of our approach has been illustrated through simulations. We believe that there is still much work to be done in this direction, particularly in investigating the performance-complexity tradeoff of the blind C&F schemes, as well as devising efficient probing strategies for alternative coding structures.

### ACKNOWLEDGMENT

The authors would like to thank Bobak Nazer and Michael Gastpar for suggesting the blind compute-and-forward problem.

### REFERENCES