Huffman-coded Sphere Shaping and Distribution Matching Algorithms via Lookup Tables

Fehenberger, Tobias; Millar, David S.; Koike-Akino, Toshiaki; Kojima, Keisuke; Parsons, Kieran; Griesser, Helmut

TR2020-051   April 25, 2020

Abstract

In this paper, we study amplitude shaping schemes for the probabilistic amplitude shaping (PAS) framework as well as algorithms for constant-composition distribution matching (CCDM). Huffman-coded sphere shaping (HCSS) is discussed in detail, which internally uses Huffman coding to determine the composition to be used and relies on conventional CCDM algorithms for mapping and demapping. Numerical simulations show that HCSS closes the performance gap between distribution matching schemes and sphere shaping techniques such as enumerative sphere shaping (ESS). HCSS is based on an architecture that is different from the trellis-based setup of ESS and that allows to tailor the used HCSS compositions to the transmission channel as well as complexity constraints. We further discuss in detail multiset ranking (MR) and subset ranking (SR) as alternatives to arithmetic-coding (AC) CCDM. The advantage of MR over AC is that it requires less sequential operations for mapping. SR operates on binary alphabets only, which can introduce some additional rate loss as a nonbinary-tobinary transformation can be required. However, the binomial coefficients required for SR can be precomputed and stored in a lookup table. We perform an analysis of rate loss and decoding performance of the proposed techniques and compare them to other prominent amplitude shaping schemes. For medium to long block lengths, MR-HCSS and SR-HCSS are shown to have similar performance to ESS, while using a very different architecture that allows the design to be tailored to the channel. SR-HCSS and uniform 64QAM are compared in additive white Gaussian noise (AWGN) simulations and shaping gains of 0.5 dB and 1 dB are demonstrated with 1 kbit and 100 kbit LUT size, respectively.

IEEE Journal of Lightwave Technology

Huffman-coded Sphere Shaping and Distribution Matching Algorithms via Lookup Tables

Tobias Fehenberger, *Member, IEEE*, David S. Millar, *Member, IEEE*, Toshiaki Koike-Akino, *Senior Member, IEEE*, Keisuke Kojima, *Senior Member, IEEE*, Kieran Parsons, *Senior Member, IEEE*, and Helmut Griesser, *Member, IEEE*

Abstract—In this paper, we study amplitude shaping schemes for the probabilistic amplitude shaping (PAS) framework as well as algorithms for constant-composition distribution matching (CCDM). Huffman-coded sphere shaping (HCSS) is discussed in detail, which internally uses Huffman coding to determine the composition to be used and relies on conventional CCDM algorithms for mapping and demapping. Numerical simulations show that HCSS closes the performance gap between distribution matching schemes and sphere shaping techniques such as enumerative sphere shaping (ESS). HCSS is based on an architecture that is different from the trellis-based setup of ESS and that allows to tailor the used HCSS compositions to the transmission channel as well as complexity constraints. We further discuss in detail multiset ranking (MR) and subset ranking (SR) as alternatives to arithmetic-coding (AC) CCDM. The advantage of MR over AC is that it requires less sequential operations for mapping. SR operates on binary alphabets only, which can introduce some additional rate loss as a nonbinary-to-binary transformation can be required. However, the binomial coefficients required for SR can be precomputed and stored in a lookup table. We perform an analysis of rate loss and decoding performance of the proposed techniques and compare them to other prominent amplitude shaping schemes. For medium to long block lengths, MR-HCSS and SR-HCSS are shown to have similar performance to ESS, while using a very different architecture that allows the design to be tailored to the channel. SR-HCSS and uniform 64QAM are compared in additive white Gaussian noise (AWGN) simulations and shaping gains of 0.5 dB and 1 dB are demonstrated with 1 kbit and 100 kbit LUT size, respectively.

Index Terms—Probabilistic Amplitude Shaping, Sphere Shaping, Huffman Coding, Distribution Matching

I. INTRODUCTION

A conventional modulation format whose signal points are equidistant (i.e., lie on a square grid) and equiprobable (i.e., occur with the same probability) can be tailored to the transmission channel for improved performance, e.g., less required signal-to-noise ratio (SNR) for a fixed throughput. This optimization is known as constellation shaping, and in general two different approaches exist. Geometric shaping, on the one hand, optimizes the position of the constellation points in the complex plane [1]–[4]. This approach offers large shaping gains for symbol-wise forward error correction (FEC) decoding. As geometrically shaped constellations often

T. Fehenberger was with Mitsubishi Electric Research Laboratories. He is now with ADVA, Munich, Germany. E-mail: tfehenberger@adva.com.

D. S. Millar, T. Koike-Akino, K. Kojima and K. Parsons are with Mitsubishi Electric Research Laboratories. E-mails: millar@merl.com; koike@merl.com; kojima@merl.com; parsons@merl.com.

H. Griesser is with ADVA, Munich, Germany. E-mail: hgriesser@adva.com.

do not have a Gray mapping, it can be challenging to achieve significant shaping gains in bit-interleaved coded modulation (BICM) systems with binary FEC [5]. Probabilistic shaping (PS), on the other hand, modifies the probability of the constellation symbols which remain on a square grid. Classic PS schemes in the literature include many-to-one mappings [6], trellis shaping [7], and shell mapping [8].

To the best of our knowledge, coded modulation system with PS have been investigated in fiber optics as early as 2012 [9], [10] by using trellis shaping and shell mapping, respectively. More work on PS has been published afterwards, such as [11], [12]. PS attracted a great deal of attention within the fiber-optic community in 2014 with the proposal of probabilistic amplitude shaping (PAS) [13]. The first demonstrations [14]–[16] were presented shortly after. Ever since then, PAS has been investigated in wireless communications [17], [18] and particularly in the fiber-optic literature where various aspects are covered, such as back-to-back sensitivity analysis [16], field trials [19], interplay with digital signal processing [20], and investigations into fiber nonlinearities, both for infinite-length [21]–[23] and finite-length cases [**?**].

The success of PAS as coded modulation framework for PS can be attributed to several aspects. As most shaping schemes, it closes the gap to the Shannon limit and offers gain of up to 1.53 dB. Furthermore, rate adaptivity is enabled, i.e, the throughput can be varied with fixed modulation order and FEC overhead. Finally, its reverse concatenation principle allows a low-complexity integration of PAS into existing coded modulation system with binary FEC. The main complexity of PAS lies in the block-wise amplitude shaper that transform a uniform binary input to a sequence of shaped amplitudes at the transmitter and undoes this operation at the receiver.

In the initial PAS proposal, a constant-composition distribution matcher (CCDM) [24] is used as amplitude shaper. All shaped output sequences generated by a CCDM have the same empirical distribution, which means that all transmit sequences are permutations of each other. While a CCDM asymptotically achieves ideal performance, its finite-length behavior is suboptimal. The arithmetic coding (AC) algorithm used for CCDM is an inherently sequential method [24]. The combination of long blocks required for low loss and sequential processing has lead to a great deal of research into advanced amplitude shaper schemes that improve upon CCDM. Examples of fixed-length amplitude shapers include multiset-partition distribution matching (MPDM) [25], multi-composition DM [26], prefix-free code DM with framing [27],

enumerative sphere shaping (ESS) [17], shell mapping (SM) [8], [28], and hierarchical DM [29].

In the first part of this manuscript (Sec. II), we study an amplitude shaping technique that uses Huffman coding to address all compositions inside an *n*-dimensional sphere and conventional CCDM algorithms for the mapping or demapping according to the selected composition. This Huffman-coded sphere shaping (HCSS) approach, which has been introduced in [30], is a generalization of MPDM and can closely approach the performance of sphere shaping schemes, yet with a different architecture than ESS and SM. An energy comparison of HCSS with other prominent state-of-the-art shaping schemes highlights their conceptual difference. We further present numerical simulations results for rate loss and decoding performance. The second part of this manuscript (Sec. III) covers subset and multiset ranking, which are alternatives to the arithmetic coding (AC) method used for CCDM mapping and demapping. Multiset ranking (MR), also briefly mentioned in [30], is explained in detail for the first time and algorithms are presented. Subset ranking (SR) has been proposed before in [31] as an alternative implementation for binary-alphabet AC-CCDM that has a greatly reduced number of sequential operations. In this work, an efficient implementation of SR based on a lookup table (LUT) is proposed. We demonstrate that a LUT of approximately 1 kbit size is sufficient to achieve significant shaping gains of 0.5 dB for 64-ary quadrature amplitude modulation (QAM). Increasing the LUT size to 100 kbit can yield shaping gains of up to 1 dB.

II. HUFFMAN CODED SPHERE SHAPING (HCSS)

In the following, we review fundamentals of PAS and amplitude shapers. We then explain HCSS, and compare it to other shaping algorithms with respect to energy aspects as well as occupation of the shells in an *n*-dimensional sphere.

A. Preliminaries of Amplitude Shapers

All considered fixed-to-fixed-length amplitude shapers are designed to carry out a seemingly simple mapping operation from a uniformly distributed bit sequence of length *k* to a shaped sequence of length *n*. This mapping function must be invertible, which makes the considered problem effectively an indexing task. Mathematically speaking, if all length-*n* sequences are in the target set of the mapping function, we have an injective and non-surjective mapping, meaning that each bit input must be mapped onto one shaped sequence, but not every possible sequence of length *n* is used (as this would just give a uniform distribution on average). Under the assumption typically made in PAS that all errors are corrected by the FEC, the inverse operation, demapping, is then a bijective mapping from shaped sequences to binary data.

Following [32], we can distinguish between two main classes of amplitude shapers, depending on the properties of the output sequences. We speak of the direct approach if a shaping code is explicitly designed, which corresponds in the nomenclature commonly used in the context of PAS to a distribution matcher, either for a single composition (CCDM) or over various compositions (e.g., MPDM). In this direct

approach, the transmission rate follows only indirectly from the probability mass function (PMF) and the employed shaper. For the indirect approach, on the other hand, a shaping code is not defined explicitly but obtained by bounding the used signal space by a sphere, where the maximum energy of the sphere determines the transmission rate. Prominent examples of indirect schemes are ESS and SM. More details on this can be found in, [33, Sec. IV]. The proposed HCSS can be considered a hybrid between these two classifications. As will be explained in Sec. II-C, a specific HCSS code is designed, but the design premise is to approach a signal space that is bounded in a similar way as for sphere shaping schemes.

B. Rate, Rate Loss, and Related Quantities

The rate of an amplitude shaper is k/n bits per amplitude symbol. Any finite-length shaping scheme exhibits a rate loss defined as

$$R_{\text{loss}} = \mathbb{H}(A) - k/n, \tag{1}$$

where $\mathbb{H}(A)$ is the entropy of the amplitudes A that take on values from the alphabet $\mathcal{A} = a_1, \ldots, a_m$ according to the PMF P_A. The rate loss directly reduces the throughput that can be achieved with PAS, see [25, Appendix]. Hence, for a linear channel such as the additive white Gaussian noise (AWGN) channel, an efficient shaping method seeks to minimize the rate loss for a fixed block length *n*, which typically should also be as small as possible for implementation reasons.[1] For a fixed *n*, the number of addressable input bits *k* is defined as

$$k = \lfloor \log_2 n_{\text{seq}} \rfloor, \tag{2}$$

where $\lfloor \cdot \rfloor$ denotes rounding down to the nearest integer and n_{seq} is the total number of all possible sequences that are output by the respective shaper. For CCDM, all these sequences have the same composition, i.e., they are permutations of each other. A composition $\underline{C}$ is the ordered set of occurrences $[c_1, \ldots, c_m]$ of the amplitudes $[a_1, \ldots, a_m]$ in a sequence of length $n = \sum_{i=1}^m c_i$. The individual elements of $\underline{C}$ are formally defined as

$$c_i = |\{ j : x_j = a_i \}|, \quad j \in 1, \ldots, n, \quad i \in 1, \ldots, m, \tag{3}$$

and we have $\sum_{i=1}^m c_i = n$. The number of permutations of a composition $\underline{C}$ is determined by the multinomial coefficient (MC), which is defined as

$$\text{MC}(\underline{C}) = \frac{n!}{\prod_{i=1}^m (c_i!)}. \tag{4}$$

For MPDM, the permutation count of all permissible compositions must be added. For sphere shaping schemes, the number of used shells is increased until the required n_{seq} and thus the desired *k* is achieved. Since the shells in the design process are preferably sorted by energy, using more shells increases the average energy of the scheme. Average energy is defined as

$$\mathbb{E} = \sum_{a \in \mathcal{A}} P_A(a) \cdot a^2. \tag{5}$$

---

[1]Notable exceptions are schemes that offer advantages in implementation at the cost of some extra rate loss, such as the parallel-amplitude transformation proposed in [31].
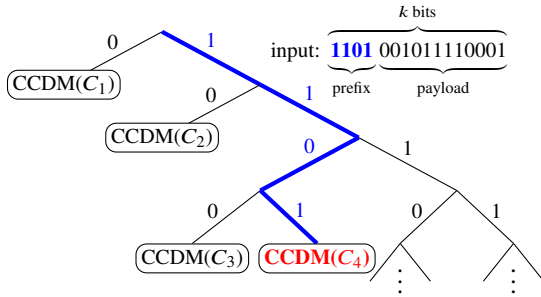
Fig. 1. Tree structure for Huffman coded sphere shaping (HCSS). A prefix (blue) of the k-bit input defines which composition (red) is to be used. The remaining payload is mapped via CCDM methods, such as arithmetic coding or the multiset and subset ranking methods discussed in Sec. III.

Fig. 2. Illustration of the shell occupation in an n-dimensional sphere for various amplitude shapers. Figure extended from [33, Fig. 6].

For non-constant-composition schemes, the amplitude PMF P_A can be computed from the weighted average of all utilized sequences.

C. Principle of HCSS

HCSS is most easily understood when viewed as an extension of CCDM and MPDM. As noted above, CCDM addresses sequences from a single composition and thus, every CCDM output block has an empirical distribution that is the target distribution.[2] MPDM is an extension of CCDM to multiple compositions with two constraints. Firstly, for pairwise MPDM, only composition pairs are utilized, meaning that for each composition there exists a complement with the same number of permutations such that the target composition, and thus the target PMF, is achieved on average over every pair. Secondly, the number of permutations of each pair must be a power of two to allow construction of a Huffman tree.

HCSS is then a further generalization of MPDM where the constraint of having composition pairs is lifted. Instead of just considering pairs, all compositions are sorted by energy and used in order of ascending energy until the total number of permutations allows to address k bits, i.e, until (2) is fulfilled. For binary input of length k, the selection of composition to be used is based on Huffman coding. Once the composition has been found, the mapping problem is that of a conventional CCDM since we have a constant composition in each node of the binary Huffman tree. In order to construct a Huffman tree, the number of permutations of each composition must be a power of two, which is the same requirement as for MPDM (and trivially also CCDM). As we will see in Sec. IV, introducing structure into the n-dimensional sphere results in small additional rate loss, but allows to use a simple binary Huffman tree and CCDM methods for sphere shaping.

The mapping of a k-bit input to a HCSS codeword is illustrated in Fig. 1. First, a prefix (bold blue) determines which composition to use, in this case C_4. The mapping of the remainder of the input bit sequence to the shaped sequence with composition C_4 can be done with any CCDM

---

[2]Note that this is only the case for the one-sided amplitude symbols because the PAS sign bits are not perfectly uniformly distributed, which means that the generated ASK or QAM sequence is not constant composition.
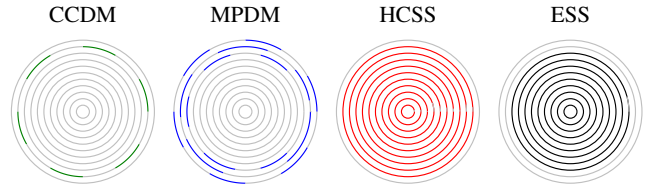
algorithm, such as arithmetic coding or the methods introduced in Sec. III. The binary Huffman tree is constructed by sorting all compositions that fulfill the energy constraint in descending order by their number of permutations, rounding them down to the closest power of two, and assigning a prefix to each of them such that the lengths of the prefix and of the payload are equal to the overall number of addressable input bits k.

D. Energy Considerations

It is well-known that it is beneficial for the AWGN channel to select all possible signal points from within an n-dimensional sphere, or equivalently have the signal space bounded by that n-sphere [17], [34]. CCDM does not have this property as only sequences of a constant composition are used, which corresponds to using only a single fixed-energy shell of this n-sphere. Furthermore, as each shell can consist of different compositions, CCDM does in general not fully utilize the shell. This is illustrated in Fig. 2, where each gray circle indicates a shell in the n-sphere, and the center of the rings corresponds to the zero-energy point. For CCDM, one shell is partly occupied. MPDM combines compositions with their complementary counterpart to achieve the desired distribution on average and thus uses several shells. In contrast to CCDM, each utilized shell can be populated with one or more compositions, which is indicated in Fig. 2 by the blue MPDM sections occupying a larger portion of each gray shell. Both HCSS and ESS can use all shells up to a certain maximum energy, and the total number of used output sequences then determines the transmission rate. The difference is that HCSS still operates on compositions, whereas the concept of compositions does not directly exist for ESS. Due to the Huffman tree employed in HCSS, the number of output sequences of each composition that can be mapped must be a power of two, which means potentially rounding down the number of permutations. This is indicated by the open sections in each HCSS shell in Fig. 2. ESS, in contrast, only has the constraint that the total number of shaped output sequences must be a power of two (a common requirements of all amplitude shaping schemes), which is ideally achieved by not fully using the outermost, highest-energy shell. Nonetheless, the design principle of HCSS allows to make adjustments to the shaper design that are not feasible with conventional sphere shaping techniques such as ESS or SM, as we will show next.

E. Design Modifications to HCSS

The composition-based architecture of HCSS gives the freedom to select which compositions to use and which ones
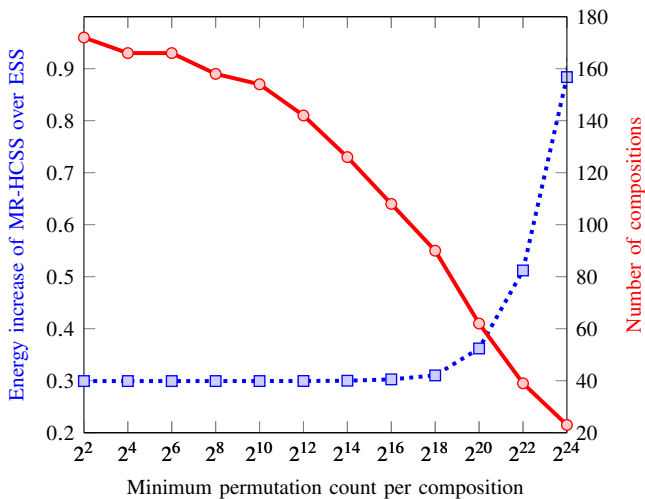
Fig. 3. Average energy increase of MR-HCSS compared to ESS (left, dotted blue) and Huffman tree depth (right, solid red) versus minimum size of composition in number of permutations for HCSS.

to omit. For instance, compositions can be excluded that might have adverse effects for the nonlinear fiber channel [35]. Furthermore, the size of the HCSS Huffman tree can be reduced by omitting composition whose contribution to the total number of permutations is small, which can give a reduction in computational complexity. This comes, however, at the expense of an increase in average energy, and thus a small additional rate loss, if less low-energy shells are used. In Fig. 3, the energy increase of MR-HCSS over ESS and the corresponding Huffman tree depth (right axis) is shown for with $n = 20$ and rate 1.5 bit/1D-sym. The energy increase is given in linear scale and is essentially a penalty, stating the average energy (as defined in (5)) of ESS minus that of MR-HCSS. This means that a higher average energy value in Fig. 3 generally means worse performance. For computing average energy as defined in (5), the amplitude levels are assumed to be $\mathcal{A} = \{1, 3, 5, 7\}$. The ESS reference has $\mathbb{E} = 8.416$, which is obtained by evaluating the average PMF that is achieved by ESS at the considered parameters. The horizontal axis shows the minimum number of permutations of each composition, i.e., compositions whose permutation count (given by the multinomial coefficient, see (4)) is smaller than that number are not used. We observe from Fig. 3 that compositions with permutation count up to 2^{18} can be excluded from HCSS without a noticeable increase in energy, yet the total number of compositions is significantly reduced from 172 compositions to 90. At this point, the additional rate loss compared to an unmodified HCSS is less than 0.001 bit/1D-sym. When increasing the minimum permutation count beyond 2^{18}, a penalty arises, and the rate loss increases from 0.098 bit/1D-sym to 0.14 bit/1D-sym. Directly related to the total number of compositions is the maximum Huffman tree depth, which, over the range shown in Fig. 3, decreases linearly from 26 bits to 11 bits.

III. CCDM ALGORITHMS: SUBSET RANKING AND MULTISET RANKING

In this section, we review the subset ranking (SR) and multiset ranking (MR) methods, which are alternatives to arithmetic coding for CCDM. We start with SR, which only supports binary output alphabets, and then cover the more general MR that does not have this restriction. Finally, we discuss the implementation of SR and MR with lookup tables.

A. Subset Ranking

Subset ranking is a CCDM algorithm that requires a binary output alphabet. In order to support nonbinary alphabets (i.e., more than two shaped amplitudes, which corresponds to a QAM order higher than 16), a nonbinary-to-binary transformation must be performed, for instance by restricting the PMF to product distributions [18], [36] or by using the parallel-amplitude (PA) framework [31, Sec. III], which is used throughout this work for SR. As we will see in Sec. IV, PA introduces some additional rate loss, yet allows higher parallelization than other schemes, see [31, Sec. V].

The idea behind SR is that any binary sequence has an equivalent representation as constant-order subset comprising the indices where one of the two binary symbols occurs in the initial sequence. The cardinality of this subset corresponds to the weight w, which is defined as the number of occurrences of one of the two binary symbols. All possible subsets of a certain weight (cardinality) can be sorted according to a specific ordering, such as lexicographical. Each subset in this sorted list of subset can then be ranked, with the rank specifying how many subsets appear in that ordered list prior to the considered subset. With this approach, an invertible mapping function is established between the rank (corresponding to the binary input sequence of length k) and the constant-order subset (which is an equivalent representation of the shaped binary output sequence). The CCDM mapping operation for a given ranking and the subset being the output is thus unranking, and demapping is called ranking for which the rank for a certain shaped sequence is sought. We will show a simple example illustrating SR in the following. For more details, we refer to [31, Sec. IV].

Example 1 (Subset Ranking). *A binary DM of length $n = 10$ and the binary distribution $P(\alpha) = 0.6$, $P(\beta) = 0.4$ can map $k = \left\lfloor \log_2 \binom{10}{4} \right\rfloor = 7$ bits to output sequences, each of which contains exactly six times α and four β-symbols. Now let us represent all possible output sequences by their respective positions of the β-symbol, where we start counting with one from the left. For instance, $[\beta\alpha\alpha\beta\alpha\alpha\alpha\alpha\beta\beta]$ is equivalent to $[1, 4, 9, 10]$ as β occurs at these indices. Next, we list all possible outputs sorted according to some criterion. In this example, we choose lexicographical order, which means that we start with $[1, 2, 3, 4]$, the next element would be $[1, 2, 3, 5]$ and so forth. Suppose the binary word to be mapped is $= [1110100]$, which is 116 in the decimal system, and this is referred to as rank. The task of SR mapping is now to choose the sequence in the sorted list that has rank 116, i.e., we pick the element with 116 predecessors, which can be computed*

(or looked-up) to be $[2,4,7,9]$. *This set states the index of the* β, *giving the final output sequence is* $[\alpha\beta\alpha\beta\alpha\alpha\beta\alpha\beta\alpha]$. *At the receiver, the inverse operation is carried out by computing the rank based on the received shaped sequence.*

B. Multiset Ranking

The previously introduced SR requires a binary alphabet. In the following, we describe multiset ranking (MR) that operates directly on nonbinary alphabets. By lifting the binary-alphabet constraint that can cause a small penalty, MR can be regarded as an implementation alternative to AC-CCDM without extra rate loss.

For the MR method, we distinguish between the target rank, which is simply the input bit sequence to be mapped, and the relative rank, which is defined as the total number of sequences with lower rank, for a given amplitude with a specified lexicographical ordering, occurring in a certain position of the shaped output sequence. Obviously, the relative rank for the smallest amplitude is always zero since the number of preceding sequences for that element by definition must be zero. For the remaining amplitudes, the relative ranks are computed as cumulative sums over the multinomial coefficients of the composition that is updated based on the considered amplitude and output position. For instance, the relative rank of the third amplitude is the sum over the previously computed relative ranks plus the MC of the target composition with the second element reduced by one. The reason for this updating rule for the third amplitude is that we want to determine the number of permutations when the considered position is occupied with the first or second amplitude. When all relative ranks are computed, the amplitude that has the largest relative rank not exceeding the target rank is chosen. Afterwards, the target rank and the remaining composition are updated, and the above steps are carried out until the shaped sequence of length n is constructed. The algorithm for MR mapping (unranking) is presented as pseudo code in Algorithm 1. The inverse operation, MR demapping, is described in Algorithm 2. For easier notation, we assume that the amplitudes are represented as natural numbers from 1 to m, i.e., $\mathcal{A} = \{1, 2, \ldots, m\}$, and denoted the shaped amplitude sequence as $\underline{s} = [s_1, \ldots, s_n]$.

In contrast to arithmetic coding, which is serial in k for mapping and n for demapping, MR is serial in n for mapping and demapping, and thus requires fewer serial operations without incurring additional rate loss. We note that the inner for-loop in Algorithm 1 is considered as fully parallelizable as the computationally demanding step of calculating the MCs can be executed in parallel.

C. Implementation with Lookup Tables

As we have seen above, the SR and MR algorithms rely on computing binomial and multinomial coefficients, respectively. In the following, we outline the storage requirements when storing these coefficients in a LUT.

The main advantage of SR over AC are the highly parallel algorithms for ranking and unranking whose computational complexity lies mostly in calculating binomial coefficients.

Algorithm 1 Multiset Ranking: Mapping

Require: $\underline{C}, r_{\text{target}}$ ▷ composition, target rank
1: **function** MR_MAPPING($\underline{C}, r_{\text{target}}$)
2:     $\underline{C}_{\text{tmp}} = \underline{C}$
3:     **for** i from 1 to n **do**
4:         $\underline{r}_{\text{rel}} \leftarrow [0, \ldots, 0]$ ▷ m zeros
5:         **for** j from 2 to m **do**   ▷ all but the first position
6:             $\underline{t} \leftarrow [0, \ldots, 0]$ ▷ temp. variable with m zeros
7:             $t_{j-1} \leftarrow 1$ ▷ 1 at index $j-1$
8:             $r_{\text{rel},j} \leftarrow r_{\text{rel},j-1} + \text{MC}\left(\underline{C}_{\text{tmp}} - \underline{t}\right)$
9:         **end for**
10:         find max. index l where $r_{\text{target}} < \underline{r}_{\text{rel}}$
11:         $s_i \leftarrow a_l$ ▷ populate output sequence
12:         $C_{\text{tmp},l} \leftarrow C_{\text{tmp},l} - 1$ ▷ update remaining comp.
13:         $r_{\text{target}} \leftarrow r_{\text{target}} - r_{\text{rel},l}$ ▷ update target rank
14:     **end for**
15:     **return** $\underline{s}$ ▷ return shaped sequence
16: **end function**

Algorithm 2 Multiset Ranking: Demapping

Require: $\underline{C}, \underline{s}$ ▷ composition, shaped sequence
1: **function** MR_DEMAPPING($\underline{C}, \underline{s}$)
2:     $\underline{C}_{\text{tmp}} = \underline{C}$
3:     **for** i from 1 to n **do**
4:         $\underline{r}_{\text{rel}} \leftarrow [0, \ldots, 0]$ ▷ m zeros
5:         **for** j from 2 to m **do**   ▷ all but the first position
6:             $\underline{t} \leftarrow [0, \ldots, 0]$ ▷ temp. variable with m zeros
7:             $t_{j-1} \leftarrow 1$ ▷ 1 at index $j-1$
8:             $r_{\text{rel},j} \leftarrow r_{\text{rel},j-1} + \text{MC}\left(\underline{C}_{\text{tmp}} - \underline{t}\right)$
9:         **end for**
10:         find index l of s_i ▷ index in amplitude list
11:         $C_{\text{tmp},l} \leftarrow C_{\text{tmp},l} - 1$ ▷ update remaining comp.
12:         $r_{\text{target}} \leftarrow r_{\text{target}} + r_{\text{rel},l}$ ▷ update target rank
13:     **end for**
14:     **return** r_{target} ▷ return rank
15: **end function**

For reasonably short block lengths, all required binomial coefficients can be precomputed and stored in a LUT. For a given n, the required number of binomial coefficients to be computed is

$$\left\lfloor \frac{n}{2} \right\rfloor - 1 \tag{6}$$

as they have a symmetry around $n/2$ (i.e., $\binom{n}{w} \equiv \binom{n}{n-w}$ for $n > w$) and since $\binom{n}{1} = n$. The number of required LUT entries for all DMs with length of at most n is thus

$$\sum_{i=4}^{n} \left(\left\lfloor \frac{i}{2} \right\rfloor - 1 \right), \tag{7}$$

where the sum starts at 4 as trivial cases can be omitted which result in a binomial coefficient equal to 1 or n. The size of each LUT entry is

$$\left\lceil \log_2 \binom{n}{w} \right\rceil \tag{8}$$

bits, with the maximum size occurring for $w = \lfloor n/2 \rfloor$. With (6)–(8) we get an overall LUT size of

$$\sum_{i=4}^{n} \sum_{w=2}^{\lfloor \frac{i}{2} \rfloor} \left\lceil \log_2 \binom{i}{w} \right\rceil \qquad (9)$$

bits. As an example, a LUT for all DMs up to length $n = 50$ has 14.3 kbit size, with the maximum LUT entry requiring 47 bit. A more detailed study of LUT sizes is given in Sec. IV-C.

For MR of length-n shaping with m amplitudes, the multinomial coefficients of all compositions up to length $n-1$ must be precomputed. The storage requirements can be reduced by taking into account that MCs do not change when the order of the composition elements are permuted, see (4). The overall size of the LUT for MR of length n is thus

$$\sum_{i=1}^{n-1} \left\lceil \log_2 \left(\text{MC}\left(C_{\text{all}}^i \right) \right) \right\rceil, \qquad (10)$$

where C_{all}^i denotes the set of all compositions with m elements and block length i.

The required LUT sizes as a function of block length n is shown in Fig. 4 for SR and MR. Note that the same LUT can be used for matching at the transmitter and inverse matching at the receiver because the binomial coefficients that are to be computed do not change. We observe that for MR with $m = 4$ amplitudes, the LUT size already exceed 1 Mbit for slightly above $n = 50$, which is due to the large number of different MCs that must be stored. For SR, in contrast, only 100 kbit of storage are required for $n = 100$, and with 1 kbit of size, a block length of $n = 20$ is feasible. We note that the entries of the LUT cannot be used directly for CCDM mapping or demapping as they need to be further processed, i.e., added and compared to other numbers such as the rank, see for example Algorithm 1. The computationally most intense task of computing those coefficients is however greatly alleviated. A numerical analysis of required LUT size for certain shaping gains is presented in Sec. IV.

IV. NUMERICAL ANALYSIS

In the following, we numerically compare the rate loss, the decoding performance using low-density parity-check (LDPC) codes and the LUT storage requirements for different amplitude shaping methods.

A. Rate Loss

Figure 5 shows the rate loss in bits per 1D amplitude symbol (bit/1D-sym) versus n. The amplitude distribution for CCDM and MPDM is $P_A = [0.4, 0.3, 0.2, 0.1]$, and the sphere shaping methods (both HCSS variants and ESS) are set to operate for each n at the rate that MPDM achieves. ESS has the lowest rate loss of all schemes because it uses the signal space in the most energy-efficient way. The HCSS methods have a slightly larger rate loss, and we observe that the gap to ESS reduces when increasing n. The reason for this behavior is that at a fixed small n, the reduction in the number of input bits k of HCSS compared to ESS translates into a larger
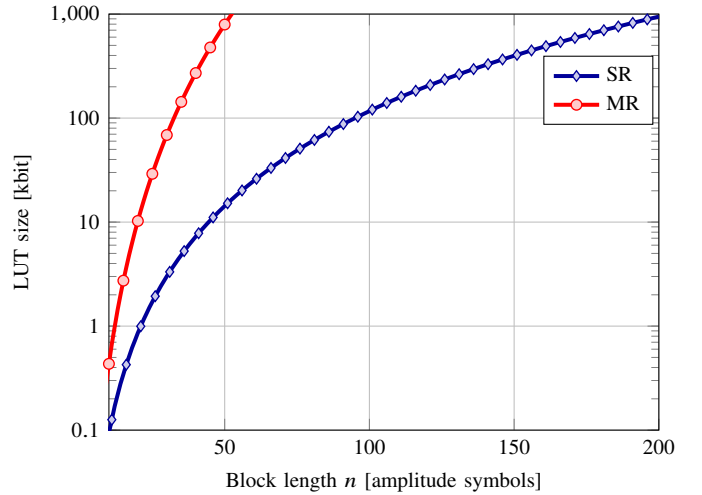

Fig. 4. LUT size in kbit versus block length n. For MR, $m = 4$ amplitudes are assumed.

absolute rate loss. The performance of MPDM (always using MR internally) is in most cases worse than SR-HCSS, yet MPDM allows to obtain a specific average distribution, which is not feasible with the sphere shaping schemes. Also note that MR-MPDM and MR-HCSS have similar complexity as they differ only in which compositions are used, but the underlying structure of having a Huffman tree and using MR for CCDM operation is identical. The jagged behavior of MPDM is because the employed pairwise partitioning can, depending on n, give a different number of composition pairs with a varying permutation count, and thus different rate losses. Interestingly, MPDM sometimes has the same performance as SR-HCSS although the rate losses stem from two different effects. For MPDM, the overall rate loss is due to the suboptimal use of the signal space by only considering pairwise partitions. For SR-HCSS, on the other hand, all compositions up to a certain maximum energy are in principle considered, yet the required parallel-amplitude transformation (see Sec. III-A) introduces some penalty. Finally, CCDM has by far the worst rate loss for the considered small-to-medium block lengths.

In Fig. 6, the rate loss penalty with respect to ESS is shown. We observe that the penalty for the HCSS schemes is only significant at very short block lengths. Already at $n = 50$, the penalty is less than 0.01 bit/1D-sym for MR-HCSS and less than 0.015 bit/1D-sym for SR-HCSS. We can clearly see from Fig. 6 how the architecture properties of the HCSS schemes, which are included to facilitate implementation, lead to additional rate loss. First, the power-of-two constraint on the number of permutations of each composition gives some extra rate loss for MR-HCSS. The parallel-amplitude framework used for SR-HCSS results in another penalty, yet allows an efficient LUT-based implementation and highly parallel operation.

B. Decoding Performance

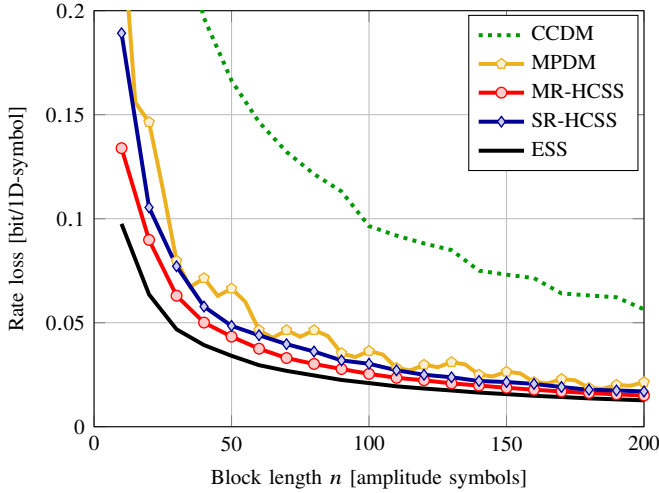For the previously investigated shaping schemes, the frame error rate (FER) after LDPC decoding is shown in Fig. 7.

Fig. 5. Rate loss in bits per 1D amplitude symbol versus *n*. For the two DM schemes, the amplitude distribution is [0.4, 0.3, 0.2, 0.1]. The sphere shaping schemes operate at the same rate as MPDM for each block length.
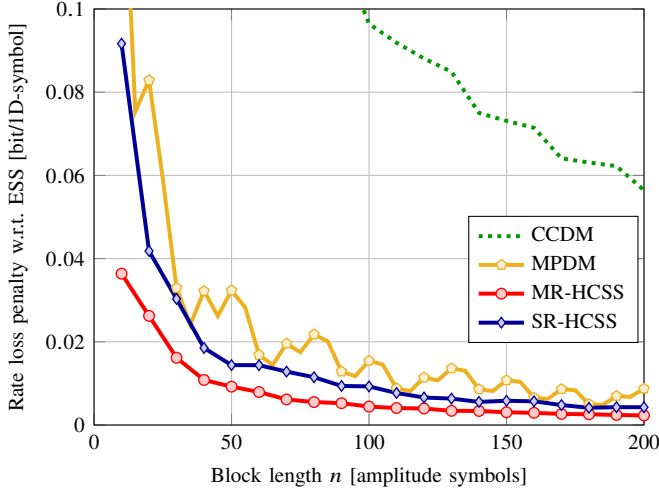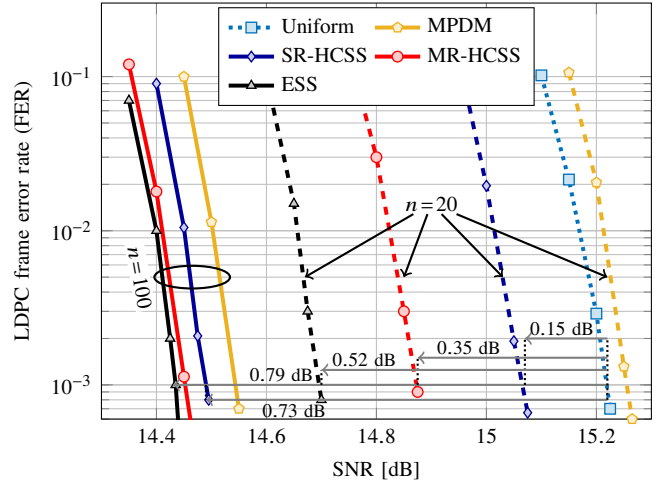

Fig. 7. Frame error rate (FER) after LDPC decoding versus SNR in dB for the AWGN channel and 64QAM. The information rate is 4.5 bit/2D-sym. DVB-S2 LDPC codes of length 64800 bits are used. All SNR differences are in comparison to uniform signaling and evaluated at a FER of 1e-3.

Fig. 6. Rate loss penalty compared to ESS versus block length *n*.

Note that we use FER instead of bit error rate (BER) as its shows the percentage of FEC blocks that contain at least one error and thus, must be disregarded or re-transmitted. We argue that this metric is more insightful than BER which is not related to the block-based nature of current communication systems. We focus on 64QAM at an information rate of 4.5 bit per 2D symbol (bit/2D-sym) over the AWGN channel and LDPC codes from the DVB-S2 standard with 64000 bit length. 50 decoding iterations were used in all cases. To achieve the 4.5 bit/2D-sym information rate, the code rates are 4/5 for shaped 64QAM and 3/4 for uniform 64QAM. All reported shaping gains are evaluated at a FER of 1e-3. For $n = 20$ (dashed curves), a large spread between the different schemes is observed, with SNR savings over uniform 64QAM ranging from -0.03 dB (i.e., a penalty because the rate loss is more dominant than the gain from having a shaped QAM distribution) for MPDM, up to 0.52 dB improvement for ESS.

The HCSS schemes with MR and SR have shaping gains of 0.15 dB and 0.35 dB, respectively. Overall, the schemes differ by more than 0.5 dB of shaping gain. This significant difference in performance is obviously related to the rate loss results discussed in Sec. IV-A. For $n = 100$ (solid curves in Fig. 7), the shaping gains are between 0.73 dB (MPDM) and 0.79 dB (ESS), leading to a gain variation of only 0.06 dB. At such a small difference, implementation aspects are believed to be the key differentiator between the different schemes, see e.g. [37, Sec. 3.3] [33, Sec. VI]. Finally, we note that CCDM has significantly worse than uniform performance for $n = 20$, and a shaping gain of approximately 0.25 dB for $n = 100$ (both not shown in Fig. 7).

C. LUT Size versus Shaping Gain

Figure 8 shows the shaping gain of SR-HCSS for 64QAM versus the size of the LUT required for storing the binomial coefficients. MR is not shown as it has been found to have lower shaping gain than SR for all considered LUT sizes. The two considered information rates of 4 bit/2D-sym and 4.5 bit/2D-sym are obtained with two different configurations, which are 1.5 bit/1D-sym shaping rate and LDPC code rate 5/6 for 4 bit/2D-sym, and 1.85 bit/1D-sym HCSS rate with LDPC code rate 4/5 for 4.5 bit/2D-sym. A similar trend of shaping gain increase with LUT size is observed in Fig. 8 for both configurations, yet with a difference of more than 0.2 dB between them. The main reason is that the 4 bit/2D-sym case can achieve a higher shaping gain at this rate, we operate still close to AWGN capacity, while the 4.5 bit/2D-sym starts to deviate from it due to the limited number of constellation points, see [25, Fig. 5]. In the following, we focus on the 4 bit/2D-sym case. With less than 1 kbit LUT size (corresponding to $n = 20$), a shaping gain of approximately 0.5 dB is achieved. Increasing the LUT to approx. 7 kbit ($n = 40$) gives an improvement of 0.75 dB over uniform 64QAM. The next doubling in block length to $n = 80$ comes at
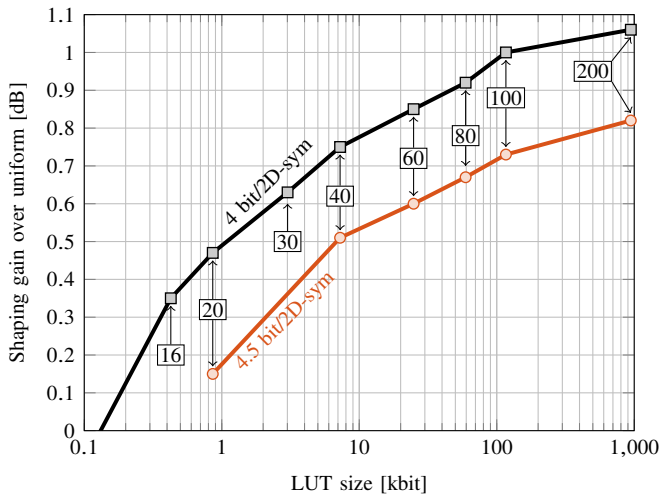
Fig. 8. Shaping gain over uniform 64QAM with SR-HCSS versus LUT size in kbit for 4 bit/2D-sym (black, squares) and 4.5 bit/2D-sym (orange, circles). The numbers in squares are the corresponding block lengths.

a LUT size increase by an order of magnitude, yet gives only 0.17 dB extra shaping gain, amounting to an absolute gain of 0.92 dB. Following this trend, the asymptotic shaping gains of approximately 1.15 dB is achieved to within 0.1 dB with SR-HCSS of length $n = 200$, yet at a relatively large LUT of almost 1 Mbit size.

V. CONCLUSIONS

We have described a sphere shaping scheme that efficiently addresses most of the signal space that lies inside an *n*-dimensional sphere. The proposed Huffman coded sphere shaping (HCSS) is almost as energy-efficient as conventional sphere shaping techniques such as enumerative sphere shaping (ESS). Its construction and mapping methods are, however, highly different from ESS as HCSS uses Huffman coding to determine the composition of the shaped sequence and CCDM methods for mapping and demapping. A numerical analysis of rate loss and decoding performance after LDPC decoding shows that HCSS outperforms all considered DM schemes and performs only slightly worse than ESS for many relevant block lengths, with the performance gap becoming very small at around $n = 50$. As CCDM algorithms are used for mapping and demapping for the desired composition, HCSS can directly leverage any improvements made in CCDM algorithms.

We have further studied two CCDM mapping/demapping methods that are alternatives to the conventional arithmetic coding (AC). Multiset ranking (MR) mainly relies on computing multinomial coefficients and has a reduced number of sequential operations for mapping than AC, which could facilitate high-throughput applications of probabilistic shaping. We have further studied an implementation of subset ranking (SR) for which the binomial coefficients are precomputed and stored in a lookup table. At the expense of a small additional rate loss compared to MR and AC, SR-HCSS is numerically shown to achieve significant shaping gains over uniform 64QAM of approximately 0.5 dB with LUT sizes of

less than 1 kbit. By increasing the LUT size to 100 kbit, the shaping gain increases to approximately 1 dB.

REFERENCES

[1] Feng-Wen Sun and H. C. A. van Tilborg, "Approaching capacity by equiprobable signaling on the Gaussian channel," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1714–1716, Sep. 1993.
[2] R. F. H. Fischer, *Precoding and signal shaping for digital transmission*. New York, NY, USA: John Wiley & Sons, 2005.
[3] H. G. Batshon, I. B. Djordjevic, L. Xu, and T. Wang, "Iterative polar quantization based modulation to achieve channel capacity in ultra-high-speed optical communication systems," *IEEE Photonics Journal*, vol. 2, no. 4, pp. 593–599, Aug. 2010.
[4] T. H. Lotz, X. Liu, S. Chandrasekhar, P. J. Winzer, H. Haunstein, S. Randel, S. Corteselli, B. Zhu, and D. W. Peckham, "Coded PDM-OFDM transmission with shaped 256-iterative-polar-modulation achieving 11.15-b/s/Hz intrachannel spectral efficiency and 800-km reach," *Journal of Lightwave Technology*, vol. 31, no. 4, pp. 538–545, Feb. 2013.
[5] D. S. Millar, T. Fehenberger, T. Koike-Akino, K. Kojima, and K. Parsons, "Coded modulation for next-generation optical communications," in *Proc. Optical Fiber Communication Conference (OFC)*. San Diego, CA, USA: Paper Tu3C.3, Mar. 2018.
[6] R. G. Gallager, *Information theory and reliable communication*. New York, NY, USA: John Wiley & Sons, 1968.
[7] G. D. Forney, "Trellis shaping," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 281–300, Mar 1992.
[8] R. Laroia, N. Farvardin, and S. A. Tretter, "On optimal shaping of multidimensional constellations," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1044–1056, Jul. 1994.
[9] B. P. Smith and F. R. Kschischang, "A pragmatic coded modulation scheme for high-spectral-efficiency fiber-optic communications," *Journal of Lightwave Technology*, vol. 30, no. 13, pp. 2047–2053, Jul. 2012.
[10] L. Beygi, E. Agrell, and M. Karlsson, "Adaptive coded modulation for nonlinear fiber-optical channels," in *IEEE GLOBECOM*. Anaheim, CA, USA: pp. 331–335, 2012.
[11] M. P. Yankov, D. Zibar, K. J. Larsen, L. P. Christensen, and S. Forchhammer, "Constellation shaping for fiber-optic channels with QAM and high spectral efficiency," *IEEE Photonics Technology Letters*, vol. 26, no. 23, pp. 2407–2410, Dec. 2014.
[12] L. Beygi, E. Agrell, J. M. Kahn, and M. Karlsson, "Rate-adaptive coded modulation for fiber-optic communications," *Journal of Lightwave Technology*, vol. 32, no. 2, pp. 333–343, Jan. 2014.
[13] G. Böcherer, P. Schulte, and F. Steiner, "Bandwidth efficient and rate-matched low-density parity-check coded modulation," *IEEE Transactions on Communications*, vol. 63, no. 12, pp. 4651–4665, Dec. 2015.
[14] T. Fehenberger, G. Böcherer, A. Alvarado, and N. Hanik, "LDPC coded modulation with probabilistic shaping for optical fiber systems," in *Proc. Optical Fiber Communication Conference (OFC)*. Los Angeles, CA, USA: Paper Th.2.A.23, Mar. 2015.
[15] F. Buchali, G. Böcherer, W. Idler, L. Schmalen, P. Schulte, and F. Steiner, "Experimental demonstration of capacity increase and rate-adaptation by probabilistically shaped 64-QAM," in *Proc. European Conference and Exhibition on Optical Communication (ECOC)*. Valencia, Spain: Paper PDP.3.4, Sep. 2015.
[16] T. Fehenberger, D. Lavery, R. Maher, A. Alvarado, P. Bayvel, and N. Hanik, "Sensitivity gains by mismatched probabilistic shaping for optical communication systems," *IEEE Photonics Technology Letters*, vol. 28, no. 7, pp. 786–789, Apr. 2016.
[17] Y. C. Gültekin, W. van Houtum, A. Koppelaar, and F. M. Willems, "Enumerative sphere shaping for wireless communications with short packets," *arXiv*, Mar. 2019. [Online]. Available: https://arxiv.org/abs/1903.10244
[18] G. Böcherer, P. Schulte, and F. Steiner, "High throughput probabilistic shaping with product distribution matching," *arXiv preprint arXiv:1702.07510*, Feb. 2017.
[19] J. Cho, X. Chen, S. Chandrasekhar, G. Raybon, R. Dar, L. Schmalen, E. Burrows, A. Adamiecki, S. Corteselli, Y. Pan *et al.*, "Trans-atlantic field trial using probabilistically shaped 64-QAM at high spectral efficiencies and single-carrier real-time 250-Gb/s 16-QAM," in *Proc. Optical Fiber Communication Conference (OFC)*. Los Angeles, CA, USA: Paper Th5B.3, Mar. 2017.
[20] T. Sasai, A. Matsushita, M. Nakamura, S. Okamoto, F. Hamaoka, and Y. Kisaka, "Laser phase noise tolerance of uniform and probabilistically-shaped QAM signals for high spectral efficiency systems," *Journal of Lightwave Technology*, 2019.

[21] T. Fehenberger, A. Alvarado, G. Böcherer, and N. Hanik, "On probabilistic shaping of quadrature amplitude modulation for the nonlinear fiber channel," *IEEE/OSA Journal of Lightwave Technology*, vol. 34, no. 22, pp. 5063–5073, Nov. 2016.

[22] J. Renner, T. Fehenberger, M. P. Yankov, F. Da Ros, S. Forchhammer, G. Böcherer, and N. Hanik, "Experimental comparison of probabilistic shaping methods for unrepeated fiber transmission," *IEEE/OSA Journal of Lightwave Technology*, vol. 35, no. 22, pp. 4871–4879, Nov. 2017.

[23] E. Sillekens, D. Semrau, G. Liga, N. A. Shevchenko, Z. Li, A. Alvarado, P. Bayvel, R. I. Killey, and D. Lavery, "A simple nonlinearity-tailored probabilistic shaping distribution for square QAM," in *Proc. Optical Fiber Communication Conference (OFC)*. Paper M3C.4, Mar. 2018.

[24] P. Schulte and G. Böcherer, "Constant composition distribution matching," *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 430–434, Jan. 2016.

[25] T. Fehenberger, D. S. Millar, T. Koike-Akino, K. Kojima, and K. Parsons, "Multiset-partition distribution matching," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 1885–1893, Mar. 2019.

[26] M. Pikus and W. Xu, "Arithmetic coding based multi-composition codes for bit-level distribution matching," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, Apr. 2019.

[27] J. Cho, "Prefix-free code distribution matching for probabilistic constellation shaping," *IEEE Transactions on Communications*, vol. 68, no. 2, Feb. 2020.

[28] P. Schulte and F. Steiner, "Shell mapping for distribution matching," *arXiv preprint arXiv:1803.03614*, Mar. 2018.

[29] T. Yoshida, M. Karlsson, and E. Agrell, "Hierarchical distribution matching for probabilistically shaped coded modulation," *Journal of Lightwave Technology*, vol. 37, no. 6, pp. 1579–1589, Mar. 2019.

[30] D. S. Millar, T. Fehenberger, T. Yoshida, T. Koike-Akino, K. Kojima, N. Suzuki, and K. Parsons, "Huffman coded sphere shaping with short length and reduced complexity," in *Proc. European Conference on Optical Communications (ECOC)*, Dublin, Ireland, Sep. 2019.

[31] T. Fehenberger, D. S. Millar, T. Koike-Akino, K. Kojima, and K. Parsons, "Parallel-amplitude architecture and subset ranking for fast distribution matching," *IEEE Transactions on Communications*, Jan. 2020.

[32] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 726–740, Jul. 1990.

[33] Y. C. Gültekin, T. Fehenberger, A. Alvarado, and F. M. J. Willems, "Probabilistic shaping for finite blocklengths: distribution matching and sphere shaping," *arXiv*, Sep. 2019. [Online]. Available: https://arxiv.org/abs/1909.08886

[34] G. D. Forney, Jr., R. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient modulation for band-limited channels," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 5, pp. 632–647, Sep. 1984.

[35] T. Fehenberger and A. Alvarado, "Analysis and optimisation of distribution matching for the nonlinear fibre channel," in *Proc. European Conference on Optical Communications (ECOC)*, Dublin, Ireland, Sep. 2019.

[36] M. Pikus and W. Xu, "Bit-level probabilistically shaped coded modulation," *IEEE Communications Letters*, vol. 21, no. 9, pp. 1929–1932, Sep. 2017.

[37] T. Fehenberger, D. S. Millar, T. Koike-Akino, K. Kojima, and K. Parsons, "Mapping strategies for shortlength probabilistic shaping," in *Proc. European Conference on Optical Communications (ECOC)*, Dublin, Ireland, Sep. 2019.