Multiset-Partition Distribution Matching

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

Abstract—Distribution matching is a fixed-length invertible mapping from a uniformly distributed bit sequence to shaped amplitudes and plays an important role in the probabilistic amplitude shaping framework. With conventional constant-composition distribution matching (CCDM), all output sequences have identical composition. In this paper, we propose multiset-partition distribution matching (MPDM) where the composition is constant over all output sequences. When considering the desired distribution as a multiset, MPDM corresponds to partitioning this multiset into equal-size subsets. We show that MPDM allows to address more output sequences and thus has lower constraints on the partitioning, a constructive MPDM algorithm is proposed which comprises two parts. A variable-length prefix of the binary data word determines the composition to be used, and the remainder of the input word is mapped with a conventional CCDM algorithm, such as arithmetic coding, according to the chosen composition. Simulations of 64-ary quadrature amplitude modulation over the additive white Gaussian noise channel demonstrate that the block-length saving of MPDM over CCDM for a fixed gap to capacity is approximately a factor of 2.5 to 5 at medium to high signal-to-noise ratios (SNRs).

Index Terms—Distribution Matching, Probabilistic Amplitude Shaping, Coded Modulation.

I. INTRODUCTION

The combination of high-order modulation, such as quadrature amplitude modulation (QAM), and strong binary codes (such as turbo-codes [1] or low-density parity-check codes [2]) that operate within a fraction of a decibel (dB) of the additive white Gaussian noise (AWGN) channel capacity [3] have become standardized in many digital communication systems. Bit-interleaved coded modulation (BICM) has achieved universal adoption, due to its low complexity and close-to-optimal performance [4]. Most coded modulation systems employ uniform signaling where each constellation point is sent with equal probability. A method to further increase the information rates is to employ constellation shaping, which gives signal-to-noise-ratio (SNR) improvements of up to 1.53 dB for the AWGN channel [5, Sec. 4.1.3] [6, Sec. IV-B] [7, Sec. IV-B]. In general, there are two flavors to constellation shaping, which are geometric shaping (with equiprobable symbols drawn from an irregularly spaced constellation), and probabilistic shaping (non-uniformly distributed symbols with a regular constellation). We focus on probabilistic shaping in this paper.

Various techniques have been devised to integrate probabilistic shaping into a coded modulation system (see [8, Sec. II] for a review). Recently, probabilistic amplitude shaping (PAS) [8] has been proposed in which the shaping blocks are placed outside the forward error correction (FEC) encoder and decoder (see Fig. 1). This reverse-concatenation principle allows a seamless integration into existing BICM systems that, for complexity reasons, typically employ binary FEC and avoid demapper-decoder iterations. Since its proposal, PAS has attracted a lot of attention, particularly in fiber-optic communications [9]–[13]. We focus on PAS as shaping framework in this paper.

An integral subsystem of a PAS system is the mapping function from the uniformly distributed data bits to shaped amplitudes and its inverse mapping. In [8, Sec. V], constant-composition distribution matching (CCDM) is employed. In simplified terms, CCDM is a fixed-length invertible operation that maps a block of Bernoulli-\(\frac{1}{2}\) distributed data bits to a sequence of shaped amplitudes [14]. Under the constant-composition principle, each output sequence must have an identical empirical distribution. The design of distribution matchers that allow to approach channel capacity is closely related to homophonic coding, see the recent review [15] for details.

Although the principle is straightforward, designing CCDMs suitable for real-time processing is challenging, largely for the following two reasons. Any finite-length DM fundamentally suffers from a rate loss that increases with decreasing length. Hence, it would be beneficial to have CCDM block lengths of 500 shaped output symbols or more, as can be seen from, e.g., [14, Fig. 2]. The most widely used algorithm for implementing CCDM is arithmetic coding [14, Sec. IV], which is an inherently sequential algorithm. The combination of sequential mapping and long block lengths currently makes a real-time implementation of CCDM a highly challenging task, particularly in the context of optical fiber communications where symbol rates may be 30 Gbaud or more. Hence, improved CCDM algorithms with reduced serialism must be devised, or architectures must be sought that allow to reduce the block length at equal performance. This paper is devoted to the latter.

In this work, we examine distribution matching techniques for which the constant-composition property of conventional CCDM is lifted. Non-constant-composition DMs are based on the principle that the ensemble average over all output sequences must have the desired composition, as opposed to the CCDM principle that every output has identical composition. While the removal of this constraint enables large gains over CCDM in the range of block lengths where brute-

---

1Note that PAS is not restricted to the use of algebraic distribution matchers (DMs) such as CCDM.
force computation or numerical optimization are feasible, these
techniques are impossible in the block-length regime where
low absolute rate loss is achievable. For example, a DM
with a sequence length of merely 10 symbols and 1.5 bits
of entropy for 4 shaped amplitudes will select $2^{15}$ sequences
from a possible $2^{20}$. Due to the scaling of this combinatorial
problem, it is clear that we need to have a constructive
algorithm for efficiently generating non-constant-composition
distribution matchers.

We propose multiset-partition distribution matching
(MPDM), which forms output sequences that are equal-length
partitionings of a multiset with the desired distribution. In
the binary-partitioning case, for example, we consider sequences
which pairwise follow the target distribution. This set of
sequences will have the set of CCDM sequences as a subset,
and is therefore a generalization of CCDM. We demonstrate
numerically that MPDM requires significantly shorter block
lengths than CCDM at a particular rate loss, which facilitates
realization in hardware.

MPDM can be viewed as a well-balanced combination of
multiple CCDM instances that each have the same alphabet
size as the target distribution. It is thus fundamentally different
from bit-level distribution matching [16], [17] where the target
distribution is factorized such that parallel CCDMs can be
used for the constituent binary distributions. Our approach,
in contrast, is more general in that the alphabet size per
CCDM remains unchanged. MPDM is thus compatible to
the aforementioned bit-level distribution matching, and also
benefits from low-complexity CCDM implementations, such as those proposed in [18]–[21].

In order to simplify the implementation of MPDM, two
constraints on the choice of sequences are imposed. Firstly,
pairwise partitioning is deployed where each constituent com-
position (and thus shaped-amplitude sequence) has a com-
plement such that their average has the desired composition.
By further requiring the number of sequences of a particular
composition to be a power of 2, MPDM with a binary-tree
structure is enabled. Hence, MPDM can be implemented by
splitting the binary data word into a prefix that selects the
composition, and a payload that is mapped onto a symbol
sequence in the conventional CCDM fashion. In this paper,
we focus on implementation aspects and performance com-
parisons of distribution matchers. A numerical analysis finds
that pairwise tree-based MPDM achieves significantly lower
rate loss and thus better AWGN performance than conventional
CCDM. To the best of our knowledge, the proposed MPDM is
the first distribution matcher that lifts the constant-composition
principle for a fixed alphabet size.

II. FUNDAMENTALS OF DISTRIBUTION MATCHING
A. Distribution Matching and Notation

A DM is an injective mapping from a sequence of length $k$
of uniformly distributed data bits $U$ to $n$ shaped amplitudes.
Its integration in the PAS framework is shown in Fig. 1. We
consider fixed-length block-wise distribution matching only
since variable-length DMs have practical disadvantages
such as varying buffer sizes. For finite-length DM, the target
distribution $P_A$ must be quantized to $P_{\tilde{A}}$ such that the number
of occurrences of each amplitude is integer. Following [14],
[22], this quantization is carried out to minimize informational
divergence between $P_A$ and $P_{\tilde{A}}$. The resulting amplitudes $\tilde{A}$
have the probability mass function (PMF) $P_{\tilde{A}}$, also referred
to as type $\tilde{A}$ [23, Sec. 11.1], [24, Sec. II], on the alphabet $\mathcal{A} = \{a_1,\ldots,a_{|\mathcal{A}|}\}$.

Consider a DM output sequence $x^n = \{x_1, x_2, \ldots, x_n\}$ of
length $n$ where each element $x_j$ with $j \in \{1,\ldots,n\}$ is chosen
from $\mathcal{A}$ according to $P_{\tilde{A}}$. The number of occurrences $n(a_i)$
of an amplitude $a_i$ in the sequence $x^n$ is

$$n(a_i) = \{\{ j : x_j = a_i \} \}, \quad j = 1, \ldots, n, \quad i = 1, \ldots, |\mathcal{A}|, \quad (1)$$

and we have $\sum_{i=1}^{\mathcal{A}} n(a_i) = n$. In the following, we write $n_i$
instead of $n(a_i)$ to indicate the number of occurrences of the
$i$th amplitude $a_i$. We call the ordered set of occurrences $C = \{n_1, \ldots, n_{|\mathcal{A}|}\}$ a composition, which has the type $P_{\tilde{A}}$. We say
that a sequence has composition $C$ if (1) corresponds to $C$. The
set of unique permutations of $x^n$ for a given $C$ is referred
to as type class [23, Sec. 11.1], and its size is the multinomial
coefficient $M(C) = \binom{n}{n_1, n_2, \ldots, n_{|\mathcal{A}|}}$.

$$M(C) = \binom{n}{n_1, n_2, \ldots, n_{|\mathcal{A}|}} = \frac{n!}{n_1! n_2! \cdots n_{|\mathcal{A}|}!}. \quad (2)$$

B. Constant Composition Distribution Matching (CCDM)

In the CCDM approach [14], each of the $2^k$ output
sequences $x^n$ is of type $P_{\tilde{A}}$, and we denote the single typical
CCDM output composition as $C_{\text{typ}} = \{nP_{\tilde{A}}(a_1), \ldots, nP_{\tilde{A}}(a_{|\mathcal{A}|})\}$. The constant-composition mapping
of a uniform input sequence to a sequence that has $C_{\text{typ}}$ is
denoted as $f_{\text{ccdm}}(C_{\text{typ}})$ and can, for example, be carried out
via arithmetic coding [14, Sec. IV]. The number of input bits
for CCDM of a particular $C_{\text{typ}}$ is given by

$$k = \lceil \log_2 M(C_{\text{typ}}) \rceil, \quad (3)$$

where $\lceil \cdot \rceil$ denotes rounding down to the closest integer. From
(3), we can compute the finite-length rate loss [16, Eq. (4)]

$$R_{\text{loss}} = H(\tilde{A}) - \frac{k}{n}, \quad (4)$$

where $H(\tilde{A})$ denotes the channel entropy. From
(4), we can compute the finite-length rate loss of a specific
CCDM as $R_{\text{loss}}$.
where \( H(\cdot) \) denotes entropy in bits. The rate loss vanishes for large \( n \) [14, Eq. (23)], which means that an infinite-length CCDM can achieve the target rate without any rate loss. For distribution matching with fixed \( n \) and \( C \), it is desirable to make \( k \) as large as possible to in order to minimize the rate loss. In the following, we introduce a new class of distribution matcher that has a significantly smaller rate loss than a conventional CCDM.

### III. Multiset-Partition Distribution Matching

#### A. Principle

MPDM is based on the observation that every possible DM output sequence \( x^k \) is not necessarily of type \( P_A \) (or equivalently have the composition \( C \)) in order to achieve on average the target distribution. As the input bits \( U \) are uniformly distributed and a DM establishes an injective mapping, it is by the law of large numbers sufficient if the ensemble average of all output sequences has the target composition.\(^2\)

Thus, a general MPDM uses those output sequences whose average of all output sequences has the target composition.

Thus, a general MPDM uses those output sequences whose average of all output sequences has the target composition.

\[
\frac{\sum_i^{N_{\text{comp}}} C_i \cdot c_i}{\sum_{i=1}^{N_{\text{comp}}} c_i} = C_{\text{typ}},
\]

where \( l \) indexes the \( N_{\text{comp}} \) possible compositions of the MPDM output sequences and \( c_i \) is the number of occurrences of \( C_i \) at the MPDM output, with \( 0 \leq c_i \leq M(C) \). The possible compositions \( C_i \) can be obtained by exhaustive search, and the choice of \( c_i \) depends on the partitioning constraints, see Sec. III-B for the binary (i.e., pairwise) case. The general partitioning problem (5) states that the average type of all sequences that are the output of a DM must be \( P_A \). The number of distinct compositions is given by

\[
N_{\text{comp}} = \binom{n + |A| - 1}{n},
\]

which can be proven, for example, with the stars-and-bars technique [25, Sec. II-5].

MPDM can also be viewed in the context of the energy of an \( n \)-dimensional sphere [26]. In the CCDM approach, all output sequences are on the constant-energy surface of such a sphere in the \( n \)-dimensional signal space, and, as pointed out in [27, Sec. III-C], not all points on that surface are used. In contrast, MPDM combines spheres of different energy levels such that the target distribution \( P_A \) is achieved at its output.\(^3\)

As will be evaluated in detail in Sec. IV, this property gives reduced rate loss compared to CCDM.

**Example 1 (Non-Constant Composition):** Figure 2 shows a set of figures demonstrating the concept of non-constant composition.

We have the typical composition \( C_{\text{typ}} = \{4, 3, 2, 1\} \) for CCDM with \( n = 10 \), which gives the entropy \( H(A) = 1.85 \) bits. The total number of distinct sequences which have this composition is \( M(C_{\text{typ}}) = 12600 \). This determines the rate of the binary distribution matcher to be \( \log_2(12600/2)/10 = 1.3 \) bits per symbol. By (4), we have a rate loss \( R_{\text{loss}} \) of 1.85–1.3 = 0.55 bits for CCDM. By combining one sequence that has \( C_1 = \{4, 2, 3, 1\} \) and one of \( C_2 = \{4, 4, 1, 1\} \), the average behavior remains that of the original composition \( C_{\text{typ}} \).

The number of distinct sequences with \( C_1 \) and \( C_2 \) are \( M(C_1) = 12600 \) and \( M(C_2) = 6300 \), respectively. If, in addition to \( C_{\text{typ}} \), these two compositions are used, \( C_1 = C_2 = 6300 \) additional sequences can each be generated such that (5) is fulfilled. Hence, by considering all three compositions in Fig. 2, we may now use 12600 sequences from \( C_{\text{typ}} \), 6300 sequences from \( C_1 \); and 6300 from \( C_2 \) — 25200 in total. This increases the rate of the non-constant composition distribution matcher by 0.1 bit/symbol and reduces the rate loss from 0.55 bits to 0.45 bits.

Suppose the number of input bits \( k \) and the DM output length \( n \) is fixed. Then, the accumulated composition of all utilized output sequences is \( C_{\text{acc}} = 2^k \cdot C_{\text{typ}} \). The non-trivial problem is now to find the partitioning of \( C_{\text{acc}} \) into \( 2^k \) integer subsets while fulfilling (5) and obeying two constraints in the subset selection: the sum of the integer elements in each subset must be equal to \( n \) in order to have a fixed-length DM, and each subset cannot occur more often than their multinomial coefficient \( M(C) \) (see (2)) such that an injective mapping function is established.

**Example 2 (General MPDM):** Consider a DM with \( k = 17 \), \( n = 10 \), and \( P_A = \{0.4, 0.3, 0.2, 0.1\} \). We have \( C_{\text{typ}} = \{4, 3, 2, 1\} \), and the accumulated composition is \( C_{\text{acc}} = \{524288, 393216, 262144, 131072\} \). A general MPDM seeks to find those \( 2^{17} \) out of the \( 2^{20} \) possible DM output sequences whose number of occurrences of each amplitude gives \( C_{\text{acc}} \), i.e., that fulfills (5). By (6), there are \( N_{\text{comp}} = 286 \) compositions for these sequences. The problem is equivalent to finding the non-unique integer sets (each of which corresponds to a particular \( C \)) that sum up to \( C_{\text{acc}} \), given the constraints that the sum over each set must be \( n = 10 \) and that each set occurs at most \( M(C) \) times.

Many partitioning problems are known to be NP-complete [30, Sec. 3.1.5], yet algorithms giving approximate solutions with reasonable complexity are known for special cases [30, Sec. 4.2]. While the exact complexity of the considered constrained multiset partitioning problem is unknown to us, it also irrelevant since finding a solution does not lead to a constructive algorithm for MPDM design. In other words, even if a solution to the partitioning problem could be found, it would remain a challenging task to establish an implementable
mapping function between DM input and output sequences, in particular if the DM dimensions prohibit the use of a lookup table. By imposing some structure onto the partitioning, a construction of a MPDM device is made feasible at the expense of a slightly increased rate loss, as we will show next.

B. Pairwise MPDM

To facilitate the implementation of MPDM, we simplify the general partitioning problem (5) by considering pairwise typical sequences only. Note that other structured partitioning schemes, for instance into triples or quadruples, are also possible.\(^4\) In this pairwise case, we require that for every composition \(C_l\), a complementary composition \(\overline{C}_l\) must exist such that

\[
C_l + \overline{C}_l = 2 \cdot C_{\text{typ}}. \tag{7}
\]

The unique pairs can be found in a relatively straightforward fashion by exhaustive search, i.e., by cycling through all possible compositions (of which there are \(N_{\text{comp}}\) as per (6)) and discarding those that, when added to their complement as per (7) do not give \(C_{\text{typ}}\). The number of valid pairs, denoted as \(\{C_l, \overline{C}_l\}\), that fulfill the relation (7) can be computed by modifying (6) with the inclusion-exclusion method \([31, \text{Theorem 4.2}]\), taking into account that certain compositions can never occur in a constrained setting such as the considered pairwise MPDM. The idea of inclusion-exclusion is to start with the unconstrained number of compositions \(N_{\text{comp}}\) as per (6) and remove from it those compositions that, on a single amplitude basis, never lead to a desired distribution. Next, invalid compositions for all pairwise combinations of two amplitudes are included as they have been excluded twice in the previous step. Triple-wise combinations must be excluded again, and so forth. This alternation between inclusion and exclusion is repeated \(|\mathcal{A}|\) times.

Example 3 (Inclusion-Exclusion): Consider the case of Example 1 where \(C_{\text{typ}} = \{4,3,2,1\}\) and \(n = 10\). According to (6), we have \(N_{\text{comp}} = 286\). For the first amplitude \(a_1\), compositions with more than 8 occurrences of \(a_1\) cannot be combined in a pairwise manner while fulfilling (7). Thus, the four compositions with either 9 or 10 occurrences of \(a_1\) are excluded. This is repeated for \(a_2, a_3, \text{and} a_4\), resulting in a respective reduction of valid compositions by 20, 56, and 120, which gives an interim composition count of 86. However, some compositions have been excluded twice (such as \(C = \{0,7,0,3\}\) violating both the \(a_2\) and the \(a_4\) constraint), and thus have to be included once again. In total, there is 11 such excessive exclusions, giving a final composition count of 97 for pairwise partitioning. This corresponds to 49 distinguishable pairs out of which one is the degenerate CCDM “pair” \(\{C_{\text{typ}}, C_{\text{typ}}\}\). Note that triple- and quadruple-wise combinations of amplitudes do not have to be considered in this simple example.

For pairwise partitioning, the MPDM output sequences that have \(C_l\) or \(\overline{C}_l\) should occur in an equiprobable manner, which implies that the total number of permutations for a pair is governed by the constituent composition that has fewer permutations. We denote the permutation count of a pair \(\{C_l, \overline{C}_l\}\) as

\[
M\left(\{C_l, \overline{C}_l\}\right) = \begin{cases} 
2 \cdot \min \left( M\left(C_l\right), M\left(\overline{C}_l\right) \right), & C_l \neq \overline{C}_l, \\
M\left(C_{\text{typ}}\right), & C_l = \overline{C}_l.
\end{cases} \tag{8}
\]

where the first case corresponds to non-degenerate pairs and the latter case is the degenerate CCDM “pair”. For a pairwise MPDM with \(N_{\text{purs}}\) distinguishable pairs \(\{C_l, \overline{C}_l\}\) that satisfy (7), the total number of permutations is

\[
N_{\text{perm}} = \sum_{l=1}^{N_{\text{purs}}} M\left(\{C_l, \overline{C}_l\}\right). \tag{9}
\]

Note that \(\{C_l, \overline{C}_l\}\) is invariant to permutations of the compositions and switching the two compositions (i.e., \(\{C_l, \overline{C}_l\}\) instead of \(\{C_l, \overline{C}_l\}\)) does not give a new unique pair. The rate loss improvement of MPDM over CCDM (see Sec. IV) is the result of including non-degenerate pairs in (9) in addition to the typical CCDM composition.\(^5\) For any DM with binary input, (9) is rounded down to the nearest power of 2, i.e., we have

\[
2^k = \left[ N_{\text{perm}} \right]. \tag{10}
\]

The same requirement is made for binary CCDM in (3).

Example 4 (Pairwise MPDM): Consider \(P_A = [0.4415, 0.3209, 0.1654, 0.0722]\) (taken from [14, Example A]) and \(n = 10\). We use [22, Algorithm 2] to quantize \(P_A\) to \(P_{\tilde{A}} = [0.4, 0.3, 0.2, 0.1]\), which has \(H(\tilde{A}) = 1.85\) bits and \(C_{\text{typ}} = \{4,3,2,1\}\), see Example 1. For pairwise MPDM, there are \(N_{\text{purs}} = 49\) pairs (including the degenerate one) that fulfill (7), see Example 3. The new total permutation count is 164214, which increases the number of input bits to \(k = 17\) and thus reduces the rate loss to 0.15 bits.

Although considering pairwise-typical compositions greatly simplifies the search for valid partitionings, the implementation of such a pairwise MPDM is not straightforward if a large lookup table is not to be used. In the following, we impose another constraint, again at the expense of transmission rate, that enables implementation of MPDM with reasonable complexity.

C. Implementation with a Binary Tree Structure

Any DM with binary input uses \(\left[ N_{\text{perm}} \right]\) out of \(N_{\text{perm}}\) output sequences. In order to design an implementable pairwise MPDM, we require, in addition to (7), that the number of permutations \(M\left(\{C_l, \overline{C}_l\}\right)\) of each pair \(\{C_l, \overline{C}_l\}\) must be a power of 2. A pair \(\{C_l, \overline{C}_l\}\) thus represents an integer \(k_l\) bits, and (8) becomes

\[
2^{k_l} = \begin{cases} 
2 \cdot \min \left( \left[ M\left(C_l\right) \right], \left[ M\left(\overline{C}_l\right) \right] \right), & C_l \neq \overline{C}_l, \\
\left[ M\left(C_{\text{typ}}\right) \right], & C_l = \overline{C}_l.
\end{cases} \tag{11}
\]

\(^4\)With MPDM, \(P_{\tilde{A}}\) can also be requantized such that the divergence between the target PMF and \(P_{\tilde{A}}\) is reduced. Potential benefits of this approach remain for future work.

\(^5\)It is only in the trivial case where \(C_{\text{typ}}\) has only one non-zero element that the number of permutations of CCDM and MPDM is identical.
The total number of permutations with this power-of-2 constraint is thus

\[ 2^k = \sum_{l=1}^{N_{\text{pairs}}} 2^{k_l}, \]

(12)

where \( N_{\text{pairs}} \) out of the initially available \( N_{\text{pairs}} \) pairs are selected to maximize \( N_{\text{perms}} \) (while keeping it a power of 2 to have a binary DM) and thus, to maximize \( k \). The selection of pairs can be done by sorting \( \{ C_i, \overline{C}_i \} \) according to \( k_l \) in descending order and including only the first \( N_{\text{pairs}} \) pairs in that ranked list until \( k \) is integer. We note that the constraint (11) can lead to fewer permutations than for unconstrained pairwise MPDM (see (9)) and thus an increased rate loss.

**Example 5 (Pairwise MPDM with Binary-Tree Structure):** Consider the case of Example 4. With (11) and (12), the total number of permutations is computed to be 122688, which gives \( k = 16 \) input bits and \( R_{\text{loss}} = 0.25 \) bits, see the marker in Fig. 4. The number of pairs that is necessary to address the 16 bits is \( N_{\text{pairs}} = 9 \) out of the initial \( N_{\text{pairs}} = 49 \). Note that \( C_{\text{typ}} \) is not included in these 9 pairs as they already maximize the integer-valued \( k \).

We outline in the following how a binary tree with variable-length prefix \( p_t \) can be constructed such that individual compositions can be addressed with a constant \( k \). Dividing (12) by \( 2^k \) gives

\[ \sum_{l=1}^{N_{\text{pairs}}} 2^{-(k-k_l)} = 1, \]

(13)

which is the Kraft inequality [23, Theorem 5.2.1] for a binary alphabet and fulfilled with equality. Thus, the lengths of the prefixes and of the bits to be mapped always sum up to \( k \):

\[ p_t + k_l = k, \quad l = \{1, \ldots, N_{\text{pairs}}\}. \]

(14)

By (12)–(14), we are effectively enforcing a selection of the compositions that follows a dyadic distribution and use a Huffman code for determining the prefix [23, Sec. 5.6]. The probability of the composition pair \( \{ C_l, \overline{C}_l \} \) being selected is \( 2^{-p_t} \) such that \( p_t \) bits of prefix can address it without loss.

The power-of-2 constraint of (12) thus enables the implementation of MPDM in a binary-tree structure as follows. The \( N_{\text{pairs}} \) different pairs are sorted by their \( k_l \) in ascending order, and the two compositions within a pair are labeled 0 and 1, respectively. Note that this single-bit label is omitted in the special case of the CCDM composition \( C_{\text{typ}} \). The two pairs with the smallest \( k_l \) (i.e., the least permutations) form a branch, with one element labeled 1 and the other 0. If more than one branch remains, i.e., if there are pairs that have not been used in the tree yet, the merging and labeling process is repeated. When only a single branch remains, the prefix tree is completed. This standard source-coding technique gives an optimal labeling for the binary tree.

Once this binary tree is generated, the mapping from \( k \)-bit uniform data word to shaped amplitude sequence is done by splitting the MPDM input sequence into three parts. The first \( p_t = k - k_l \) bits are the prefix that identifies the pair. The next bit chooses the composition within that pair. For the mapping of the final \( k_l - 1 \) bits onto the shaped symbols according to the selected composition, conventional CCDM based on arithmetic coding can be employed [14, Sec. IV]. This tree-structure design is illustrated in Fig. 3.

At the receiver, inverse MPDM of a shaped sequence \( x^n \) must be performed in order to recover the initially transmitted data word. Note that MPDM is designed as an invertible function and hence will not introduce any errors if the input, i.e., the FEC decoder output, is error-free. First, the composition of \( x^n \) is determined by, e.g., a simple histogram operation, from which the binary prefix of length \( k - k_l + 1 \) can be looked up. In order to obtain the remaining \( k_l - 1 \) payload bits, an inverse CCDM algorithm using arithmetic coding can be employed. This recovers the transmitted bit sequence. Note that the entire MPDM codeword must be detected before inverse MPDM mapping can begin because the composition determines the prefix. For potential real-time processing, demapping therefore must be parallel on a per-codeword basis, i.e., processing the next codeword must begin before the current one finishes.

The key steps for constructing a MPDM are summarized in Algorithm 1. The resulting MPDM of rate \( \frac{k}{n} \) will have pairwise partitioning and use a binary-tree structure for selecting the component compositions.

**Algorithm 1 Construction of a Pairwise Binary-Tree MPDM**

**Require:** \( n, P_A \rightarrow \) MPDM output length, target distribution

1: Determine typical composition \( C_{\text{typ}} \) → See Sec. II-A
2: Find all composition pairs \( \{ C_l, \overline{C}_l \} \) with \( l = \{1, \ldots, N_{\text{pairs}}\} \)
3: Restrict usage per pair to largest power of two that is not greater than \( 2 \cdot \min \left( M(C_l), M(\overline{C}_l) \right) \) → See (11)
4: Sort the pairs in descending order by the usage count of step 3
5: Choose the first \( N_{\text{pairs}} \) such that the total number of MPDM permutations is \( 2^k \) where \( k \) is as large as possible
6: Each pair \( \{ C_l, \overline{C}_l \} \) chosen in step 5 is assigned a prefix of length \( k - k_l \) bits
7: A one-bit prefix determines the composition within a pair
In the next section, we compare pairwise MPDM with a binary-tree-structured implementation (which we simply refer to as MPDM) to conventional CCDM.

### IV. NUMERICAL RESULTS

This section numerically studies the rate loss and AWGN performance of pairwise MPDM with the tree-structure design outlined in Sec. III. Rate loss is computed with (4), where the input length \( k \) is computed from (3) for CCDM and from (12) for MPDM. For the AWGN channel results, we consider quadrature amplitude modulation (QAM) channel input as the concatenation of two one-dimensional amplitude-shift keying (ASK) symbols. The figure of merit is the achievable information rate (AIR) for bit-metric decoding reduced by the DM rate loss,

\[
\text{AIR}_{\text{DM}} = \mathbb{H}(B) - \sum_{i=1}^{m} \mathbb{H}(B_i|Y) - R_{\text{loss}},
\]

where \( B = [B_1, \ldots, B_M] \) describes the binary input into the modulator and \( Y \) the symbolwise channel output, see Fig. 1. The derivation of (15) is given in the Appendix. A quantized version of the optimal Maxwell-Boltzmann distribution [26, Sec. IV] for each SNR is used as \( P_A \). We focus on 64QAM for the AWGN rate analysis, emphasizing that MPDM is feasible with any modulation format that is compatible with PAS.

Fig. 4 shows rate loss over block length for the target PMF of [14, Example A]. We observe that the pairwise MPDM achieves a smaller rate loss compared to CCDM for all block lengths. For a rate loss of 0.025 bits per amplitude symbol, MPDM can operate with approximately \( n = 140 \) symbols, whereas a conventional CCDM requires a fourfold increase in length. Note that the jagged shape of CCDM and MPDM rate loss is due to flooring operations in (3) and (10), respectively.

In Fig. 5, AIR\(_{\text{DM}}\) in bits per 2D-symbol (bit/2D-sym) is shown over SNR in dB for 64QAM. In addition to the AIRs for MPDM of short (\( n = 30 \)), medium (\( n = 100 \)), and large size (\( n = 250 \)), the AWGN capacity (solid black), an infinite-length DM without rate loss (dotted) and uniform 64QAM (dashed) are included as references. We observe that MPDM with length as small as \( n = 30 \) has larger AIR than uniform 64QAM over the relevant SNR range. By increasing \( n \) to 250 symbols, the MPDM achieves performance within 0.2 dB of the Shannon bound at an AIR of 4 bit/2D-sym.

In Fig. 6, a comparison of MPDM and CCDM as a function of the block length \( n \) in 1D amplitude symbols is shown for a fixed SNR of 14 dB. At this SNR level and for 64QAM, an infinite-length CCDM without any rate loss operates within approximately 0.1 bit/2D-sym of the AWGN capacity. For
MPDM lengths above 30 symbols, we note that performance is better than that of uniform 64QAM, while CCDM requires at least 80 symbols to overcome the rate loss. MPDM with \( n = 60 \) is able to operate within 0.1 bit/2D-sym of the infinite-length limit and thus within 0.2 bit/2D-sym of capacity. We further note that for \( n = 60 \), MPDM achieves half of the available shaping gain of 0.24 bit/2D-sym. By increasing the MPDM length to \( n = 250 \), 90\% of the shaping gain are attainable.

Fig. 7 shows the SNR gap to capacity in dB over \( \text{AIR}_{\text{DM}} \). For all considered rates, a length-250 MPDM operates within approximately 0.1 dB of its asymptotic limit. When comparing CCDM with \( n = 250 \) to MPDM of various lengths, we observe that the MPDM length reduction is approximately a factor of 2.5 at low AIRs (left gray marker). At \( \text{AIR}_{\text{DM}} = 3 \) bit/2D-sym, MPDM of length 100 has an SNR gap to capacity of 0.3 dB. When increasing \( \text{AIR}_{\text{DM}} \), CCDM approaches (and eventually crosses) the MPDM curve of \( n = 50 \), which corresponds to a fivefold length reduction (right gray marker). For large AIRs beyond 5.5 bit/2D-sym (not shown) where the QAM PMF is close to uniform, the length reduction is up to a factor of 10, indicating that the MPDM benefit depends on how strongly the quantized PMF is shaped. For a heavily shaped distribution, pairwise MPDM only gives few additional permutations over a conventional CCDM. As we have seen from the results in this section, this number and thus the potential input sequence \( k \) increases drastically when the PMF is closer to a uniform one, leading to superior performance of MPDM over CCDM.

**V. Conclusion**

We have proposed a novel distribution matching scheme, referred to as multiset-partition distribution matching (MPDM), which generalizes conventional CCDM by lifting the constant-composition property of the shaped distribution matcher output sequences. By including sequences with many different compositions which have the desired composition on average, MPDM achieves a lower rate loss than CCDM for a fixed block length in all relevant cases of distribution matching. By imposing constraints on the choice of partitionings and their number of occurrences, a computationally efficient, constructive algorithm for distribution matching and dematching is devised. MPDM is numerically found to allow the block length to be reduced by a factor of 4 for the same rate loss as CCDM. AWGN simulations with 64QAM demonstrate that this reduction depends on the SNR (i.e., the target distribution) and amounts to a factor of 2.5 to 5 for a fixed gap to AWGN capacity.

**ACKNOWLEDGMENTS**

The authors would like to thank the anonymous reviewers for their comments which greatly helped to improve the presented paper.

**APPENDIX**

**AIR EVALUATION FOR FINITE-LENGTH DM**

The following derivation shows how an achievable information rate (AIR) for a finite-length DM can be computed from the conventionally estimated AIR (assuming infinite-length DM) and the DM rate loss. Both DM and FEC are considered as codes with rates \( R_{\text{sh}} \) and \( R_{\text{FEC}} \), respectively. We then evaluate the information content per ASK symbol for an infinite-length and finite-length DM.

In the PAS scheme, \( m − 1 \) shaped amplitude bits of each ASK symbol are combined with 1 uniform sign bit. The source of these sign bits can be the uniform data that is to be transmitted or the parity bits of the FEC code. A schematic illustration of such an average ASK-symbol composition is given in Fig. (8). The combined striped areas represent the overall amount of information contained in each symbol. This information content \( I_{\text{PAS}} \) in bits per ASK symbol is

\[
I_{\text{PAS}} = \left( \frac{m - 1}{m} \cdot R_{\text{sh}} + \left( R_{\text{FEC}} - \frac{m - 1}{m} \right) \right) \cdot m,
\]

where the first term inside the brackets corresponds to the information contained in the shaped amplitudes (blue striped...
area in Fig. (8)) and the second term is the information within the sign bits (red striped area), sometimes denoted $\gamma$ [8, Sec. IV-D]. For PAS, $R_{\text{FEC}}$ must be at least $\frac{m-1}{m}$ [8, Sec. IV-B]. The shaping rate $R_{\text{sh}}$ in (16) is defined as

$$R_{\text{sh}} = \begin{cases} \frac{k}{m-1} & \text{for finite-length DM}, \\ \frac{\mathbb{H}(\bar{A})}{m-1} & \text{for infinite-length DM}, \end{cases}$$

(17)

respectively, and thus describes the ratio of information contained in the shaped amplitude bits. With the definitions (16) and (17), we can state a performance measure of a finite-length DM. We define the efficiency $\eta$ of a finite-length DM as the ratio of $I_{\text{PAS}}$ for finite-length and infinite-length DM, i.e.,

$$\eta = \frac{m-1 \cdot \frac{k}{m-1} + (R_{\text{FEC}} - \frac{m-1}{m})}{m-1 \cdot \frac{\mathbb{H}(\bar{A})}{m-1} + (R_{\text{FEC}} - \frac{m-1}{m})} = \frac{\frac{k}{m} + 1 + m \cdot (R_{\text{FEC}} - 1)}{\mathbb{H}(\bar{A}) + 1 + m \cdot (R_{\text{FEC}} - 1)}.$$ 

(18)

(19)

In the following, we use this shaping efficiency to compute AIRs for finite-length DM.

Consider an AIR in bits per symbol that is computed without including any DM rate loss. A highly relevant AIR for PAS and binary FEC is the bit-metric decoding (BMD) rate $R_{\text{BMD}}$ defined as [8, Eq. (63)]

$$R_{\text{BMD}} = \left[ \mathbb{H}(B) - \sum_{i=1}^{m} \mathbb{H}(B_i|Y) \right],$$

(20)

where $B = [B_1, B_2, \ldots, B_m]$ is the binary input vector and $Y$ the channel output (see Fig. 1). The computation of $R_{\text{BMD}}$ can, for example, be carried out via numerical integration if the channel law is known, or in Monte Carlo simulations and by mismatched decoding for an unknown channel [32, Sec. VI]. We use numerical integration for the AWGN results in Sec. IV. Note that $R_{\text{BMD}}$ according to (20) is achievable only for a DM without rate loss. The BMD rate for PAS and a finite-length DM, referred to as $\text{AIR}_{\text{DM}}$, is given by

$$\text{AIR}_{\text{DM}} = \eta \cdot R_{\text{BMD}}.$$ 

(21)

This means that by using a finite-length DM, the information content of each symbol is reduced by $\eta$. Note that (21) also holds for AIRs other than $R_{\text{BMD}}$, such as mutual information. To simplify (21), we consider PAS with capacity-achieving codes that operate at their thresholds, in which case we have

$$R_{\text{BMD}} \equiv I_{\text{PAS}}.$$ 

(22)

Solving (16) for the FEC rate then gives

$$R_{\text{FEC}} = \frac{R_{\text{BMD}}}{m} + \left( 1 - \frac{\mathbb{H}(\bar{A})}{m-1} \right) \cdot \frac{m-1}{m}.$$ 

(23)

By inserting (23) into (19) we get

$$\eta = 1 + \frac{\frac{k}{m} - \mathbb{H}(\bar{A})}{R_{\text{BMD}}},$$

(24) which, inserted into (21), finally gives a simplified expression for the BMD rate of a finite-length DM.

$$\text{AIR}_{\text{DM}} = R_{\text{BMD}} + \frac{k}{n} - \mathbb{H}(\bar{A})$$

(25)

$$= R_{\text{BMD}} - R_{\text{loss}},$$

(26)

where $R_{\text{loss}}$ was introduced in (4). We use (26) in Sec. IV to compare the AIRs of DMs that have different rate losses, in particular of a conventional CCM and the proposed MPDM.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in Proc. International Conference on Communications (ICC), Geneva, Switzerland: vol. 2, pp. 1064–1070, May 1993.

[2] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” IEEE Electronics Letters, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[3] S.-Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” IEEE Communications Letters, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[4] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” IEEE Transactions on Information Theory, vol. 44, no. 3, pp. 927–946, May 1998.

[5] R. F. H. Fischer, Predecoding and Signal Shaping for Digital Transmission. New York, NY, USA: John Wiley & Sons, 2005.

[6] 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.

[7] R. F. D. Forney and G. Ungerboeck, “Modulation and coding for linear Gaussian channels,” IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2384–2415, Oct. 1998.

[8] 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. 4451–4465, Dec. 2015.

[9] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “How to achieve the capacity of asymmetric channels,” IEEE Transactions on Information Theory, vol. 64, no. 5, pp. 3371–3393, May 2018.

[10] F. Buchali, F. Steiner, G. Böcherer, P. Schulte, and W. Idler, “Rate adaptation and reach increase by probabilistically shaped 64-QAM: An experimental demonstration,” Journal of Lightwave Technology, vol. 34, no. 7, pp. 1599–1609, Apr. 2016.

[11] T. Feohenberger, A. Alvarado, G. Böcherer, and N. Hanik, “On probabilistic shaping of quadrature amplitude modulation for the nonlinear fiber channel,” Journal of Lightwave Technology, vol. 34, no. 22, pp. 5063–5073, Nov. 2016.

[12] A. Ghazisaeidi, I. F. de Jauregui Ruiz, R. Rios-Muller, L. Schmalen, P. Tran, P. Brindel, A. C. Meseguer, Q. Hu, F. Buchali, G. Charlet et al., “65Tb/s transoceanic transmission using probabilistically-shaped PDM-64QAM,” in Proc. European Conference on Optical Communications (ECOC). Düsseldorf, Germany: Paper Th.3.C.4, Sep. 2016.

[13] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation with probabilistic shaping for optical fiber systems,” in Proc. Optical Fiber Communication Conference (OFC). Los Angeles, CA, USA: Paper Th.A2.23, Mar. 2015.

[14] 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.

[15] F. Buchali, F. Steiner, G. Böcherer, L. Schmalen, P. Schulte, and W. Idler, “Rate adaptation and reach increase by probabilistically shaped 64-QAM: An experimental demonstration,” Journal of Lightwave Technology, vol. 34, no. 7, pp. 1599–1609, Apr. 2016.

[16] T. Feohenberger, A. Alvarado, G. Böcherer, and N. Hanik, “On probabilistic shaping of quadrature amplitude modulation for the nonlinear fiber channel,” Journal of Lightwave Technology, vol. 34, no. 22, pp. 5063–5073, Nov. 2016.

[17] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “How to achieve the capacity of asymmetric channels,” IEEE Transactions on Information Theory, vol. 64, no. 5, pp. 3351–3393, May 2018.

[18] 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.
[19] G. Böcherer, F. Steiner, and P. Schulte, “Fast probabilistic shaping implementation for long-haul fiber-optic communication systems,” in Proc. European Conference on Optical Communications (ECOC). Gothenburg, Sweden: Paper Tu.2.D.3, 2017.

[20] T. Yoshida, M. Karlsson, and E. Agrell, “Short-block-length shaping by simple mark ratio controllers for granular and wide-range spectral efficiencies,” in Proc. European Conference on Optical Communications (ECOC). Gothenburg, Sweden: Paper Tu.2.D.2, Sep. 2017.

[21] ——, “Low-complexity variable-length output distribution matching with periodical distribution uniformization,” in Proc. Optical Fiber Communication Conference (OFC). San Diego, CA, USA: Paper M4E.2, Mar. 2018.

[22] G. Böcherer and B. C. Geiger, “Optimal quantization for distribution synthesis,” IEEE Transactions on Information Theory, vol. 62, no. 11, pp. 6162–6172, Sep. 2016.

[23] T. M. Cover and J. A. Thomas, Elements of information theory, 2nd ed. New York, NY, USA: John Wiley & Sons, 2006.

[24] I. Csiszár, “The method of types,” IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2505–2523, Oct. 1998.

[25] W. Feller, An introduction to probability theory and its applications. Volume 1, 3rd ed. John Wiley & Sons, 1968.

[26] F. R. Kschischang and S. Pasupathy, “Optimal nonuniform signaling for Gaussian channels,” IEEE Transactions on Information Theory, vol. 39, no. 3, pp. 913–929, May 1993.

[27] Y. C. Gültekin, W. van Houtum, S. Serbetli, and F. M. Willems, “Constellation shaping for IEEE 802.11,” in Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QB, Canada, Oct. 2017.

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

[29] Y. C. Gültekin, F. M. Willems, W. van Houtum, and S. Serbetli, “Approximate enumerative sphere shaping,” in Proc. IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, Jun. 2018.

[30] M. Garey and D. Johnson, Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman, 1979.

[31] R.B.J.T. Allenby and Alan Slomson, How to count: an introduction to combinatorics, 2nd ed., ser. Discrete mathematics and its applications. CRC Press, 2011.

[32] D. Arnold, H.-A. Loeliger, P. Vontobel, A. Kavcic, and W. Zeng, “Simulation-based computation of information rates for channels with memory,” IEEE Transactions on Information Theory, vol. 52, no. 8, pp. 3498–3508, Aug. 2006.