MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Parallel-Amplitude Architecture and Subset Ranking for Fast Distribution Matching

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

TR2020-050 April 18, 2020

Abstract

A distribution matcher (DM) maps a binary input sequence into a block of nonuniformly distributed symbols. To facilitate the implementation of shaped signaling, fast DM solutions with high throughput and low serialism are required. We propose a novel DM architecture with parallel amplitudes (PA-DM) for which m - 1 component DMs, each with a different binary output alphabet, are operated in parallel in order to generate a shaped sequence with m amplitudes. With negligible rate loss compared to a single nonbinary DM, PA-DM has a parallelization factor that grows linearly with m, and the component DMs have reduced output lengths. For such binary-output DMs, a novel constant-composition DM (CCDM) algorithm based on subset ranking (SR) is proposed. We present SR-CCDM algorithms that are serial in the minimum number of occurrences of either binary symbol for mapping, and fully parallel for demapping. For distributions that are optimized for the additive white Gaussian noise (AWGN) channel, we numerically show that PA-DM combined with SR-CCDM can reduce the number of sequential processing steps by more than an order of magnitude, while having a rate loss that is comparable to conventional nonbinary CCDM with arithmetic coding.

IEEE Transactions on Communications

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2020
201 Broadway, Cambridge, Massachusetts 02139

Parallel-Amplitude Architecture and Subset Ranking for Fast 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*—A distribution matcher (DM) maps a binary input sequence into a block of nonuniformly distributed symbols. To facilitate the implementation of shaped signaling, fast DM solutions with high throughput and low serialism are required. We propose a novel DM architecture with parallel amplitudes (PA-DM) for which $m-1$ component DMs, each with a different binary output alphabet, are operated in parallel in order to generate a shaped sequence with m amplitudes. With negligible rate loss compared to a single nonbinary DM, PA-DM has a parallelization factor that grows linearly with m, and the component DMs have reduced output lengths. For such binary-output DMs, a novel constant-composition DM (CCDM) algorithm based on subset ranking (SR) is proposed. We present SR-CCDM algorithms that are serial in the minimum number of occurrences of either binary symbol for mapping, and fully parallel for demapping. For distributions that are optimized for the additive white Gaussian noise (AWGN) channel, we numerically show that PA-DM combined with SR-CCDM can reduce the number of sequential processing steps by more than an order of magnitude, while having a rate loss that is comparable to conventional nonbinary CCDM with arithmetic coding.**

***Index Terms*—Constant Composition Distribution Matching, Subset Ranking, Probabilistic Amplitude Shaping, Coded Modulation.**

I. Introduction

Numerous techniques have been proposed to close the so-called ultimate shaping gap of 1.53 dB signal-to-noise ratio (SNR) for the additive white Gaussian noise (AWGN) channel [1]. The essence of all these constellation-shaping techniques is to mimic the capacity-achieving Gaussian distribution. The two main approaches to achieve this are geometric shaping, which describes the optimized rearrangement of the constellation points in the complex plane, and probabilistic shaping, which keeps the constellation on a fixed grid yet utilizes a non-uniform, i.e. shaped, distribution. Integrating probabilistic shaping into a coded modulation architecture is a non-straightforward task, and various methods have been proposed to this end [2]–[5].

Probabilistic amplitude shaping (PAS) as proposed in [6, Sec. IV] has recently attracted a lot of attention as a method for incorporating probabilistic shaping into bit-interleaved coded modulation (BICM) systems. The reverse concatenation principle of PAS allows to use existing binary forward error correction (FEC) without the need for demapper-decoder iterations at the receiver. PAS enables significant shaping gains and rate adaptivity for a fixed-rate FEC. It has been used in many different communication settings, such as the optical channel [7]–[10], for orthogonal frequency-division multiplexing [11, Sec. IV], Terahertz frequency wireless communication [12] and polar coded modulation [13]–[15].

The distribution matcher (DM) plays an integral role in the PAS framework as the transmitter-side processing block for mapping a sequence of uniform data bits into shaped amplitudes [16]. At the receiver, the inverse operation, demapping, is carried out. In this paper, we consider block-wise, fixed-length, invertible DMs with binary input for mapping and binary output for demapping. All finite-length DMs suffer from a rate loss that ultimately limits the throughput of a shaped coded modulation system. The rate loss can by decreased by increasing the DM block length, which has the disadvantages of long processing time (and thus latency) and high memory requirements.

In order to properly characterize and compare different DMs, we differentiate between the DM *system*, describing the general DM architecture and its properties, and the DM *method*, which relates to the actual implementation (e.g., algorithm) of the DM mapping and demapping function. A widely used DM system is based on constant-composition distribution matching (CCDM) [17, Sec. III], and the proposed algorithm to realize CCDM is arithmetic coding [17, Sec. IV].

For CCDM, each shaped output sequence has the same composition, i.e., the relative frequency of each amplitude within each block is fixed for all possible output sequences. As shown in [17, Sec. III-B], the CCDM rate loss becomes negligibly small for output lengths beyond approximately 500 symbols. Arithmetic coding (AC) was proposed in [17, Sec. IV] as an implementation of CCDM. The main drawback of this method is that it is serial in the number of input bits k for mapping and in the sequence length n of the shaped amplitudes for demapping. To the best of our knowledge, there is no constructive CCDM algorithm other than AC, and we refer to it as AC-CCDM.[1] The serial nature of AC-CCDM in combination with the long blocks required for low rate loss currently make real-time operation of CCDM highly challenging.

Recently proposed DM systems lift the constant-composition principle, thereby reducing the length that is

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.

[1] The use of lookup tables is not considered because their size and thus hardware requirements are infeasible in practice at an acceptable rate loss.

required for a certain rate loss in comparison to conventional CCDM. In [18], distribution matching via multiset partitioning is proposed. Shell mapping to index the output sequences is proposed in [19]. Both techniques are shown to give a block length reduction by approximately a factor of 5 compared to CCDM. The low-complexity DM of [20] generates two shaped output sequences for each binary input word and chooses the one with less average energy, which implicitly leads to a Gaussian-like distribution. In [21], an enumerative amplitude shaping method is proposed that is based on choosing those sequences in a trellis that have a certain maximum energy. The DM proposed in [22] compares different sequences generated by a mark ratio controller and selects the sequence that has desired properties. In [23], a prefix-free code is used to generate bits-to-codeword mappings that are variable in length, with an additional framing technique to enable fixed data-rate transmission.

Parallelization of a nonbinary-alphabet DMs can be achieved with product distribution matching [11] and bit-level distribution matching (BL-DM) [24]. These two independently proposed schemes realize a nonbinary-to-binary transformation by factorizing the nonbinary distribution of m amplitudes into $\log_2 m$ binary component distributions.[2] For each bit level, a single CCDM is used and the respective binary outputs are combined to give the desired nonbinary output sequence. In addition to parallelization by a factor $\log_2 m$ compared to a single nonbinary DM, BL-DM can have a smaller rate loss than employing a single nonbinary DM, at the expense of a limited choice of target distributions as they must be product distributions. A combination of parallel bit-level distribution matching and a multi-composition codebook was proposed in [25], making use of the improved short-length performance of multi-composition codebooks and the complexity gains of BL-DM.

In this paper, a novel distribution matcher with parallel amplitudes (PA-DM) is proposed for which several binary DMs are operated in parallel instead of a single nonbinary DM.[3] A binary CCDM is employed for each of the $m-1$ out of m shaped amplitudes, with the alphabet of every binary output subsequence comprising a specific amplitude and a different, arbitrary zero-symbol representing the absence of that amplitude. These subsequences are then sequentially combined to generate the desired sequence of shaped amplitude symbols. In PA-DM, the number of parallel DMs grows linearly with m, which results in a higher degree of parallelization than for BL-DM, which has a DM per bit level and thus a number of parallel DMs that is logarithmic in m. Therefore, PA-DM has reduced algorithmic serialism, which may enable lower latency and higher throughput for resource constrained hardware implementations, and thus allows for faster distribution matching than previously introduced DM architectures.

We further propose a method for CCDM mapping and demapping via subset ranking (SR) as an alternative to AC-CCDM for binary alphabets. The proposed method is closely related to the enumerative techniques used by Schalkwijk [26] and Cover [27]. In this paper, we focus on SR implementations that reduce the number of sequential operations as much as possible. In contrast to AC-CCDM, CCDM mapping with the SR algorithm, which we refer to as SR-CCDM, is serial in the smallest number of occurrences of either binary symbol in the output sequence, and demapping via SR-CCDM is fully parallel. We show that this serialism constitutes a significant improvement over AC-CCDM, which is sequential in the length of its input. A further discussion of computational complexity is omitted in this manuscript as the actual speed of the investigated CCDM algorithms, in particular in an application specific integrated circuit (ASIC), depends greatly on hardware specifications and implementation details that are out of the scope of this paper. For a distribution with $m = 8$ shaped amplitudes that is optimized for the additive white Gaussian noise (AWGN) channel, combining PA-DM and SR-CCDM is numerically shown to give a reduction in serialism of more than an order of magnitude with similar performance as conventional CCDM.

II. Preliminaries

A. Notation

The realizations $a_i,\ i \in \{1,\ldots,m\}$ of a random variable A are drawn from an alphabet $\mathcal{A}$ according to a probability mass function (PMF) P_A. Vectors of length n are denoted as $\mathbf{x}^{(n)} = [x_1,\ldots,x_n]$. If the elements of such a vector are binary, e.g., a and b, we can also write $\{a,b\}^n$. Sets are denoted as calligraphic letters, e.g., $\mathcal{N} = \{1,\ldots,n\}$.

B. Probabilistic Amplitude Shaping (PAS)

We briefly outline PAS in the following and refer to [6, Sec. IV] for details. A block diagram of PAS is shown in Fig. 1, where we assume for simple representation that the DM output and FEC input lengths are compatible. When the DM is shorter than the FEC, several DM sequences must be combined within each FEC block.

The binary data word to be transmitted is split into the DM input sequence $\mathbf{b}^{(k)}$ and uniform data bits. The DM mapping f_{DM} transforms $\mathbf{b}^{(k)}$ into a sequence $\mathbf{x}^{(n)}$ comprising the shaped amplitudes $\{a_1,,\ldots,a_m\}$. The sequence is given a binary label and input into a systematic FEC encoder. The information bits of the FEC output correspond to the shaped amplitudes, while the parity bits combined with the uniform data bits represent the sign bits of the constellation symbols. After modulation, typically with two-dimensional quadrature amplitude modulation (QAM), the shaped symbols are transmitted over a channel such as the AWGN channel. The demapper computes log-likelihood ratios that are used for FEC decoding or estimation of the achievable information rate (AIR) for bit-metric decoding (BMD) [6, Sec. VI]. When decoding is successful, which is assumed herein, the bits that correspond to shaped amplitude are transformed into the sequence of shaped amplitude bits $\mathbf{x}^{(n)}$. After DM demapping f_{DM}^{-1}, the initial word $\mathbf{b}^{(k)}$ is recovered.

[2] In the following, we jointly refer to these two proposals as BL-DM since they carry out the same task.

[3] All considered DM mappers have binary input, so the distinction between binary and nonbinary alphabets relates only to its output.

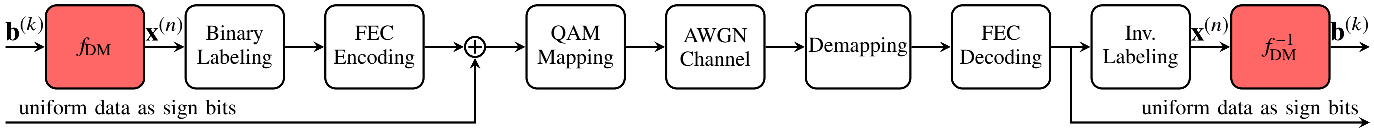

Fig. 1. Block diagram of PAS. The plus node combines the shaped amplitude bits (which remain unchanged by the systematic FEC encoder) and the sign bits, which are the parity bits and possible some uniform input bits. This paper covers DM systems and methods (red boxes).

C. Constant-Composition Distribution Matching (CCDM)

1) Principle: We consider distribution matchers that map a binary input $\mathbf{b}^{(k)} \in \{0,1\}^k$ to a shaped output sequence $\mathbf{x}^{(n)} = [x_1, \ldots, x_n]$ of length n. The DM mapping function establishes an invertible mapping $f_{\mathrm{DM}} : \mathbf{b}^{(k)} \to \mathbf{x}^{(n)}$, and the inverse operation (demapping) is $f_{\mathrm{DM}}^{-1} : \mathbf{x}^{(n)} \to \mathbf{b}^{(k)}$.

The m different output amplitudes that can occur in $\mathbf{x}^{(n)}$ are taken from the alphabet $\mathcal{A} = \{a_1, \ldots, a_m\}$. The CCDM output sequence is said to have the composition $C = \{n_1, \ldots, n_m\}$ with n_i denoting the number of times the i^{th} amplitude a_i occurs, i.e.,

$$n_i = \left|\{j : x_j = a_i\}\right| \tag{1}$$

with $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. This implies that the relative frequency P_A of a_i is $P_A(a_i) = \frac{n_i}{n}$, which is referred to as type [28, Sec. II]. Throughout this paper, the type of all CCDM output sequences is fixed, i.e., all CCDM outputs have the same composition.

2) Input Length and Rate Loss: The number of input bits k of a DM depends on the number of different output sequences, which is given by the multinomial coefficient

$$M(C) = \binom{n}{n_1, n_2, \ldots, n_m} = \frac{\left(\sum\limits_{i=1}^{m} n_i\right)!}{\prod\limits_{i=1}^{m} (n_i!)}. \tag{2}$$

It is a natural choice to consider only DMs with an integer number of input bits. The input length k in bits is thus

$$k = \log_2 \lfloor M(C) \rfloor_2, \tag{3}$$

where $\lfloor \cdot \rfloor_2$ denotes rounding down to the closest power of two. The rate loss of a DM is then defined as

$$R_{\mathrm{loss}} = \mathbb{H}(A) - \frac{k}{n}, \tag{4}$$

where $\mathbb{H}(A)$ is the entropy of the amplitudes A with the quantized distribution P_A. Such a quantization is necessary in many finite-length cases to achieve an integer-valued composition. The quantization criterion used in this paper is the minimization of Kullback-Leibler divergence between the initial unquantized PMF and the quantized distribution P_A [29, Sec. IV].

3) Arithmetic Coding as CCDM Method: CCDM mapping and demapping can be carried with arithmetic coding (AC) [17, Sec. IV]. More details on AC including a discussion of the algorithm implementation can be found in [30, Ch. 4]. The underlying principle is drawing without replacement where in every AC step, interval boundaries are computed based on those elements of the composition that have not yet been used in the output sequence. Since the size of these intervals depends on previous steps, AC is an inherently sequential algorithm that, in the worst case, is serial in the number of input elements to choose from, which is k for mapping and n for demapping.[4] Within each serially executed AC operation, the number and complexity of computations to be performed varies as it depends on the specific interval boundaries.

III. Distribution Matching with Parallel Amplitude Levels

In the following, a distribution matching transformation from a single DM with nonbinary output to parallel DMs that each have a binary output alphabet corresponding to a shaped amplitude. Since the amplitudes are effectively processed in parallel, we refer to this scheme as parallel amplitude (PA)-DM.

A. Preliminaries: Binomial and Multinomial Coefficients

To explain the proposed approach, it is insightful to express the multinomial coefficient $M(C)$ of a composition $C = \{n_1, \ldots, n_m\}$ with length $n = \sum_{i=1}^{m} n_i$ (see (2)) as a product of binomial coefficients (BCs),

$$M(C) = \binom{n}{n_1, n_2, \ldots, n_m} \tag{5}$$

$$= \underbrace{\binom{n}{n_1}}_{\mathrm{BC}_1} \cdot \underbrace{\binom{n-n_1}{n_2}}_{\mathrm{BC}_2} \cdot \ldots \cdot \underbrace{\binom{n-n_1-\ldots-n_{m-2}}{n_{m-1}}}_{\mathrm{BC}_{m-1}} \cdot \underbrace{\binom{n_m}{n_m}}_{\mathrm{BC}_m} \tag{6}$$

$$= \prod_{i=1}^{m} \underbrace{\binom{n - \sum\limits_{j=0}^{i-1} n_j}{n_i}}_{\mathrm{BC}_i}, \tag{7}$$

where we define $n_0 = 0$ in (7) for notational convenience. We recall that the first factor of (7) represents the number of ways to choose n_1 out of n elements (disregarding their order), the second one the ways to choose n_2 elements out of the remaining $n - n_1$ elements and so forth. Varying the ordering of the binomial expansion can give different component binomial coefficients (see also Sec. III-D), but their product is always equal to $M(C)$ and the last factor BC_m in (7) is equal to 1. In the following, we use the product of binomial coefficients of

[4] Note that there are cases where the AC algorithm can be terminated early because the remainder of the output sequence follows with probability 1. We neglect these cases and discuss only worst-case serialism which occurs when all AC steps must be carried out.

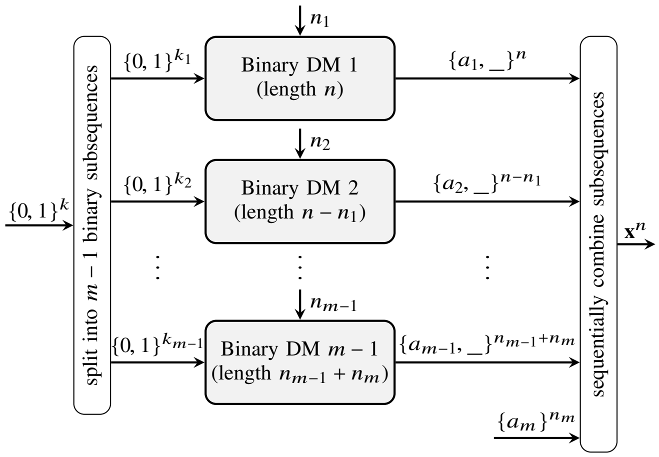

Fig. 2. Mapping structure of distribution matching with parallel amplitudes (PA-DM). $m-1$ parallel binary DMs of varying lengths are employed and their outputs are sequentially combined to achieve the nonbinary shaped sequence $\mathbf{x}^{(n)}$.

(7) to transform a nonbinary DM into binary component DMs with parallel amplitudes.

B. PA-DM Method

In the PA-DM architecture, the first $m-1$ BCs in (7) each correspond to a DM instance that maps a binary input to a sequence whose alphabet comprises the considered amplitude and another symbol which we denote as _ and which represents the absence of that amplitude. Figure 2 shows a block diagram of the mapping with PA-DM. First, the binary input of length k is split into $m-1$ substrings that are each input into a binary-input binary-output DM. The mapping operation of the first DM is to place n_1 occurrences of the first amplitude a_1 in n positions, with the output sequence $\{a_1, \_\}^n$ being the result of mapping the input of length $k_1 = \log_2\lfloor \mathrm{BC}_1 \rfloor_2$. The second DM maps $k_2 = \log_2\lfloor \mathrm{BC}_2 \rfloor_2$ bits by placing the amplitude a_2 n_2-times in the remaining (unused) $n - n_1$ positions. By repeating this for all amplitudes up to a_{m-1}, $\mathbf{x}^{(n)}$ can be gradually constructed. Finally, the remaining n_m positions of $\mathbf{x}^{(n)}$ that are not yet occupied are filled with a_m. This gives the desired nonbinary output sequence $\mathbf{x}^{(n)}$.[5] An example of PA-DM mapping is given below in Example 1. Demapping for PA-DM is achieved by performing the above steps in inverse order, i.e., by first decomposing the shaped sequence into binary subsequences, applying inverse distribution matching, and combining the outputs to generate the initially transmitted $\mathbf{b}^{(k)}$. The method for the binary component DMs can be either conventional AC-CCDM or the subset-ranking method described in Sec. IV.

An important benefit of PA-DM is that it allows the DM mapping to be split in parallel instances, thereby enabling high-throughput DM implementations. The number of parallel DMs (and thus the parallelization factor compared to a single nonbinary DM) is $m-1$ and hence grows linearly with the one-dimensional modulation order, whereas the number of parallel DMs is logarithmic in m for BL-DM. The parallelization factors compared to a nonbinary DM are summarized in Table I for two-sided amplitude shift keying (ASK), i.e., including the PAS sign bit. Since the amplitudes are binary for 16QAM, a parallelization factor of 1 describes that neither PA-DM nor BL-DM give any benefit. For QAM of order 64 and more, however, PA-DM employs significantly more DMs than BL-DM. An additional advantage of PA-DM could be that the binary component DMs have decreasing output length and thus potentially improved computational complexity, while they are of identical length for BL-DM.

TABLE I
PARALLELIZATION FACTORS OF PA-DM AND BL-DM COMPARED TO NONBINARY DM FOR ASK FORMATS WITH m SHAPED AMPLITUDES

Format	m	PA-DM	BL-DM
4ASK (16QAM)	2	1	1
8ASK (64QAM)	4	3	2
16ASK (256QAM)	8	7	3
32ASK (1024QAM)	16	15	4

C. Rate Loss of PA-DM

As each factor of (7) corresponds to a DM that maps an integer $k_i = \log_2\lfloor \mathrm{BC}_i \rfloor_2$ bits, the aggregate number of input bits of all DMs in the PA-DM architecture is

$$k = \sum_{i=1}^{m} k_i = \sum_{i=1}^{m} \log_2\lfloor \mathrm{BC}_i \rfloor_2. \tag{8}$$

For a nonbinary DM, in contrast, we have $k = \log_2\lfloor M(C) \rfloor_2$, see (3). Depending on the specific composition, rounding down each individual BC to the largest power of 2 can yield no additional rate loss, or can also result in a small loss up to $m-2$ bits compared to a single nonbinary DM.

Example 1 (Mapping Operation for PA-DM): Consider the composition $C = \{4,3,2,1\}$ for the amplitudes $\{\alpha,\beta,\gamma,\delta\}$ and an output sequence $\mathbf{x}^{(n)}$ with $n = 10$ that is supposed to have this composition. By (2), we have $M(C) = 12600$ permutations, and thus $\log_2\lfloor 12600 \rfloor_2 = 13$ input bits that can be mapped with a conventional nonbinary DM. By splitting the multinomial coefficient into a product of binomials according to (7), we have $M(C) = \binom{10}{4} \cdot \binom{6}{3} \cdot \binom{3}{2} \cdot \binom{1}{1} = 210 \cdot 20 \cdot 3 \cdot 1 = 12600$, which gives $\log_2\lfloor 210 \rfloor_2 + \log_2\lfloor 20 \rfloor_2 + \log_2\lfloor 3 \rfloor_2 + \log_2\lfloor 1 \rfloor_2 = 7 + 4 + 1 + 0 = 12$ bits at the PA-DM input. Thus, PA-DM has an additional rate loss of 1 bit compared to a nonbinary DM. Now suppose that the 12-bit data word to be mapped is $\mathbf{b}^{(k)} = [011101000101]$, which is split into subsequences of lengths $k_1 = 7$, $k_2 = 4$, and $k_3 = 1$. Depending on the mapping algorithm (see Sec. IV), the $m-1 = 3$ DM mapping outputs are as follows, with _ denoting the absence of an amplitude:

- $f_{\mathrm{DM}_1} : [0111010] \rightarrow [\alpha, \_, \_, \alpha, \_, \_, \alpha, \alpha, \_, \_]$
- $f_{\mathrm{DM}_2} : [0010] \rightarrow [\beta, \beta, \_, \_, \beta, \_]$
- $f_{\mathrm{DM}_3} : [1] \rightarrow [\gamma, \_, \gamma]$

[5] We note that this sequential combination can also be done in a tree-like fashion by repeatedly combining two amplitudes at once, which reduces the number of sequential operations.

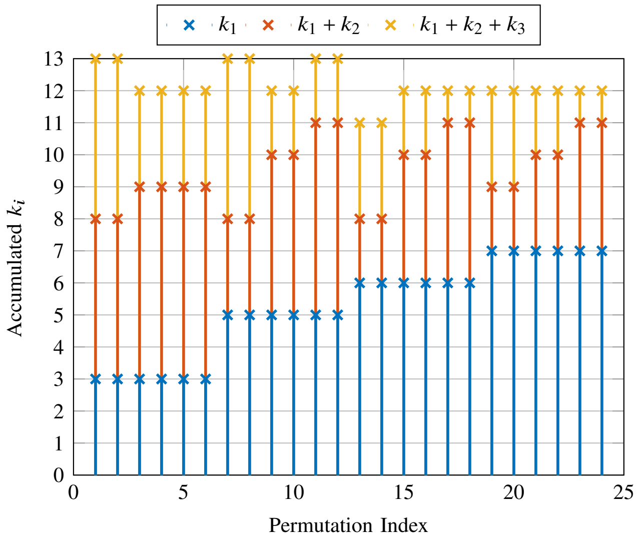

Fig. 3. Accumulated PA-DM input length over permutation index of lexicographically sorted composition $C = \{4, 3, 2, 1\}$. There are six permutations and thus ordering of DMs in the PA-DM system that achieve the maximum input length of 13 bits.

These three output sequences are then combined sequentially. The 6 free positions of the first DM output are filled with the output of the second DM, giving the temporary sequence $[\alpha, \beta, \beta, \alpha, \_, \_, \alpha, \alpha, \beta, \_]$. The remaining 3 positions are used by the third DM and we have $[\alpha, \beta, \beta, \alpha, \gamma, \_, \alpha, \alpha, \beta, \gamma]$. The remaining open position is filled with $n_m = 1$ occurrence of δ, giving the final CCDM output sequence $\mathbf{x}^{(n)} = [\alpha, \beta, \beta, \alpha, \gamma, \delta, \alpha, \alpha, \beta, \gamma]$. △

D. Ordering of Binomial Coefficients

As previously noted in the context of (7), the product of binomial coefficients is always equal to the multinomial coefficient, but the individual factors can vary. Hence, depending on the ordering of the BCs, each component binary DM can take a different number of bits $k_i = \log_2\lfloor \mathrm{BC}_i \rfloor_2$ at its input. Due to the nonlinearity of the flooring of each BC (see (8)), the order of the BCs has an impact on k and thus on the DM rate loss. By a simple one-time exhaustive search over all possible orderings (of which there are at most $m!$), the rate loss of PA-DM can be minimized, as illustrated in the following Example 2. A detailed rate loss comparison between PA-DM, BL-DM, and a nonbinary DM for AWGN-optimized distributions can be found in Sec. V-A.

Example 2 (Optimize BC Ordering to Minimize Rate Loss): Given the composition $C = \{4, 3, 2, 1\}$ for the amplitudes $\{\alpha, \beta, \gamma, \delta\}$ as described in Example 1, there are $m! = 24$ different orderings of the binomial coefficients. In Fig. 3, the accumulated number of input bits k_i per DM is shown over the index of the BC orderings, which are sorted lexicographically. This means that permutation index 1 corresponds to $C = \{1, 2, 3, 4\}$ and index 24 is $C = \{4, 3, 2, 1\}$. We observe from Fig. 3 that there are 6 composition orderings that allow to address $k = 13$ bits. One of these orderings is $C = \{1, 2, 3, 4\}$, for which we have $M(C) = \binom{10}{1} \cdot \binom{9}{2} \cdot \binom{7}{3} \cdot \binom{4}{4} = 10 \cdot 36 \cdot 35 \cdot 1 = 12600$. In this case $\log_2\lfloor 10\rfloor_2 + \log_2\lfloor 36\rfloor_2 + \log_2\lfloor 35\rfloor_2 = 13$ bits can be mapped by PA-DM, resulting in zero additional rate loss in comparison to a single nonbinary DM. △

IV. Distribution Matching via Subset Ranking

This section outlines binary-output CCDM mapping and binary-input CCDM demapping methods with low serialism. The key parameters are the output length n in bits, the DM input length defined as $k = \log_2\left\lfloor \binom{n}{w} \right\rfloor_2$, and the weight w denoting the numbers of occurrences of a symbol a in the binary sequence $\mathbf{x}^{(n)} \in \{a, b\}^n$, i.e.,

$$w = |\{i \in \{1, \ldots, n\} : x_i = a\}|. \tag{9}$$

Since $\mathbf{x}^{(n)}$ is binary, we have $n - w$ occurrences of b.

The proposed DM method is based on techniques for the ranking of subsets that are drawn from a set, which is a well-known problem in enumerative combinatorics (e.g. [31, Sec. 2.4]). A similar approach has been applied by Schalkwijk [26] and Cover [27] for source coding, and recently been used in enumerative sphere shaping [32]–[34]. We focus on highly parallel algorithms for subset ranking, with an application to CCDM, noting that the proposed approach can be used for any binary enumerative coding technique.

In the following, we review the preliminaries for subset ranking (SR) before linking it to distribution matching. Algorithms are presented, their application is discussed, and compared to a conventional AC-CCDM.

A. Preliminaries and Definitions for Subset Ranking

Let $\mathcal{N} = \{1, \ldots, n\}$ with n being the DM output length as introduced earlier in this manuscript. We further define the set $\mathcal{S}$ to consist of the $\binom{n}{w}$ w-element subsets of the n-set $\mathcal{N}$. The w-element subset $\mathcal{T} \subseteq \mathcal{N}$ contains the integer elements $\{t_1, \ldots, t_w\}$ and thus constitutes the elements of $\mathcal{S}$.

We are interested in ordering the subsets $\mathcal{S}$, for which a natural choice is lexicographic (lex) ordering. To impose this order on $\mathcal{S}$, we first introduce the list representation $\overrightarrow{\mathcal{T}}$ of $\mathcal{T}$ as

$$\overrightarrow{\mathcal{T}} = [t_1, t_2, \ldots, t_w], \tag{10}$$

where the elements of $\overrightarrow{\mathcal{T}}$ are sorted in ascending order,

$$t_1 \leq t_2 \leq \ldots \leq t_w, \tag{11}$$

as indicated by the arrow direction of $\overrightarrow{\mathcal{T}}$. The lex ordering of the subsets $\mathcal{S}$ is obtained by sorting the sequences $\overrightarrow{\mathcal{T}}$ in a dictionary-style fashion, i.e., by applying ascending order to the component integers $[t_1, t_2, \ldots, t_w]$.

Another common ordering besides lex is colexicographic (colex). In analogy to (10), we define the colex list representation of $\mathcal{T}$ as

$$\overleftarrow{\mathcal{T}} = [t_1, t_2, \ldots, t_w]\,. \tag{12}$$

with

$$t_1 \geq t_2 \geq \ldots \geq t_w\,. \tag{13}$$

The colex ordering in $\mathcal{S}$ is achieved by applying lex ordering to all sequences $\overleftarrow{\mathcal{T}}$.

Given a specific ordering, the *ranking* of a w-element subset $\mathcal{T}$ determines its position (or rank) within all $\binom{n}{w}$ subsets $\mathcal{S}$. Thus, the rank of a specific subset for a given ordering is the number of precursors that this subset has. For the example of lex ordering, a list $\overrightarrow{\mathcal{T}_a} = \left[t_{a,1}, \ldots, t_{a,w}\right]$ is a precursor to $\overrightarrow{\mathcal{T}_b} = \left[t_{b,1}, \ldots, t_{b,w}\right]$ if the smallest (leftmost) position $j \in \{1, \ldots, w\}$ where we have $t_{a,j} > t_{b,j}$ is smaller than the leftmost position $k \in \{1, \ldots, w\}$ where $t_{a,k} < t_{b,k}$. Formally, a ranking is a bijective function from $\mathcal{S}$ to the integer rank r, i.e.,

$$f_{\text{rank}} : \mathcal{S} \rightarrow r, \qquad r \in \{0, \ldots, \binom{n}{w} - 1\}. \tag{14}$$

The inverse operation is called *unranking* and defined as

$$f_{\text{unrank}} : r \rightarrow \mathcal{S}, \qquad r \in \{0, \ldots, \binom{n}{w} - 1\}. \tag{15}$$

In the following, we introduce a specific notation for the ranking and unranking functions depending on the ordering, which is indicated by the subscript lex or colex. The ranking of a particular subset $\mathcal{T}$ with lex ordering is denoted as

$$\text{rank}_{\text{lex}}\left(\mathcal{T}\right) = r_{\text{lex}}, \tag{16}$$

and we write for colex ordering

$$\text{rank}_{\text{colex}}\left(\mathcal{T}\right) = r_{\text{colex}}. \tag{17}$$

In analogy, the unranking functions are

$$\text{unrank}_{\text{lex}}\left(r_{\text{lex}}\right) = \mathcal{T} \tag{18}$$

for lex and

$$\text{unrank}_{\text{colex}}\left(r_{\text{colex}}\right) = \mathcal{T}, \tag{19}$$

for colex. Note that lex and colex ranking are linked with the simple relation [31, Theorem 2.4]

$$\text{rank}_{\text{lex}}\left(\mathcal{T}\right) + \text{rank}_{\text{colex}}\left(\mathcal{T}'\right) = \binom{n}{w} - 1, \tag{20}$$

where

$$\mathcal{T}' = \{n + 1 - t_i : t_i \in \mathcal{T}\}. \tag{21}$$

This relationship can be useful if ranking or unranking algorithms of a certain ordering has computational advantages.

Example 3 (Ranking for Lex and Colex Ordering): Consider $w = 2$ and the $n = 5$-set $\mathcal{N} = \{1, \ldots, 5\}$. There are $\binom{5}{2} = 10$ subsets $\mathcal{T}$ in the set $\mathcal{S}$. The subsets, their list representation and the corresponding ranking for lex and colex ordering are listed in Table II. △

TABLE II

RANKS FOR LEX (LEFT) AND COLEX (RIGHT) ORDERING FOR $n = 5$ AND $w = 2$ AS PER EXAMPLE 3.

$\mathcal{T}$	$\overrightarrow{\mathcal{T}}$	r_{lex}
[1,2]	[1,2]	0
[1,3]	[1,3]	1
[1,4]	[1,4]	2
[1,5]	[1,5]	3
[2,3]	[2,3]	4
[2,4]	[2,4]	5
[2,5]	[2,5]	6
[3,4]	[3,4]	7
[3,5]	[3,5]	8
[4,5]	[4,5]	9

$\mathcal{T}$	$\overleftarrow{\mathcal{T}}$	r_{colex}
[1,2]	[2,1]	0
[1,3]	[3,1]	1
[2,3]	[3,2]	2
[1,4]	[4,1]	3
[2,4]	[4,2]	4
[3,4]	[4,3]	5
[1,5]	[5,1]	6
[2,5]	[5,2]	7
[3,5]	[5,3]	8
[4,5]	[5,4]	9

B. Binary Sequence as a Constant-Order Subset

A binary sequence $\{a, b\}^n$ with alphabet $\{a, b\}$ can be described by a subset of the integers $\{1, \ldots, n\}$ that denotes the positions of either symbol, for example a, in that sequence. The complementary set then gives the locations of the other symbol, here b. Applying a constant order (such as ascending) to this integer subset gives an equivalent description of the sequence $\{a, b\}^n$. This correspondence is used in the next section to propose a CCDM method via subset ranking.

Example 4 (Binary Sequence as Integer Subset): Suppose we have the binary sequence $\{aabbabaa\}$ with $n = 8$. The integer subset describing the positions of symbol a in ascending order is $\{1, 2, 5, 7, 8\}$. The complementary subset for b is thus $\{3, 4, 6\}$. △

C. Subset Unranking and Ranking as DM Mapping and Demapping

We now link the above outlined subset ranking to the DM terminology. With the ranking and unranking functions (16) to (19), a bijective mapping between the subset $\mathcal{T}$ and its rank is established. The rank (in binary representation) of $\mathcal{T}$ corresponds to the uniform binary sequence $\mathbf{b}^{(k)}$ that is the input of the binary-alphabet DM mapper. The w-element subset $\mathcal{T}$ that corresponds to this rank describes which positions of the DM mapper output sequence carry one of the two binary output symbols, see Sec. IV-B.[6] The DM mapping operation from uniform data word $\mathbf{b}^{(k)}$ to shaped sequence $\mathbf{x}^{(n)}$ can thus be considered an unranking problem. In analogy, the DM demapper carries out a ranking operation: given a shaped sequence that corresponds to the subset $\mathcal{T}$, the rank is to be determined.

Example 5 (DM Mapping and Demapping with Lex Subset Ranking): Consider a binary DM with $n = 10$ and the desired binary distribution $P_A(0) = 0.6$, $P_A(1) = 0.4$. We have $w = 4$ and thus the DM input length $k = \left\lfloor \binom{10}{4} \right\rfloor_2 = 7$ bits. Suppose the binary word to be mapped is $\mathbf{b}^{(k)} = [1110101]$, which is $r_{\text{lex}} = 117$ in denary representation. With an unranking algorithm of Sec. IV-D, the subset $\mathcal{T}$ in lex ordering that has $r_{\text{lex}} = 117$ is determined as $\overrightarrow{\mathcal{T}} = [2, 4, 8, 9]$.[7] From this, the DM output sequence of length 10 is determined, as follows in Sec. IV-B. The sequence elements that have indices $[2, 4, 8, 9]$ are set to '1', i.e., we have $\mathbf{x}^{(n)} = [0101000110]$. At the demapper, $\overrightarrow{\mathcal{T}}$ is determined from the sequence $\mathbf{x}^{(n)}$,

[6] Which symbol is represented by w is a somewhat arbitrary choice. The same SR functionality is achieved when the w-element subset $\mathcal{T}$ represents the positions of the complementary binary symbol.

[7] Note that colex ordering is also feasible. For $r_{\text{colex}} = 117$ we would get $\overleftarrow{\mathcal{T}} = [9, 8, 6, 3]$.

and a ranking of this subset gives the initial data word $\mathbf{b}^{(k)} = [1110101]$. △

D. Ranking and Unranking Algorithms

In the following, we present pseudo-code algorithms for subset ranking and unranking [31, Sec. 2.4] and discuss their serialism. Ranking for the subset $\mathcal{T}$ in lex and colex ordering is presented in Algorithms 1 and 2, respectively. We note that the inner nested for-loop of Algorithm 1 (line 6) can be easily replaced with parallel vector operations, which makes the lex ranking algorithm serial in w. The colex ranking does not have any loops and is thus of great interest for low-latency high-throughput DM demapping.

The unranking algorithms for lex and colex ordering are given as Algorithms 3 and 4, respectively. The inner nested loops in Algorithm 3 (line 4) and Algorithm 4 (line 4) can again be executed in parallel. Furthermore, if $w > \frac{n}{2}$, the unranking algorithm can be set to determine the positions of the complementary binary symbol, thereby limiting the required number of loop iterations in the unranking Algorithms 3 and 4 to $\min(w, n-w)$.

Algorithm 1 Lex ranking function $\mathrm{rank}_{\mathrm{lex}}(\cdot)$ of (16)

```
Require: $\overrightarrow{\mathcal{T}}$, $w$        ▷ Ordered subset, weight of binary seq.
 1: function LexRank($\overrightarrow{\mathcal{T}}$, $w$)
 2:     $r_{\mathrm{lex}} \leftarrow 0$
 3:     $t_0 \leftarrow 0$                               ▷ For notational convenience
 4:     for $i$ from 1 to $w$ do
 5:         if $t_{i-1} + 1 \leq t_i - 1$ then
 6:             for $j$ from $t_{i-1} + 1$ to $t_i - 1$ do
 7:                 $r_{\mathrm{lex}} \leftarrow r_{\mathrm{lex}} + \binom{n-j}{w-i}$
 8:             end for
 9:         end if
10:     end for
11:     return $r_{\mathrm{lex}}$                              ▷ See (16)
12: end function
```

Algorithm 2 Colex ranking function $\mathrm{rank}_{\mathrm{colex}}(\cdot)$ of (17)

```
Require: $\overleftarrow{\mathcal{T}}$, $w$        ▷ Ordered subset, weight of binary seq.
 1: function ColexRank($\overleftarrow{\mathcal{T}}$, $w$)
 2:     $\mathbf{j} \leftarrow [1, 2, \ldots, w]$            ▷ Integer list from 1 to $w$
 3:     $r_{\mathrm{colex}} \leftarrow \sum_{i=1}^{w} \binom{t_i - 1}{w + 1 - \mathbf{j}_i}$
 4:     return $r_{\mathrm{colex}}$                            ▷ See (17)
 5: end function
```

Algorithm 3 Lex unranking function $\mathrm{unrank}_{\mathrm{lex}}(\cdot)$ of (18)

```
Require: $n$,$w$,$r_{\mathrm{lex}}$     ▷ DM output length, weight of binary
    seq., rank
 1: function LexUnrank($n$,$w$,$r_{\mathrm{lex}}$)
 2:     $j \leftarrow 1$
 3:     for $i$ from 1 to $w$ do
 4:         while $\binom{n-j}{w-i} \leq r_{\mathrm{lex}}$ do
 5:             $r_{\mathrm{lex}} \leftarrow r_{\mathrm{lex}} - \binom{n-j}{w-i}$
 6:             $j \leftarrow j + 1$
 7:         end while
 8:         $t_i \leftarrow j$
 9:         $j \leftarrow j + 1$
10:     end for
11:     return $\overrightarrow{\mathcal{T}} = [t_1, t_2, \ldots, t_w]$       ▷ See (10)
12: end function
```

Algorithm 4 Colex unranking function $\mathrm{unrank}_{\mathrm{colex}}(\cdot)$ of (19)

```
Require: $n$,$w$,$r_{\mathrm{colex}}$     ▷ DM output length, weight of binary
    seq., rank
 1: function ColexUnrank($n$,$w$,$r_{\mathrm{colex}}$)
 2:     $j \leftarrow n$
 3:     for $i$ from 1 to $w$ do
 4:         while $\binom{j}{w+1-i} > r_{\mathrm{colex}}$ do
 5:             $j \leftarrow j - 1$
 6:         end while
 7:         $t_i \leftarrow j + 1$
 8:         $r_{\mathrm{colex}} \leftarrow r_{\mathrm{colex}} - \binom{j}{w+1-i}$
 9:     end for
10:     return $\overleftarrow{\mathcal{T}} = [t_1, t_2, \ldots, t_w]$       ▷ See (12)
11: end function
```

E. Comments on Computational Complexity

We observe from the above algorithms that an integral part of ranking and unranking is to compute binomial coefficients. For the considered application as DM mapping and demapping functions, it is important that the computation is exact since an inaccurate rank calculation, for instance due to rounding, could lead to the DM introducing a transmission error. Thus, integer arithmetic should be employed rather than relying on typical floating-point precision. We further note that the values of the binomial coefficients can be huge for typical DM lengths. For instance, for a short binary CCDM with $n = 100$, binomial coefficients must be computed that exceed the maximum value of an unsigned 64-bit integer.

A method of computing binomial coefficients that could be particularly suitable for such large numbers is by prime factorization of $n!$, where n is integer. We first note that only prime numbers $p \leq n$ appear in the factorization of n. The number of times that $n!$ is divisible by the prime p, which we denote as $d_p(n!)$, is defined as

$$d_p(n!) = \sum_{i=1}^{\lfloor \log_p n \rfloor} \left\lfloor \frac{n}{p^i} \right\rfloor. \tag{22}$$

This expression is known as Legendre's theorem [35, Sec. 2.6]. With this relation, the factorial can be expressed as

$$n! = \prod_{p=2}^{n} p^{d_p(n!)}, \tag{23}$$

where p is prime and the product runs over prime numbers only, i.e., $p \in \{2, 3, 5, 7, \ldots\}$.

The above definition of a factorial can be applied to calculating a binomial coefficient $\binom{n}{w}$. With the concept of prime factorization, we have

$$\binom{n}{w} = \frac{n!}{(n-w)!\cdot w!} = \frac{\prod_{p=2}^{n} p^{d_p(n!)}}{\left(\prod_{p=2}^{n-w} p^{d_p((n-w)!)}\right)\cdot\left(\prod_{p=2}^{w} p^{d_p(w!)}\right)}, \tag{24}$$

with the products again over primes only. The computations for (24) can be further simplified by excluding those elements in the numerator and denominator that will eventually cancel out. The definition (24) can be beneficial because the numbers in intermediate steps of computing the binomial coefficient are relatively small; neither the bases nor exponents of (24), i.e., the primes p and $d_p(n!)$ as per (22), exceed n. Also, the computation can partly be implemented with bit shifts and additions.

Example 6: We wish to compute 21!. The relevant primes are $p = \{2, 3, 5, 7, 11, 13, 17, 19\}$. With the exponents computed as per (22), we have $21! = 2^{18} \cdot 3^9 \cdot 5^4 \cdot 7^3 \cdot 11^1 \cdot 13^1 \cdot 17^1 \cdot 19^1 = 51090942171709440000$. In particular, the multiplication of the already huge number $3^9 \cdot 5^4 \cdot 7^3 \cdot 11^1 \cdot 13^1 \cdot 17^1 \cdot 19^1$ with 2^{18} can be performed efficiently with 18 bit shifts. △

V. Numerical Results

In the following, we compare the finite-length rate loss of the PA-DM of Sec. III to a nonbinary DM and the BL-DM system of [11], [24]. The reduction in serialism from the subset-ranking (SR) CCDM technique of Sec. IV compared to CCDM via arithmetic-coding (AC), denoted as AC-CCDM, is analyzed in Sec. V-B.

A. Rate Loss Comparison

Numerical simulations over the AWGN channel are performed to compare the performance of PA-DM to a nonbinary (NB) DM and BL-DM for CCDM-based shaping. The figure of merit is the achievable information rate (AIR) for complex QAM signaling and bit-metric decoding [6, Sec. VI] minus the finite-length rate loss of the considered DM system, which gives an AIR for the finite-length DM, see [18, Appendix]. The AIRs for 64QAM as a function of the SNR of the AWGN channel are shown in Fig. 4 for PA-DM (dotted), BL-DM (dashed), and conventional nonbinary DM (solid). The channel capacity $\log_2(1 + \text{SNR})$ and the asymptotic AIR for infinite-length DM (i.e., with zero rate loss) and for uniform signaling are included for reference. The targeted PMF is the optimal Maxwell-Boltzmann PMF [4] at each SNR, quantized at each block length n as to minimize Kullback-Leibler divergence [29, Sec. IV]. We observe from Fig. 4 that for short lengths such as $n = 50$, BL-DM has improved performance over NB-DM and PA-DM. The reason for this is that the sum of rate losses of the individual BL-DM instances is smaller than the total rate loss of the other schemes. Note, however, that this length regime is of limited interested since the AIR is smaller than with uniform 64QAM. The performance improvement of BL-DM over the other DM systems decreases with increasing n. For $n = 500$ symbols, all three investigated systems have nearly identical performance. We further note that the rate loss of PA-DM is smaller than 0.05 bits/2D-sym compared a single nonbinary DM for all considered output lengths and SNRs.

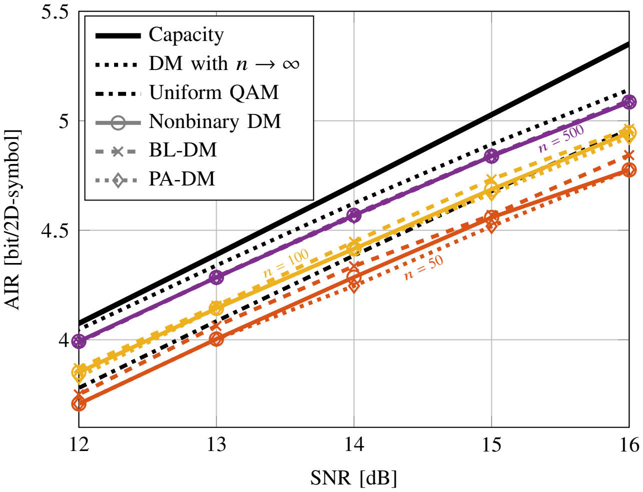

Fig. 4. Achievable information rate (AIR) for bit metric decoding and finite-length DM systems with $n = \{50, 100, 500\}$ (colored lines with markers) versus the SNR in dB of the AWGN channel for 64QAM ($m = 4$). The channel capacity and the AIRs for an infinite-length DM and for uniform signaling are included as references (black lines).

TABLE III
Comparison of PA-DM, BL-DM, and nonbinary (NB) DM, all for $n = 100$ and 64QAM ($m = 4$) at 13 dB SNR

	PA-DM	BL-DM	NB-DM
Number of DMs	$m - 1 = 3$	$\log_2 m = 2$	1
Compositions	(16, 32, 6, 46)	(78, 22) and (61, 39)	(46, 32, 16, 6)
(n, k, w) per DM	(100, 60, 16) (84, 77, 32) (52, 24, 6)	(100, 72, 22) (100, 92, 39)	(100, 161, –)
Total R_{loss}	0.1	0.09	0.1

In the following, we perform a detailed analysis of the DM systems for 64QAM, $n = 100$ and 13 dB SNR. The results are listed in Table III. First and foremost, we note that the rate losses are very similar: 0.09 bits per 1D amplitude symbol for BL-DM and 0.1 bit for NB-DM and PA-DM. The parameters of the individual binary DMs are also given in Table III. In comparison to BL-DM, PA-DM uses three binary DMs instead of two, thus allowing a higher degree of parallelization, see also Table I. Furthermore, the output lengths n, number of input bits k and smallest number of occurrences w of either binary symbol is smaller for the component DMs of PA-DM compared to BL-DM, which potentially allows a DM implementation with a smaller number of sequential computations. As the reduction in serialism depends on the employed algorithm, we compare the degree of serialism between AC-CCDM (outlined in Sec. II-C) and SR-CCDM (introduced in Sec. IV-D) in the following.

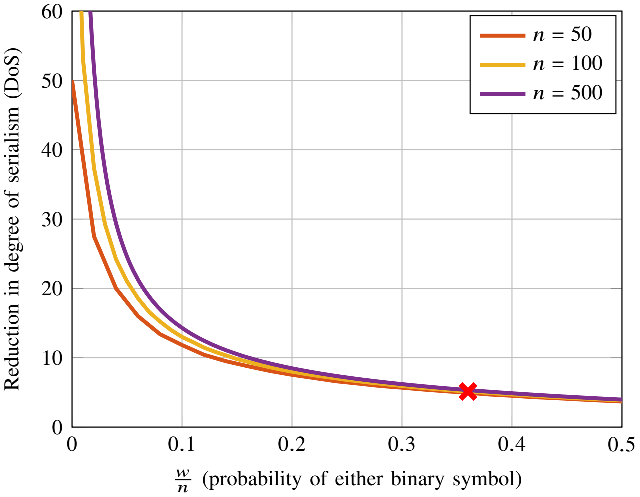

Fig. 5. Reduction in degree of serialism (DoS) of SR-CCDM compared to AC-CCDM (see (25) for the formal definition) versus the ratio between weight w of one of the binary symbols and DM length, which corresponds to the PMF for that binary symbol. The DoS reduction is shown only for $w \leq \frac{n}{2}$ as the results for larger values of w are a mirrored copy of those presented in the above figure. The marker at $\frac{w}{n} = 0.36$ corresponds to Example 7.

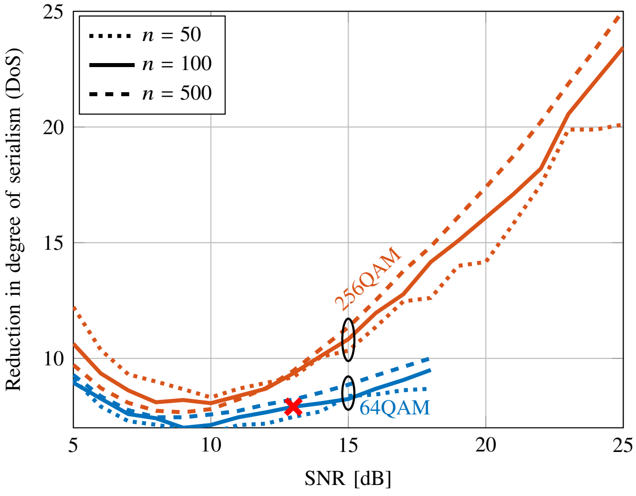

Fig. 6. Reduction in degree of serialism (DoS) of PASR-CCDM compared to AC-CCDM over the SNR in dB of an AWGN channel for 64QAM and 256QAM. The marker corresponds to Table III.

TABLE IV
DEGREE OF SERIALISM (DOS) FOR SR-CCDM AND AC-CCDM

	SR-CCDM	AC-CCDM
Mapping	$\min(w, n-w)$	k
Demapping	1 (no serialism)	n

B. Degree of Serialism (DoS) Comparison

In order to assess the computational complexity of DM algorithms, we introduce the notion of degree of serialism (DoS), which describes the number of loop iterations that is executed in either scheme for mapping and demapping. Although this metric does not incorporate the complexity or the required number of clock cycles for the operations within each iteration, it can serve as an insightful metric for evaluating the latency and the potential of parallelization for the investigated CCDM algorithms.

For SR-CCDM, the unranking algorithms have a serialism of $\min(w, n-w)$, and ranking with colex does not require any iterations (see Algorithm 2), which we define as a serialism of 1.[8] In contrast, the AC-CCDM mapping and demapping algorithms are in the worst case serial in the length of their respective inputs, which is k for mapping and n for demapping. This DoS is summarized in Table IV. The combined reduction in DoS from SR-CCDM (with colex sorting) over AC-CCDM is thus

$$\frac{k+n}{\min(w, n-w)+1}. \qquad (25)$$

For comparing parallel DM architectures schemes such as PA-DM and BL-DM, the DoS reduction is computed for the respective worst-case component DM.

In Fig. 5, the DoS reduction is numerically evaluated over $\frac{w}{n}$, which corresponds to the probability of occurrence of either binary symbol, for a CCDM with $n = \{50, 100, 500\}$ shaped bits out and $k = \log_2 \left\lfloor \binom{n}{w} \right\rfloor_2$ input bits. We observe that the stronger the binary PMF is shaped, the larger the DoS reduction, which can be more than an order of magnitude for a strongly shaped distribution. The following example illustrates the steps of this analysis for $w = 64$.

Example 7 (Serialism of Subset Ranking vs. Arithmetic Coding): Consider a CCDM with $n = 100$ and $w = 64$, which has $k = 90$ input bits. The combined worst-case serialism of AC mapping and demapping is $k+n = 190$. Mapping with SR has a serialism of $\min(w, n-w) = 36$, and demapping always has serialism 1 for colex ordering. Thus, the total reduction in serialism from the subset-ranking method is $190/37 \approx 5.14$. This reduction is shown in Fig. 5 as marker. △

Considering the example of Table III, we note that SR-CCDM is also beneficial for the BL-DM system, reducing the DoS of the worst component DM by a factor of 4.8, from $100+92 = 192$ to $39+1 = 40$. When using PA-DM instead of BL-DM, the serialism is further reduced to $32+1 = 33$, corresponding to an improvement of a factor of 5.8 from SR-CCDM. Compared to a nonbinary DM, the total reduction in serialism from jointly applying PA-DM and SR-CCDM, which is referred to as PASR-CCDM, is $261/33 \approx 7.9$, at no performance loss.

In Fig. 6, the reduction in DoS, again for the worst-case DM, is shown for 64QAM and 256QAM over the SNR of the AWGN channel. Similar to the results of Fig. 4, the DM compositions are obtained from quantized Maxwell-Boltzmann distributions. We observe that the DoS reduction can be up to a factor 10 for 64QAM, and amount to more than 20 for 256QAM. The additional PASR-CCDM rate loss compared to NB-DM was in all cases either zero or $1/n$, i.e., one extra bit. The reason for the parabola-like shape of the curves is as follows. The DoS of NB-DM grows with SNR

[8] The serial combination of the component subsequences in the PA-DM scheme is neglected here.

because for higher SNR, the distribution is more uniform-like, which in general gives a larger k and thus a higher DoS. For PASR-CCDM, however, the DoS depends on each composition (and its ordering), and for the considered compositions, the DoS is numerically found to grow fast at low SNR, causing the dip in the DoS reduction curve, while for high SNR, the DoS of PASR-CCDM grows slower than that of NB-DM.

VI. Conclusion

A DM system with parallel amplitudes (PA-DM) has been proposed that employs binary-alphabet DMs for $m-1$ out of m amplitudes (the last amplitude requires no DM). The system has no or negligibly small additional rate loss compared to a single nonbinary DM. The output lengths of the component DMs are decreasing and the number of parallel DMs grows linearly with the modulation order. These features could greatly help to increase the throughput of practical DMs.

We have further introduced a binary-alphabet CCDM mapping and demapping method via subset ranking (SR). A key difference of SR to arithmetic-coding based CCDM is that the total number of serial operations required for SR mapping and demapping is the smallest number of occurrences of either binary output symbol (i.e., the minimum weight) plus one. For SR-CCDM, the computational complexity lies mostly in calculating binomial coefficients. Combining PA-DM and SR-CCDM is numerically shown for AWGN-optimized distributions to give a serialism reduction by more than an order of magnitude compared to a nonbinary DM, which could facilitate a practical implementation of short-length CCDMs.

Acknowledgments

T. Fehenberger would like to thank Patrick Schulte (Technical University of Munich) for insightful discussions on the serialism of arithmetic-coding CCDM.

References

[1] 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.

[2] 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.

[3] J. G. D. Forney, “Trellis shaping,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 281–300, Mar 1992.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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,” *Journal of Lightwave Technology*, vol. 35, no. 22, pp. 4871–4879, Nov. 2017.

[9] 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.

[10] 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.

[11] G. Böcherer, P. Schulte, and F. Steiner, “High throughput probabilistic shaping with product distribution matching,” *arXiv preprint arXiv:1702.07510*, Feb. 2017.

[12] X. Li, J. Yu, L. Zhao, W. Zhou, K. Wang, M. Kong, G.-K. Chang, Y. Zhang, X. Pan, and X. Xin, “132-gb/s photonics-aided single-carrier wireless terahertz-wave signal transmission at 450ghz enabled by 64qam modulation and probabilistic shaping,” in *Proc. Optical Fiber Communication Conference (OFC)*. San Diego, CA, USA: Paper M4F.4, Mar. 2019.

[13] T. Prinz, P. Yuan, G. Böcherer, F. Steiner, O. İşcan, R. Böhnke, and W. Xu, “Polar coded probabilistic amplitude shaping for short packets,” in *Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, Japan, Jul. 2017.

[14] T. Matsumine, T. Koike-Akino, D. S. Millar, K. Kojima, and K. Parsons, “Polar-coded modulation for joint channel coding and probabilistic shaping,” in *Proc. Optical Fiber Communication Conference (OFC)*. San Diego, CA, USA: Paper M4B.2, Mar. 2019.

[15] S. Iqbal, M. P. Yankov, and S. Forchhammer, “Rate-adaptive probabilistic shaping enabled by punctured polar codes with pre-set frozen bits,” in *Proc. Optical Fiber Communication Conference (OFC)*. San Diego, CA, USA: Paper M4B.1, Mar. 2019.

[16] G. Böcherer, P. Schulte, and F. Steiner, “Probabilistic shaping and forward error correction for fiber-optic communication systems,” *Journal of Lightwave Technology*, vol. 37, no. 2, pp. 230–244, Jan. 2019.

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

[18] T. Fehenberger, D. S. Millar, T. Koike-Akino, K. Kojima, and K. Parsons, “Multiset-Partition Distribution Matching,” *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1885–1893, Mar. 2019.

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

[20] J. Cho, S. Chandrasekhar, R. Dar, and P. J. Winzer, “Low-complexity shaping for enhanced nonlinearity tolerance,” in *Proc. European Conference on Optical Communications (ECOC)*. Düsseldorf, Germany: Paper W.1.C.2, Sep. 2016.

[21] 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.

[22] 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.

[23] J. Cho, “Prefix-free code distribution matching for probabilistic constellation shaping,” *IEEE Transactions on Communications*, Jun. 2019.

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

[25] ——, “Arithmetic coding based multi-composition codes for bit-level distribution matching,” *CoRR*, vol. abs/1904.01819, 2019. [Online]. Available: http://arxiv.org/abs/1904.01819

[26] J. P. M. Schalkwijk, “An algorithm for source coding,” *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 395–399, May 1972.

[27] T. Cover, “Enumerative source encoding,” *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 73–77, Jan. 1973.

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

[29] 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.

[30] K. Sayood, *Introduction to data compression*, 4th ed. Elsevier Science, 2012.

[31] D. L. Kreher and D. R. Stinson, *Combinatorial algorithms: generation, enumeration, and search*. CRC press, 1998, vol. 7.

[32] F. M. J. Willems and J. J. Wuijts, “A pragmatic approach to shaped coded modulation,” in *Symposium on Communications and Vehicular Technology in the Benelux*, 1993.

[33] 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.

[34] Y. C. Gültekin, T. Fehenberger, A. Alvarado, and F. M. J. Willems, “Probabilistic shaping for finite blocklengths: distribution matching and sphere shaping,” *arXiv:1909.08886*, Sep. 2019.

[35] V. Moll, *Numbers and Functions: From a Classical-experimental Mathematician’s Point of View*. American Mathematical Society, 2012.