1

Introduction

The ever-growing need for cheaper, faster, and more reliable communication and storage systems has forced many researchers to seek means to attain the ultimate limits on reliable information transmission and storage. Low-density parity-check (LDPC) codes are currently the most promising coding technique to achieve the channel capacities (or Shannon limits) for a wide range of channels. Discovered by Gallager in 1962 [40], these codes were rediscovered in the late 1990s [83, 81]. Ever since their rediscovery, a great deal of research effort has been expended in design, construction, encoding, decoding algorithms, structural analysis, performance analysis, generalizations, and applications of these remarkable codes. Many LDPC codes have been adopted as standard codes for various current and next-generation communication systems, such as wireless, optical, satellite, space, digital video broadcast (DVB), multi-media broadcast (MMB) systems, and others. Applications to high-density data storage systems, such as flash memories, are now being seriously investigated. In fact, they appear in some recent data storage products. This rapid dominance of LDPC codes in applications is due to their capacity-approaching performance, which can be achieved with *practically implementable iterative decoding algorithms*. More applications of these codes are expected to come and their future is promising. However, further research is needed to better understand the structural properties and performance characteristics of these codes.

Most methods to construct LDPC codes can be classified into two general categories: *graph-theoretic-based* and *algebraic-based* (or *matrix-theoretic-based*) methods. The best-known graph-theoretic-based construction methods are the *progressive edge-growth* (PEG) [43, 44] and the *protograph* (PTG) based methods [105]. The algebraic-based methods for constructing LDPC codes were first introduced in 2000 [56, 55, 38, 57]. Since then, various algebraic methods for constructing LDPC codes, binary and nonbinary, have been developed using mathematical tools such as finite geometries, finite fields, and combinatorial designs [58, 76, 110, 109, 35, 107, 3, 19, 39, 74, 111, 102, 101, 112, 64, 65, 116, 97, 100, 50, 99, 113, 114, 25, 46, 26, 70, 68, 77, 78]. Most of the algebraic constructions have several important ingredients including *base matrices*, *matrix dispersion* (or *matrix expansion*), and *masking*. By a proper choice and combination of these ingredients, algebraic LDPC codes with excellent overall error performance can be constructed. Algebraic LDPC codes have, in general, much lower error-floors than randomly constructed LDPC codes. For example, algebraic LDPC codes have been recently reported that can achieve a decoded bit error rate as low as 10^{-15} without visible error-floors over the additive white Gaussian noise channel (AWGNC) [70, 68, 77, 78].

One of the earliest algebraic-based methods proposed for constructing LDPC codes is the *SP-construction* (also called the *hybrid construction*). This specific and very flexible algebraic-based method for constructing LDPC codes was devised in 2002 [76]. Since then, several other powerful algebraic-based methods have been proposed. These methods basically evolved from the SP-construction method, although the connection went unnoticed.

The SP-construction method presented in [76] starts with a small base matrix $\mathbf{B}$ and a set $\boldsymbol{R}$ of *sparse matrices* of the same size, not necessarily square. Then, each nonzero entry in $\mathbf{B}$ is replaced by a member matrix in $\boldsymbol{R}$ following certain *replacement rules* (or *constraints*) [76, 110, 109, 111, 97] and each zero entry is replaced by a zero matrix (ZM) of the same size as that of a member matrix in $\boldsymbol{R}$. The replacement operation expands the base matrix $\mathbf{B}$ into an array $\mathbf{H}$ of sparse matrices in $\boldsymbol{R}$ and/or ZMs. The null space of $\mathbf{H}$ gives an LDPC code, called an *SP-LDPC code*. We refer to this matrix replacement of a nonzero entry in $\mathbf{B}$ as *superposition*. The matrices in $\boldsymbol{R}$ are called *member* (or *constituent*) matrices and the set $\boldsymbol{R}$ is called the *replacement set*. In [110], the structural properties of the Tanner graph [103] associated with the array $\mathbf{H}$ was investigated. It was shown that the Tanner graph associated with the array $\mathbf{H}$ is an *expansion* of the Tanner graph associated with the base matrix $\mathbf{B}$.

The PTG-based method for constructing LDPC codes was introduced by Thorpe in 2003 [105]. This method was devised to construct the Tanner graph of an LDPC code of large size using a relatively small well-designed bipartite graph, called a *protograph*, as the base graph. In the construction, the first step is to choose a protograph with a *near-capacity iterative decoding threshold* [80, 34] as a building block (or as a base). The second step is to create copies of the chosen protograph. The third step is to permute the edges of the copies according to certain rules to connect them into a Tanner graph of larger size. The null space of the adjacency matrix of the resultant Tanner graph gives an LDPC code, called a *PTG-based LDPC code* (or *PTG-LDPC code*). The second and third steps of the construction together form a *graph expansion* process. Since the introduction of this graphical method for constructing LDPC codes, it has been extensively investigated over the last 12 years [105, 80, 34, 30, 31, 32, 33, 1, 11, 2, 88, 85] (see also their references). Impressive theoretical results (in terms of code ensemble properties) have been developed [34, 31, 1, 2, 85] and many PTG-LDPC codes with good error performance have been constructed. Most recently, a new class of LDPC codes has emerged, called *spatially coupled LDPC codes*, that has attracted a great amount of research enthusiasm and may be regarded as an evolution of PTG-LDPC codes from the graph-theoretical point of view.

From the above brief descriptions of the PTG-based and the SP-based methods of LDPC code construction, we can see that both methods have *two key ingredients*, *a base* and *an expansion of the base*. If we interpret these two construction methods from the same point of view, either from the graph-theoretic point of view or from the algebraic point of view, we see that they are closely related, i.e., there is a *strong connection* between the SP-construction and the PTG-based construction.

In this book, we view the SP-based and PTG-based constructions from a *broader perspective* and interpret them from both the algebraic (or matrix-theoretic) and the graph-theoretic points of view. From the algebraic point of view, we show that the SP-construction of LDPC codes includes, as *special cases*, many of the algebraic construction methods developed since 2002. From the graph-theoretic point of view, we show that the PTG-based construction of LDPC codes is a special case of the SP-construction. Furthermore, we develop an algebraic method for constructing PTG-LDPC codes.

We note that, while SP-LDPC codes contain PTG-LDPC codes as special cases from both the graph-theoretic and the matrix-theoretic points of view, their design approaches have historically been very different. SP-LDPC codes (and their relatives) have been designed using algebraic approaches with an eye toward very low decoding error-floors and highly structured (lower-complexity) decoders. By contrast, PTG-LDPC codes have a long history of being designed by searching "good" ensembles and choosing codes from the ensembles. An ensemble is considered to be good in the sense of decoding threshold [80, 34] or minimum distance [31, 32, 1]. Whether or not these ensemble approaches can be extended to SP-LDPC codes which are not PTG-LDPC codes will require further research.

Also in this book, we unify all the major algebraic methods for constructing LDPC codes based on *matrix dispersions* of base matrices under a *single framework* in terms of the SP-construction. We also introduce a new class of LDPC code with a *doubly* QC structure as well as algebraic methods for constructing *spatially coupled* (SC) and *globally coupled* (GC) LDPC codes. The constructions of these codes are also special cases of the SP-construction.

Although the focus of this book is on binary LDPC codes, all the developments, interpretations, and constructions presented for binary LDPC codes can be generalized to their nonbinary (NB) counterparts.

The rest of this book is organized as follows. Chapter 2 gives some definitions and basic concepts of matrices and introduces some fundamental structural properties and performance characteristics of LDPC codes which will be used in the later chapters. In Chapter 3, we give a brief review of the PTG-based method for constructing binary LDPC codes from the conventional graph-theoretic point of view. In Chapter 4, an algebraic method for constructing binary PTG-LDPC codes is presented. In Chapter 5, we first present the SP-construction of LDPC codes from a broader perspective than that given in [76, 109, 111]. Then, we give a graph-theoretic interpretation of the SP-construction of LDPC codes and show that the PTG-based construction is actually a special case of the SP-construction. Chapter 6 presents various constructions of base matrices and replacement sets for the SP-construction of LDPC codes. Chapter 7 presents a special type of the SP-method for constructing QC-LDPC codes. This construction method, called *matrix dispersion*, is based on dispersing (or expanding) the nonzero entries of an algebraically constructed base matrix over a finite field into circulants of the same size. We also give the necessary and sufficient conditions on a base matrix to ensure that the dispersion results in a Tanner graph with girth at least 6 or at least 8. This chapter basically puts all the major algebraic methods for constructing LDPC codes under the framework of the SP-construction. In Chapter 8, a class of LDPC codes with a doubly QC-structure is given. Chapter 9 presents algebraic methods for constructing *spatially coupled* (SC) QC-LDPC codes, both terminated and not terminated.

In Chapter 10, a new type of LDPC code, called the GC-LDPC code, is presented. Two specific methods for constructing this new type of code are presented. The first method is devised based on cyclic base arrays of matrices over NB fields. The second method is based on the *direct product* of two LDPC codes. We show that GC-LDPC codes in product form are effective at correcting erasures clustered in bursts. Also presented in this chapter is a *reduced-complexity local/global two-phase iterative decoding scheme* for GC-LDPC codes, which allows correction of *local* as well as *global* random errors and erasures. Chapter 11 generalizes the SP-construction of binary LDPC codes to construct NB LDPC codes. Several effective methods for constructing NB LDPC codes are presented. Chapter 12 concludes the book with some remarks on possible future research.

Throughout the book, examples of constructing LDPC codes and their error performance over the AWGNC and the binary erasure channel (BEC) are presented. Also, at the end of each chapter, there is a discussion on some related or unsolved issues and possible research directions.

Furthermore, three appendices are included at the end of the book. In Appendix A, two classes of arrays of circulant permutation matrices constructed based on two types of finite geometries are given. These arrays can be used to construct base matrices and replacement sets of matrices for the SP-construction of LDPC codes. They can also be used to construct well performing LDPC codes directly. Appendix B gives an algorithm for searching compatible masking and base matrices for the construction of QC-LDPC codes. Appendix C presents an iterative algorithm for decoding NB LDPC codes.