THE CATHOLIC UNIVERSITY OF AMERICA

New Iterative Inference Algorithms for Source Coding based on Markov Random Fields

A DISSERTATION

Submitted to the Faculty of the

Department of Electrical Engineering and Computer Science

School of Engineering

Of The Catholic University of America

In Partial Fulfillment of the Requirements

For the Degree

Doctor of Philosophy

By

Jose M. Fernandez

Washington, D.C.

2012

New Iterative Inference Algorithms For Source Coding Based On Markov Random Fields

Jose M. Fernandez, Ph.D.

Director: Phillip Regalia, Ph.D.

The global deployment of wireless communication systems poses significant challenges for system designers which have to accommodate an ever-increasing number of users while simultaneously meet demands for increased levels of security and privacy. Many of these problems involve aspects of lossy source coding that are yet to be well understood. For instance, the nature and combined effectiveness of sparse graphs and message-passing algorithms in source coding continues to be the subject of debate and active research. This is in stark contrast to the channel coding case where specific capacity-approaching codes (i.e. Turbo Codes, Low-Density Parity Check Codes, etc.) and classical message-passing schemes (i.e. Belief Propagation) are clearly understood, widely accepted, and increasingly in use. Furthermore, the emergence of cavity methods drawn from statistical physics (i.e. Survey Propagation) gave rise to the widespread assumption that the source coding problem could not be solved by simple Belief Propagation-based iterations over Markov Random Fields.

This notion is challenged heretofore by the introduction of two novel message-passing algorithms. These two simple schemes, namely Truthiness Propagation and Modified Truthiness Propagation, are developed based upon modified Bethe free energy approximations (equivalent to log-partition function approximations) and shown to be closely related to Belief Propagation, thus situating them on firm theoretical

ground. The new algorithms exhibit rate-distortion performance near the Shannon limit even for modest codeword lengths when combined with both regular and irregular Low-Density Generator Matrix Codes. This feature offers a distinct advantage not seen with other message-passing schemes. Furthermore, their complexity is manageable since the decimation steps prevalent in other recently proposed techniques are not required.

Finally, these modified instantiations of Belief Propagation are applied to a number of applications relevant to the codeword quantization problem (i.e. general decoding problem) via simple examples in dirty paper coding, data hiding, secrecy coding, and wireless sensor networks.

This dissertation by Jose M. Fernandez fulfills the dissertation requirement for the doctoral degree in Philosophy approved by Phillip Regalia, Ph.D., as Director, and by Nader Namazi, Ph.D., and Mohammed Arozullah, Ph.D., as Readers.

Phillip Regalia, Ph.D., Director

Nader Namazi, Ph.D., Reader

Mohammed Arozullah, Ph.D., Reader

Contents

List of Figures

List of Tables

List of Abbreviations

BC Broadcast Channel

BP - Belief Propagation

BiP - Bias Propagation

BSC - Binary Symmetric Channel

BVP - Bethe Variational Problem

CSI - Channel State Information

CDI - Channel Distribution Information

DPC - Dirty Paper Coding

GF - Galois Field

LDGM - Low Density Generator Matrix

LDPC - Low Density Parity Check

LLR - Log-Likelihood Ratio

LSB - Least Significant Bit

MAC - Multiple Access Channel

MEMS - Micro-Electro-Mechanical Systems

MIMO - Multiple Input Multiple Output

MRF - Markov Random Field

MTP - Modified Truthiness Propagation

TAP - Thoules-Anderson-Palmer

TCQ - Trellis Coded Quantization

TIF - Tagged Image File

TP - Truthiness Propagation

SISO - Single Input Single Output

SP - Survey Propagation

WSN - Wireless Sensor Network

Acknowledgements

This dissertation bears the fruit of the sacrifice and effort of many people besides the author whose contributions and inspiration helped me to navigate through this long and very arduous journey. First, I would like to praise our lord Jesus Christ for giving me both the physical and mental strength to carry me through this endeavor.

I am forever indebted to my loving wife Yeimary Santos. She is the love of my life and there is no way that I could have made it this far without her unconditional love and support. She is deserving of much of the credit for this accomplishment as the author himself. My son Yamil, whose smile and laughs helped me get through many difficult moments. His arrival gave me a totally new perspective and focus to follow through with this task.

I wish to express my eternal gratitude to my thesis advisor, Dr. Phillip Regalia, for his unusual patience during my many struggles and for providing invaluable guidance in key moments during my research. His continuous understanding of my other life and work commitments was crucial in order to make this dissertation a reality. My deepest gratitude is also due to the members of the dissertation committee, Dr. Nader Namazi and Dr. Mohamed Arozullah for without their sage advice and input this work would not have been successful.

I also wish to express love and gratitude to my beloved extended family; my parents Jose and Maria and my sisters Santializ and Virneliz; for their understanding

and endless encouragement through the duration of my studies.

I would also like to thank the Johns Hopkins University Applied Physics Laboratory, for their unwavering financial support and my line supervisors Dan Fenner and John Schmidt not only for constantly encouraging me to pursue graduate studies but for giving me the flexibility to do so.

Last but not least, I would like to acknowledge my late grandfather, Lorenzo Fernandez, whose exemplary love, humility, work ethic, and attitude have been a lifelong inspiration to me. This work is dedicated to his memory.

1

Introduction to Markov Random Fields and Factor Graphs

The recent emergence of advanced codes and algorithms based on graph constructions underscores the increasingly important role that graphical models play in addressing many challenges in modern communication systems [1, 2]. The formalism afforded by graphical models allows for both easy and insightful representation of highly complex multivariate statistical functions as well as the development of methods to extract useful information from them. These extraction methods, commonly referred to as statistical inference, form the basis of the applicability of graphical models to source and channel coding problems. The objective of statistical inference in these applications is often to compute the marginal distributions from the joint probability density function being represented by the graph model. In the context of source coding, the ef-

ficient (albeit approximate) computation of marginal distributions is key in providing minimum weight (or minimum energy) representation of data (i.e. data compression) for efficient storage and/or transmission [3]. The insight and computational tractability of these approximation methods stems from a certain class of graph model called a Markov random field (MRF) [4]. In turn, the benefits of Markov random fields translate nicely into an even more useful graphical representation for coding applications called a factor graph. These factor graph representations not only serve to elucidate classical statistical inference algorithms but open the door to the development of new alternatives designed to address challenging problems in multi-user communications and many other areas [5].

1.1 Graphical Models

Graphical models are probabilistic models represented by graphs which denote statistical relationships among random variables [6]. They combine multiple elements of graph theory and probability theory in a framework amenable for visualization and analysis of multi-dimensional probability distributions. A graph $G(V, E)$ consists of a group of vertices (or nodes) V and a group of edges E. Each vertex $s \in V$ represents a random variable x_s with a given domain X_s. The edges interconnect random variables and represent statistical dependencies among them. Probabilistic graphical models can be classified in two broad categories, namely Bayesian belief networks (BBN) and Markov random fields [6]. These two classes are also known as directed and un-directed graphs, respectively. In the case of directed graphs, every single edge

is routed from parent vertices to child vertices. The overall (or joint) probability distribution in a directed graph can be written as follows:

$$p(\mathbf{X}) = \prod_{s \in V} p(x_s | \text{parent}(x_s))$$

where $\mathbf{X}$ denotes the ensemble of random variables x_s and parent(x_s) is the state of the group of vertices connected to vertex x_s. Figure 1.1 shows a relatively simple example of a directed graph (BBN) composed of three vertices and three edges. The

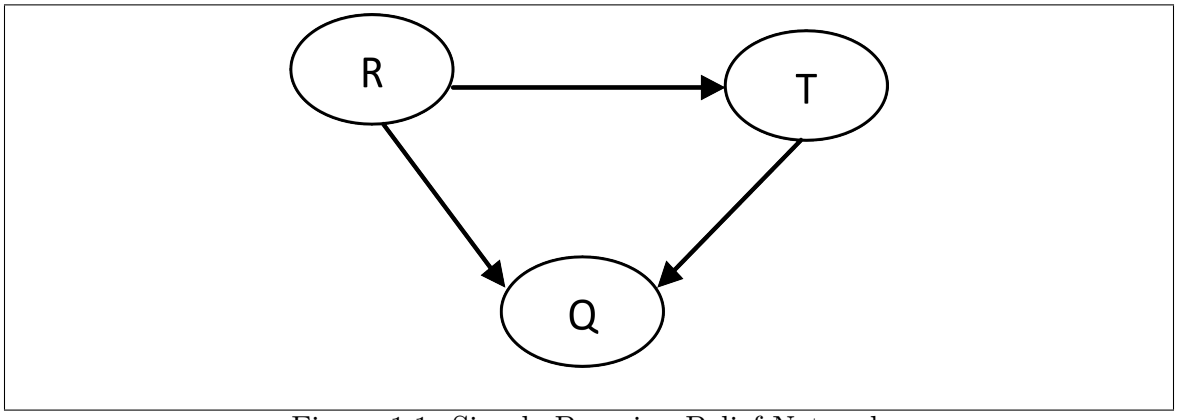

Figure 1.1: Simple Bayesian Belief Network

joint probability distribution for the Bayesian belief network shown in Figure 1.1 is given by:

$$P(R, T, Q) = P(Q|R, T)P(T|R)P(R)$$

Note how the joint probability function can be written as the product of the individual vertex distribution functions conditional on the state of their parent vertices. This also implies that any particular random variable in the graph is conditionally independent of all others given its neighboring random variables [7]. This is known as the local Markov property. Bayesian belief networks satisfy the local Markov

property. Bayesian belief networks are very useful mechanisms when attempting to capture causal relationships among random variables. Important instances of such models are evident in hidden Markov models and neural networks [8].

The other major class of graphical models is the un-directed graph, or Markov random field. A Markov random field is the multi-dimensional generalization of the better-known Markov chain [8]. Markov random fields are typically used to represent spatial dependencies among random variables in a probabilistic model. The concept of Markov random fields has been widely used in statistics as well as computer vision and other image processing applications [4]. A number of preliminary concepts need proper definition in order to understand Markov random fields. Each and every random variable in a Markov random field is part of a neighborhood system N defined simply as:

$$N = N_j \quad | \quad \forall j \in V$$

where N_j is the local neighborhood surrounding variable j, which is part of the set V of random variables called a random field. A random variable cannot be a neighbor to itself and the relationship between neighboring variables has to be mutual [4]. A variable k within the random field V is considered a neighbor of variable j if:

$$N_j = j \in V \quad | \quad d(j,k) \leq h, j \neq k$$

where $d(j,k)$ is the distance between the random variables and its specific definition depends on the context. The random field V and the set of neighborhoods N form an undirected graph where the vertices are the random variables in V and the set

N defines the connections among the vertices. The cliques in this type of graph are defined by subsets of connected vertices in V. They can be single-variable cliques, dual-variable cliques, and so forth. The union of all cliques constitutes the undirected graph defined earlier. If the variables in the random field $V = (V_1, V_2, \ldots, V_k)$ are given the values $v = (v_1, v_2, \ldots, v_k)$ in a certain domain, then the probability that V_j is equal to v_j can be expressed by $P(V_j = v_j) = P(v_j)$. For joint events, similar definitions are valid (i.e. $P(V = v) = P(v)$). Therefore, V is a Markov random field if it has the following two properties [5, 8]:

- Positive Property: $P(v) > 0, \forall v \in$ all possible configurations of v
- Markov Property: $P(v_j|v_{V-j}) = P(v_j|v_{N-j})$

where the subscript $V - j$ implies all the variables in the set V except for j and N_j is the neighborhood of vertices around j.

Markov random fields could be specified either in terms of conditional probability distributions or joint probability distributions. Both of these probability descriptors are typically very difficult to obtain directly, and even when the conditional distributions are available the transition to a joint distribution is virtually intractable [7, 8]. Nevertheless, if the undirected graph satisfies the two properties stated earlier then it is possible to characterize the Markov random field as a factorization over the sets of interconnected vertices (cliques) of the graph. Following the same notation used earlier to describe the joint distribution of a Bayesian belief network, the joint

probability distribution of a Markov random field is:

$$p(\mathbf{X}) = \frac{1}{A} \prod_{C \in \text{cliques}(G)} \Psi_C(x_C)$$

where the product of clique potentials Ψ_C is taken over all cliques in the graph and A is a normalization constant. The joint distribution function description in terms of products of clique potentials originates from the equivalence between Markov random fields and Gibbs random fields established by the Hammersley-Clifford theorem [9]. Unlike Markov random fields, Gibbs distributions are described by a global function of the form [4]:

$$P(v) = \frac{1}{A} \exp\left[-\frac{1}{T} U(v)\right]$$

where A is called the partition function, T is a temperature constant (assumed to be 1 without loss of generality) and $U(v)$ is the energy function. The partition function is given by:

$$A = \sum_{v \in \text{ all configurations in } V} \exp\left[-\frac{1}{T} U(v)\right]$$

where $U(v)$ is expressed as follows:

$$U(v) = \sum_{c \in C} \Psi_c(v)$$

where the term Ψ_c represents the potential of clique c. Clearly, the importance of the Hammersley-Clifford theorem cannot be overstated since it provides a path to specify

a Markov random field in terms of its factorized clique potentials [9]:

$$P(v_j|v_{V-j}) = \frac{\exp\left[-\sum_{c\in C_j}\Psi_c(v)\right]}{\sum_{v\in \text{all configurations in } V}\exp\left[-\sum_{c\in C_j}\Phi_c(v)\right]}$$

where the potentials $\Psi_c(v)$ are calculated only over the cliques that contain variable j.

A Markov random field example commonly used in image processing is the pairwise Markov random field. Consider the case (without loss of generality) in which every vertex has an observation node attached. Figure 1.2 shows the resulting lattice Markov random field. The filled circles represent the observation nodes $\mathbf{Y} =$

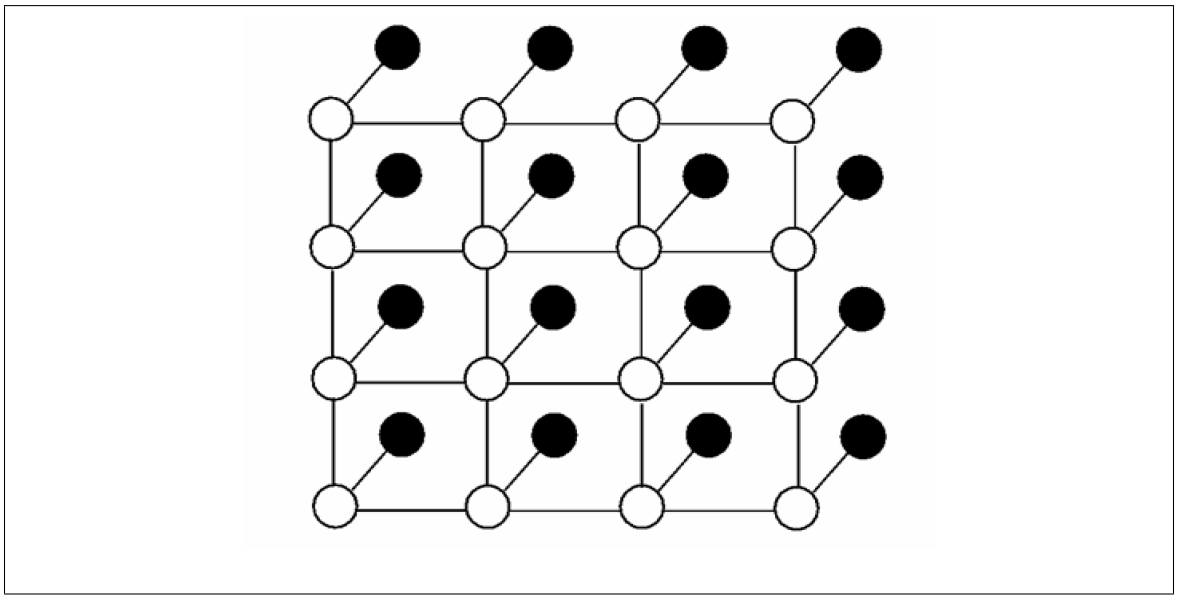

Figure 1.2: Square Lattice Pair-wise Markov Random Field [7]

$[y_1, \ldots, y_n]^T$ while the other circles are the underlying nodes $\mathbf{X} = [x_1, \ldots, x_n]^T$. The edges represent spatial compatibility functions (potentials) $\Psi_{ij}(*,*)$ between nodes at positions i and j on the lattice. The Hammersly-Clifford theorem yields the following

joint probability density function [9]:

$$p(X,Y) = \frac{1}{A}\prod_{i,j}\Psi_{i,j}(x_i,y_j)\prod_{i}\Psi_{i,i}(x_i,y_i)$$

In computer vision, the pairwise Markov random field is used to describe the spatial constraints among pixels in an image [4]. The set of nodes Y sense information from the image and the nodes X are used to infer information from the underlying scene.

Another interesting aspect of Markov random fields is that they could be converted into equivalent Bayesian belief networks (via factor graphs) and vice versa. The details about this procedure are discussed in [5, 8].

1.2 Factor Graph Representation

A particular class of undirected graphs, called factor graphs, has proven to be more suitable for coding applications [5]. The factorized description of a joint distribution represented by a Markov random field has a structure fit for a factor graph description. In fact, the same probabilistic model can be described interchangeably without any loss in generality by either a factor graph or a Markov random field. In a general sense, a factor graph is a convenient visual representation of any mathematical function or expression. In a formal sense, it is an undirected bipartite graph of variables on one side that serve as arguments of local functions on the other side [5]. These local functions are factors in the product that yields an overall global function. The bipartite nature of factor graphs means that the vertices are divided into two dis-

joint sets, namely that the set V in $G(V, E)$ is split between the subset of variables nodes and the subset of factors nodes. Figure 1.3 shows a pictorial view of a factor graph. Variables are represented by circles on the left hand side and the factors are represented by squares on the right side. The global function is the product of all the factors. In turn, these factors are a function of the subset of variables connected to each.

An important aspect of the factor graph shown in Figure 1.3 is that it does not contain any cycles. A factor graph is cycle-free if all traces which begin at one variable node end at a different variable node. Conversely, the factor graph shown in Figure 1.4 contains multiple cycles. Note that traces in this factor graph do not have a distinct beginning or end and continue to loop around indefinitely. The primary reason to identify cycles in a factor graph is that the algorithms that typically operate over these graphs are dramatically impacted due to their presence [5].

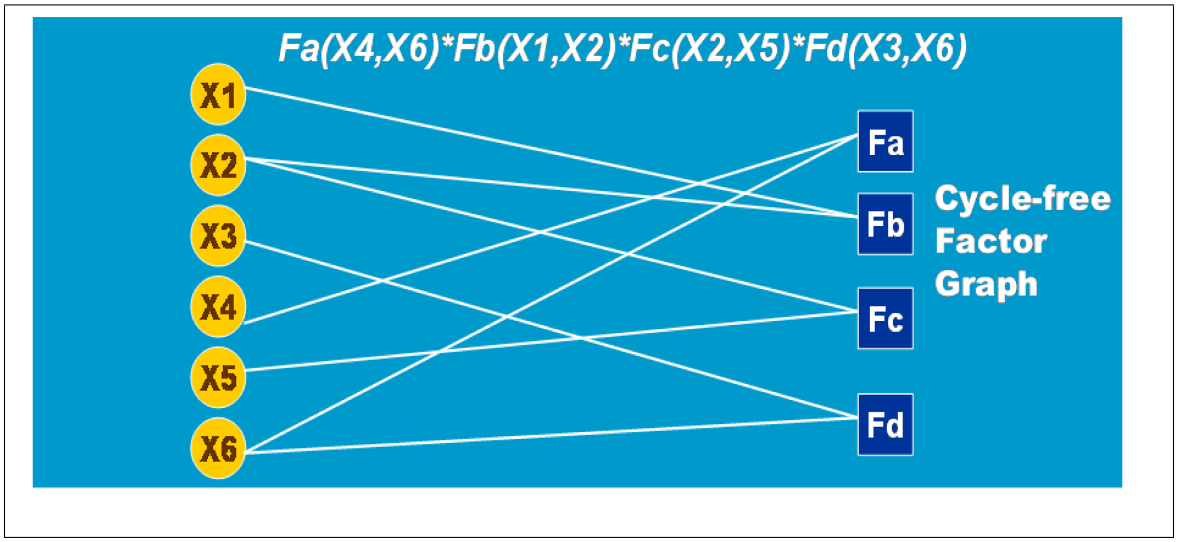

Figure 1.3: Factor Graph without Cycles

The joint density function given by the factor graph in Figure 1.3 is:

$$p(\mathbf{X}) = F_a(x_4, x_6)F_b(x_1, x_2)F_c(x_2x_5)F_d(x_3, x_6)$$

where $\mathbf{X} = [x_1, x_2, \ldots, x_6]$ is the set of variable nodes and the F_j for $j = [a, b, c, d]$ is the set of factor nodes.

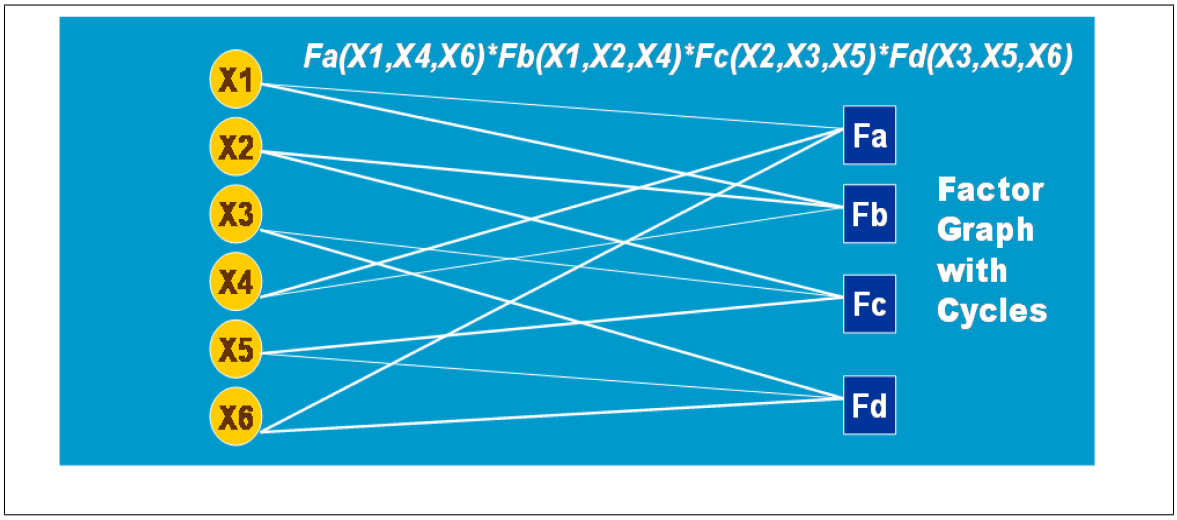

Figure 1.4: Factor Graph with Cycles

Factor graphs are a generalization of Tanner graphs which were introduced to describe certain error-correcting codes [10, 11]. In essence, the Tanner graphs use the variable nodes to represent individual bits in a codeword and the function nodes represent parity check nodes. The two graphs are very similar but in factor graphs a function node is allowed to represent any arbitrary mathematical relationship among the variables connected to it.

These types of graphs afford a simple way of understanding a large family of inter-related algorithms that process complex global functions based upon simple local computations [8]. These algorithms exploit the factor graph structure in order

to perform certain inference tasks. Even though factor graphs originated from coding theory and are commonly used in that context, they are also being used to describe more general probabilistic and behavioral models to support a wide array of applications [5]. As stated before, it is relatively straightforward to obtain factor graph representations from both Bayesian belief networks and Markov random fields.

1.3 Statistical Inference over Markov Random Fields

Beyond the use of Markov random fields (and other graph types) to model certain problems, the ultimate goal is to enable, and develop, algorithms to efficiently extract information of interest out of highly complex probability distributions by exploiting their internal graphical structure [7]. Ideally, if this structure is deemed to be sparse enough, there are relatively simple algorithms that extract the desired information efficiently. Nevertheless, many probabilistic models of interest have non-sparse graphical structures so different methods need to be considered. A common alternative is to use Markov chain monte-carlo methods and there is abundant literature about the subject and its applications [12]. One area where Markov chain monte-carlo methods fall short in their implementation is in regards to error-control coding [7]. Thus, an alternate path needs to be pursued given the relevance of channel and source coding to the present work. Fortunately, a different framework based on variational principles has sparked recent interest and provided the foundation for the development of an extensive set of algorithms designed to reduce the computational burden involved in statistical inference problems [7, 13]. Furthermore, the novel iterative inference

methods to be developed and discussed subsequently are heavily anchored to these variational methods. The roots of all these methods can be traced to statistical physics.

In a broad sense, these methods perform some sort of statistical inference operation across a graphical model. The types of inference tasks can be boiled down into two main categories [7]:

1. Computation of marginal distributions for a single variable node or over a subset of the variable nodes.
2. Computation of modes of the joint density distribution.

The first task above is of utmost importance in many applications including source coding. As such, the focus going forward will be centered on this particular statistical inference task. The graphical model structure (i.e. cycle-free or not) is an important factor in considering whether certain algorithms can be brought to bear to perform exact statistical inference on the underlying probability distribution. Given the nature of the problems at hand, the present focus could be further narrowed to inference approximation algorithms instead. Thus, returning to the probability distribution described by a Markov random field:

$$p(\mathbf{X}) = \frac{1}{A} \prod_{c \in C} \Psi_c(x_c)$$

where Ψ_c are potential functions restricted to single or two-node cliques in C and A is the normalization or partition function. The joint probability density described by

the Markov random field above can be re-written in terms of exponential family distributions [7]. The rationale for introducing exponential distributions is due to their convexity properties which are very useful in optimization problems. The tailored expression then becomes:

$$p(\mathbf{x}, \theta) = \exp\left[\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t - A(\theta)\right]$$

The third term in the exponent above is the well-known log partition function defined by the expression [7]:

$$A(\theta) = \log \int_{\chi^n} \exp\left[\theta, \phi(\mathbf{x})\right] v(\mathrm{d}\mathbf{x})$$

where $\phi(\mathbf{x})$ represents a collection of potential functions defining the mapping $\chi^n \to \mathbb{R}$ on the base measure v defined via $\mathrm{d}v = h(\mathbf{x})\mathrm{d}\mathbf{x}$, with $h(\mathbf{x})$ being arbitrary and $\mathrm{d}\mathbf{x} = \prod \mathrm{d}x_s$ being the counting measure with respect to the mapping above. Note that the probability distribution $p(\mathbf{x}; \theta)$ strictly applies to pair-wise Markov random fields. A closed-form description of $p(\mathbf{x}; \theta)$ is usually required but unattainable due to the inherent complexity of some of the terms in the expression defined earlier. Even if the overall probability distribution was available, the problem of computing marginals becomes intractable due to the large number of summations that must be made [7]. These difficulties inevitably lead to approximation methods.

As stated earlier, a relevant class of such approximate inference methods is based on the variational principle [13]. The term variational refers to mathematical techniques of formulating inference problems as optimization problems and their respective solution. The general idea is to pose the marginal distributions of interest as

the prospective solution to an optimization problem [14]. This optimization problem can be solved notionally by relaxing some of the conditions by which the optimization takes place yielding the desired marginals. Consequently, given the distribution $p(\mathbf{x};\theta)$ defined above, it has been established that the log partition function A is the solution to the following optimization problem [14]:

$$A = \max_{q \in \mathcal{Q}} \left(\sum_{x} q(x) \left[\sum_{c} \log \Psi_c(x) \right] - \sum_{x \in \chi^n} q(x) \log q(x) \right)$$

where $\Psi_c(x)$ represent the clique functions (or potentials) along the graph. This expression is uniquely maximized when $q = p(\mathbf{x};\theta)$. The probability distribution q belongs to the set of all distributions on the χ^n discrete space called $\mathcal{Q}$. Hence, the rationale of the variational approach is to obtain a $q \approx p$ by approximating the entropy term in the maximization expression above and choosing a suitable set $\mathcal{Q}$ to maximize over.

Returning to the log partition function $A(\theta)$, a well-known result of this function is that it is a conjugate dual of itself [7]. This property is represented by the optimization expression below:

$$A(\mu) = \sup_{\theta \in \mathbb{R}^d} \{\langle \mu, \theta \rangle - A^*(\theta)\}$$

where $\mu = E_p[\phi(\mathbf{x})]$ maintains the expression above bounded as long as it belongs to the relative interior of $MARG(G)$. The set $MARG(G)$ refers to the marginal polytope and is defined by the collection of potentials $\phi(\mathbf{x})$ belonging to the graph $G(V, E)$. The polytope contains the set of realizable μ vectors that validate the

conjugate duality of $A(\theta)$. In other words, μ vectors lying outside of this polytope force the supremum expression above to be unbounded [14].

Contrasting the conjugate duality expression for $A(\theta)$ above with the classical variational principle presented earlier, in the former the optimization takes places over a different space (μ vectors in $MARG(G)$) rather than the space of all distributions as in the latter. One challenging aspect about the optimization expression is that the size of the marginal polytope grows very quickly with increasing graph size making it intractable to compute the set exactly. Another problem is that the dual log partition function is available in closed form only for cycle-free graphs. Some of these difficulties can be circumvented rather nicely by imposing certain constraints on the marginal polytope as described subsequently in chapters 2 and 3 [7, 14].

2

Basics of Message-Passing Algorithms and Source Coding

Given the importance of statistical inference in many science and engineering fields, it is not surprising that developing efficient inference algorithms became the focal point of multiple research fronts over the last four decades [13]. These algorithms, collectively known as message-passing algorithms, are used to extract relevant features of probability functions by allowing the nodes in a graphical model to share or pass messages about their state to surrounding nodes. Their development has followed a rather curious path with flurries of breakthroughs and gaps along the way. This is evident by the fact that similar (if not the same) schemes have been independently discovered multiple times in many different disciplines [5]. In the context of coding theory, message-passing algorithms could be traced back to the pioneering work

of R.G. Gallagher, where an a posteriori probability scheme to decode his newly-developed low-density parity check (LDPC) codes was introduced [15]. Nonetheless, as shown later, their theoretical underpinnings are deeply rooted in developments in the field of statistical physics in the early parts of the 20th century [16, 17]. In essence, message-passing algorithms are nothing more than dynamic programming schemes designed to share intermediate terms among nodes in the computation of a quantity of interest along a graph. The most important one of all is called belief propagation (BP) and is the central theme in what follows.

2.1 Belief Propagation

It could be argued that the majority of message-passing algorithms are nothing more than instantiations of the belief propagation algorithm. The term belief propagation was coined by J. Pearl in his ground-breaking work [18]. Belief propagation is a recursive algorithm devised to perform statistical inference based upon passing messages along a graph. It is also known in the information theory literature as the sum-product algorithm [5]. In general terms, belief propagation computes the marginals of functions described on certain graphs. The graphs on which belief propagation typically operates are factor graphs and Markov random fields, although its update equations are not dependent on the particular structure of the graph. The messages that are passed around over these graphs have a relatively simple interpretation when the underlying model in the graph is stochastic in nature [8]. The algorithm initializes the variable nodes in the graph with a-priori probabilities. Each node sends a mes-

sage to a connecting node if it has received messages from all of its other neighboring nodes. This type of message could be interpreted as a conditional probability. The messages continue to propagate in this manner until all messages have been sent exactly once in every direction. Upon termination, the marginal of a variable is simply the product of the incoming messages from all its adjacent nodes.

In the context of factor graphs, there are two types of messages since there are two types of nodes: variable and function nodes. Messages are defined in a very similar manner for other types of graphs. The message from a variable node X_n to a function node f_m is defined as follows [5]:

$$X_n \ \rightarrow \ f_m(x_n) = \prod_{f_i \in N(X_n) \backslash \{f_m\}} f_i \ \rightarrow \ X_n(x_n)$$

The message from a function f_m to a variable node X_n is defined by the following expression:

$$f_m \ \rightarrow \ X_n(x_n) = \sum_{x_m : X_n = x_n} f_m(\mathbf{x}_m) \prod_{X_i \in N(f_m) \backslash \{X_n\}} X_i \ \rightarrow \ f_m(x_i)$$

where $N(\cdot)$ implies the neighboring nodes of the argument and $\mathbf{x}_m$ is the equivalent of $N(f_m)$. The index $N(\cdot)\backslash\{\cdot\}$ means all the neighbors of the node in the argument except for the node inside the keys. The marginal, or belief $z(x_i)$, of a variable X_i can be computed upon completion of the algorithm as follows [5]:

$$z(x_i) = \prod_{f_j \in N(X_i)} f_j \ \rightarrow \ X_i(x_i)$$

Or,

$$z(\mathbf{x}_i) = f_j(\mathbf{x}_j) \prod_{X_i \in N(f_j)} X_i \rightarrow f_j(x_i)$$

Figure 2.1 shows the bi-directional flow of messages along a simple cycle-free factor graph. It is important to mention that the algorithmic procedure above only applies to graphs without cycles. Messages are passed only twice over each edge and the exact posterior probabilities is computed afterwards. For the case of graphs with cycles, the initialization phase is not well defined. The variable nodes in the graph are not singly connected so there is ambiguity as to where to start the iteration. Furthermore, once the recursions begin the algorithm will continue to iterate passing messages back and forth along the graph endlessly unless a stoppage is forced. Fortunately, some of these difficulties have been somewhat circumvented and it has been found that standard belief propagation often produces performance results that are very close to the Shannon capacity when used to decode sparse graphical codes [19]. Nonetheless, the marginals computed by the belief propagation algorithm are only approximate for the case of graphs with cycles and its convergence is no longer guaranteed [5].

A crucial theoretical link has been found between belief propagation and the concept of free energy in statistical physics which helps to explain the success of BP as well its shortcomings for certain graphs [20]. If we are given a physical system of N dimensions, where each dimension can adopt a particular state x_n, then (under

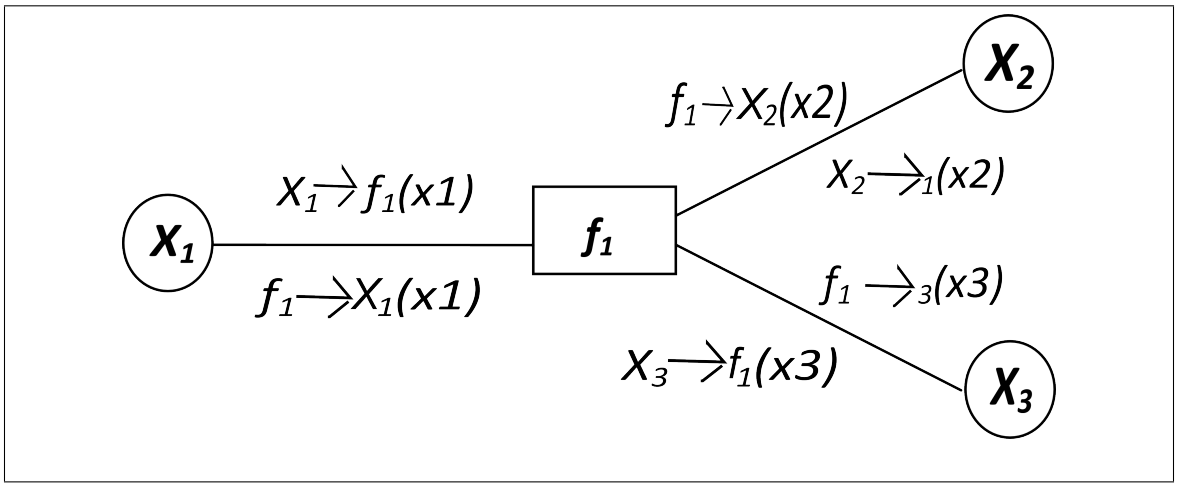

Figure 2.1: Belief Propagation Messages passed along a cycle-free Factor Graph

thermal equilibrium) the joint probability of a state x is defined by Boltzmanns law:

$$p(\mathbf{x}) = \frac{1}{A(T)} \exp\left[-\frac{E(\mathbf{x})}{T}\right]$$

where T is the system temperature and $A(T)$ is known as the partition function defined as follows:

$$A(T) = \sum_{x \in S} \exp\left[-\frac{E(\mathbf{x})}{T}\right]$$

where S is the set of all possible state configurations of $\mathbf{x}$ and $E(\mathbf{x})$ is the energy of the system corresponding to the state configuration $\mathbf{x}$. Hence, the Helmholtz free energy F of a system is [8, 20]:

$$F = -\ln A$$

If we are given the joint probability function $p(\mathbf{x})$ we could use Boltzmanns relationship to represent the system energy. For instance, if $p(\mathbf{x})$ is given by the

factorization:

$$p(\mathbf{x}) = \frac{1}{A}\prod_{a=1}^{N} f_a(x_a)$$

Substituting the equation above in Boltzmanns equation (and setting $T = 1$) and solving for the energy $E(\mathbf{x})$ the following is obtained:

$$E(\mathbf{x}) = -\sum_{a=1}^{N} \ln f_a(x_a)$$

Oftentimes the joint probability function $p(\mathbf{x})$ has a cumbersome structure that makes the free energy difficult to compute. Thus, in order to compute the free energy F the joint probability distribution needs to be estimated. One way of estimating the free energy is via the variational approach [7, 8, 20]. This concept was introduced in chapter 1 and will be further developed here. The variational approach introduces the term $b(\mathbf{x})$ (equivalent to the term $z(\mathbf{x})$ presented earlier) dubbed the belief which attempts to approximate $p(\mathbf{x})$ and the variational (Gibbs) free energy. The variational free energy equation is given by:

$$\begin{aligned} F(b(\mathbf{x})) &= \sum_{x \in S} b(x)E(x) + \sum_{x \in S} b(x)\ln b(x) \\ &= F + D(b(\mathbf{x})\|p(\mathbf{x})) \end{aligned}$$

where the term $D(b(\mathbf{x})\|p(\mathbf{x}))$ is the Kullback-Liebler distance between $b(\mathbf{x})$ and $p(\mathbf{x})$ [8, 20]. Therefore, at least in theory, minimizing the Gibbs free energy provides an exact method to compute the Helmholtz free energy and the joint probability density $p(\mathbf{x})$. Nonetheless, the variational free energy expression above is often mathemati-

cally intractable so a number of free energy approximations have been proposed to address this problem.

2.1.1 Mean-Field Free Energy Approximation

The simplest approach which yields an analytically tractable approximation to the Gibbs free energy is called the mean field approximation. In this case, the joint probability distribution is approximated by the product of the single node beliefs [7, 8, 20]:

$$p(\mathbf{x}) \cong b(\mathbf{x}) = \prod_i b_i(x_i)$$

where the beliefs $b_i(x_i)$ are subject to the following constraint:

$$\sum_i b_i(x_i) = 1$$

Additionally, the condition that the pair-wise node beliefs decompose as the product of the corresponding single node beliefs is also imposed: $b_{ij}(x_i, x_j) = b_i(x_i)b_j(x_j)$. These assertions can be substituted in the Gibbs free energy equation shown earlier to obtain the following:

$$U_{MF}\{b(x)\} = -\sum_{a=1}^{M} \sum_{x_a} \ln f_a(x_a) \prod_{i \in N(a)} b_i(x_i)$$

$$H_{MF}\{b(x)\} = -\sum_{i=1}^{N} \sum_{x_i} b_i(x_i) \ln b_i(x_i)$$

Thus, the mean filed approximation to the Gibbs free energy is given by:

$$F_{MF} = U_{MF} - H_{MF}$$

Where the term U_{MF} is the mean field approximation to the variational average energy and H_{MF} is the mean field approximation to the variational entropy. Likewise, additional approximations could be obtained depending on the assumptions made about the structure of the joint density function. A slightly different approach will be taken in section 2.1.2 based on selecting regions across the graph [20]. This approach is more insightful and produces better approximations than the mean field technique. Furthermore, that approach leads directly to the BP algorithm.

2.1.2 Bethe Free Energy Approximation

A well-known approximation called the Bethe free energy attempts to estimate both the variational average energy as well as the variational entropy term by employing both single node and pair-wise node beliefs. The remarkable theoretical connection alluded to earlier is that the fixed points derived from belief propagation are equivalent to the stationary points (beliefs) in the Bethe free energy approximation. For cycle-free graphs, the Bethe approximation becomes equal to the Gibbs free energy [20].

Since the variational entropy term is only an approximation, it partially explains the performance (or lack thereof) of belief propagation. It also provides much needed insight into multiple improvements that can be made to the standard version

of the BP algorithm. Standard BP is also a special case of a group of techniques sometimes referred to as generalized belief propagation [8, 20]. Generalized belief propagation algorithms are essentially based on lumping together regions of nodes and passing messages among these regions in lieu of passing messages just among single nodes. Therefore, they are referred to as constrained region-based approximations. In the context of pair-wise MRFs, if the regions are constrained to single nodes, then the mean-field energy approximation is obtained. If the regions are chosen as node pairs instead, then the Bethe energy approximation is obtained. If a factor graph describes the joint density function, then two particular regions are chosen:

1. The regions encompassing each factor node and its surrounding variable nodes.
2. The regions formed by single variable nodes.

Since these regions will invariably overlap, the concept of counting numbers was introduced to ensure that every node is accounted for only once when approximating the Gibbs free energy [20]. The particular choice of regions above along with the enforcement of normalization, consistency, and inequality constraints among the beliefs constitutes the Bethe approximation. Examples with other choices of regions are illustrated by Kikuchis cluster variation method [17, 20].

The derivation of the Bethe free energy approximation stationary points (BP fixed points) simply becomes a constrained optimization exercise. Given the two

selected regions and constraints specified above a Lagrangian expression is obtained:

$$L(b,\lambda) = F_{Bethe} + \sum_i \lambda_i \left[\sum_{x_i} b_i(x_i) - 1 \right] + \sum_a \lambda_a \left[\sum_{x_a} b_a(x_a) - 1 \right] + \sum_i \sum_{a \in N(i)} \sum_{x_i} \lambda_{ai} \left[b_i(x_i) - \sum_{x_a \setminus x_i} b_a(x_a) \right]$$

where F_{Bethe} is the Bethe free energy approximation given by:

$$U_{Bethe} = -\sum_{a=1}^{M} \sum_{x_a} b_a(x_a) \ln f_a(x_a)$$

$$H_{Bethe} = -\sum_{a=1}^{M} \sum_{x_a} b_a(x_a) \ln b_a(x_a) + \sum_{i=1}^{N} (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i)$$

defined on a factor graph with M factor nodes and N variable nodes. The quantity d_i represents the degree of each variable node (number of nodes connected to node i). Taking the derivative with respect to both $b_a(x_a)$ and $b_i(x_i)$ and solving for the respective beliefs gives the following:

$$b_i(x_i) \propto \prod_{c \in N(a)} m_{ca}(x_a)$$

$$b_a(x_a) \propto f_a(x_a) \prod_{i \in N(a)} \prod_{c \in N(i)} m_{ci}(x_i)$$

which, together with the normalization and marginalization equations:

$$\sum_{x_a} b_a(x_a) = 1; \sum_{x_i} b_i(x_i) = 1$$

$$\sum_{x_a \setminus x_i} b_a(x_a) = b_i(x_i)$$

are the fixed points of the standard belief propagation algorithm [8, 20].

2.1.2.1 Log-Partition Function Interpretation of Bethe Approximations

An alternative viewpoint of the Bethe approximation that is relevant to subsequent developments will be described next. This formulation is based on approximating the variational Bethe entropy with approximations to the log-partition function [7, 14, 21]. Before doing this, however, certain concepts from convex analysis and exponential families, already introduced in chapter 1, need to be treated formally.

2.1.2.1.1 Exponential Family Representations and Conjugate Duality

The description presented here follows the exposition and notation in [7]. Let a random vector $\mathbf{x} = \{x_s | s = 1, \ldots, n\}$ be defined in the Cartesian space χ^n. The random vector $\mathbf{x}$ collects the random variables x_s represented as nodes in a MRF. An exponential family is a particular class of densities taken with respect to the base measure v (counting measure for discrete or Lebesgue measure for continuous) defined by $\mathrm{d}v = h(\mathbf{x})\mathrm{d}\mathbf{x}$ for some arbitrary function $h : \chi^n \to \mathbb{R}^+$ and $\mathrm{d}\mathbf{x} = \prod \mathrm{d}x_s$. This collection of densities can be represented by:

$$p(\mathbf{x}; \theta) = \exp\left[\langle \theta, \phi(\mathbf{x}) \rangle - A(\theta)\right]$$

where $\phi(\mathbf{x})$ is a group of vector-valued functions known as potentials (or sufficient statistic) which maps the space $\chi^n \to \mathbb{R}^+$. The vector θ is an exponential set of

parameters, and $A(\theta)$ is the log-partition function. The log-partition function is given by the integral:

$$A(\theta) = \log \int_{\mathcal{X}^n} \exp\langle \theta, \phi(\mathbf{x}) \rangle v(\mathrm{d}\mathbf{x})$$

The parameter vector θ indexes a particular distribution within the family of $p(\mathbf{x}; \theta)$. The exponential parameter vector is part of the set:

$$\Theta := \{\theta \in \mathbb{R}^d \mid A(\theta) < \infty\}$$

Depending on how the Θ set is defined, the exponential distributions so defined belong to one of three groups:

1. Regular - Θ is an open set.
2. Minimal - unique parameter vector Θ associated with each distribution.
3. Overcomplete - affine subset of parameter vector Θ for each distribution.

A different (convex) set M is defined as:

$$M := \{\mu \in \mathbb{R}^d \mid \exists\, p(\cdot) \text{ such that } \int \phi(\mathbf{x}) p(\mathbf{x}; \theta) v(\mathrm{d}\mathbf{x}) = \mu\}$$

where μ is a vector composed of the expectations of the potentials $\phi(\mathbf{x})$. These

expectations correspond to the mapping $\Theta \to M$ defined below:

$$E[\phi(\mathbf{x})] = \int_{\chi^n} \phi(\mathbf{x}) p(\mathbf{x};\theta) v(\mathrm{d}\mathbf{x})$$

This mapping is one-to-one as long as the exponential representation is minimal and the expectations μ belong to the relative interior of the set M. Hence, the entropy of the density $p(\mathbf{x};\theta)$ is given by:

$$H(p(\mathbf{x};\theta)) = -\int_{\chi^n} p(\mathbf{x};\theta) \log p(\mathbf{x};\theta) v(\mathrm{d}\mathbf{x}) = -E_\theta[\log p(\mathbf{x};\theta)]$$

A remarkable result is that as long as μ belongs to the relative interior of M, then the dual conjugate of $A(\theta)$ is equal to the negative entropy of $p(\mathbf{x};\theta)$ shown above [7]. The dual conjugate (Fenchel-Legendre) of $A(\theta)$ comes out to be:

$$A^*(\mu) := sup_{\theta \in \Theta}\{\langle \mu, \theta \rangle - A(\theta)\}$$

which means the conjugate of A is the supremum of the expression inside the brackets that includes A itself (thus a dual conjugate). This also implies that the log-partition function $A(\theta)$ is a convex function of θ and in particular its representation belongs to the minimal exponential family. As such, for all $\theta \in \Theta$ the supremum is uniquely achieved when:

$$\mu = E_\theta[\phi(\mathbf{x})]$$

Simply put, the approximation of the variational entropy could be achieved (at

least in theory) as long as the set of mean parameters μ falls in the relative interior of the set M [7]. For discrete random vectors, the set M is a convex hull containing a finite set of $\phi(\mathbf{x})$ vectors associated with a graph G and is commonly referred to as the marginal polytope $MARG(G)$ [14].

The main challenges in dealing with the optimization expressions presented above are related to the nature of both the $MARG(G)$ and the dual conjugate A^*. First, the size of $MARG(G)$ grows quickly with the number of nodes in the underlying graph, thus making the problem intractable. Also, the dual conjugate A^* is itself a variational expression and typically lacks a closed-form solution. Hence, the problem of computing a closed-form solution could potentially be just as complicated as the original problem at hand (that of approximating the variational entropy). Once again, these problems are usually circumvented by making certain simplifying assertions about the structure of both $MARG(G)$ and A^*.

For acyclic graphs, the dual conjugate A^* has an explicit form composed of the sum of the following two terms:

$$H_s(x_s) := -\sum_{x_s} \mu_s(x_s) \log \mu_s(x_s)$$

$$I_{st}(\mu_{st}) := \sum_{x_s, x_t} \mu_{st}(x_s, x_t) \log \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}$$

Returning now to the log-partition function equation introduced in chapter 1,

$$A(\mu) = H_{Bethe} = \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} I_{st}(\mu_{st})$$

it assumes an acyclic graph (tree) where the $H_s(\mu_s)$ terms represent the singleton entropies and $I_{st}(\mu_{st})$ are the edgewise mutual information terms [7].

The Bethe approximation assumes that the log-partition function above applies to graphs with cycles and that it is well defined for any $\mu \in MARG(G)$. As noted earlier, defining the marginal polytope structure is quite a challenge. The Bethe approximation circumvents this difficulty by defining a set of necessary constraints for the marginals μ [14]. These constraints are exact for acyclic graphs and are summarized in the expression below:

$$MARG(G) = LOCAL(G) = \left\{ \tau \geq 0 \quad \middle| \quad \sum_{x_s} \tau_s(x_s) = 1 \quad \middle| \quad \sum_{x_{st}} \tau_{st}(x_{st}) = \tau_t(x_t) \right\}$$

where τ_s and τ_{st} are known as pseudo-marginals. For cyclic graphs, the Bethe approximation asserts that the true marginal polytope is approximated by a convex outer bound defined by the local consistency equations above. In other words, a candidate marginal τ may belong to $LOCAL(G)$, but not necessarily to $MARG(G)$. Hence, the expression for the Bethe variational problem is:

$$\max_{\tau \in LOCAL(G)} \left\{ \langle \theta, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st}) \right\}$$

It is well-known that the sum-product algorithm (BP) updates yield the stationary points of the optimization expression above [7]. Nonetheless, it is also known that (except for trees) the sum-product algorithm can lead to globally inconsistent marginals. This phenomenon manifests as the belief propagation fixed points falling

into local minima instead of converging to the unique global minimum.

Based upon the formulation of the Bethe variational problem above the Lagrangian equation then becomes:

$$L(\tau,\lambda) = \langle\theta,\tau\rangle + \sum_{s\,\in\,V\setminus\varepsilon} H_s(\tau_s) - \sum_{(s,t)\,\in\,E} I_{st}(\tau_{st}) + \sum_{(s,t)\,\in\,E}\left[\sum_{x_s}\lambda_{ts}(x_s)C_{ts}(x_s) + \sum_{x_t}\lambda_{st}(x_t)C_{st}(x_t)\right]$$

where the inner product term is specifically defined as:

$$\langle\theta,\tau\rangle = \theta_s(x_s)\tau_s(x_s) + \theta_t(x_t)\tau_t(x_t) + \theta_{st}(x_s,x_t)\tau_{st}(x_s,x_t)$$

and the consistency constraints are defined below:

$$C_{ts}(x_s) = \tau_s(x_s) - \sum_{x_t}\tau_{st}(x_s,x_t) = 0$$
$$C_{st}(x_t) = \tau_t(x_t) - \sum_{x_s}\tau_{ts}(x_s,x_t) = 0$$

The Lagrangian multiplier λ has the following definition:

$$\lambda_{ts}(x_s) = \prod_{t\,\in\,N(s)} M_{ts}(x_s)$$

where $M_{ts}(x_s)$ is the collection of messages impinging on node s. After taking the derivatives of the Lagrangian equation with respect to both τ and λ, and substituting

the given multiplier definition the two expressions below are obtained:

$$\tau_s(x_s) = \kappa_1 \exp(\theta_s(x_s)) \prod_{t \in N(s)} M_{ts}(x_s)$$

$$\tau_{st}(x_s, x_t) = \kappa_2 \exp(\theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t)) \prod_{u \in N(s) \setminus t} M_{us}(x_s) \prod_{u \in N(t) \setminus s} M_{ut}(x_t)$$

where the κ_1 and κ_2 are proportionality constants to ensure that the normalization constraint is met. Once the consistency constraints are enforced by substitution in the expressions above the sum-product (BP) update equation appears as:

$$M_{ts}(x_s) = \kappa \sum_{x_t} \exp(\theta_{st}(x_s, x_t) + \theta_t(x_t)) \prod_{u \in N(t) \setminus s} M_{ut}(x_t)$$

A considerable amount of effort has gone into finding tighter bounds on $MARG(G)$ and/or better approximations to $A(\theta)$ [14, 21, 22, 23]. Most of these methods trade finer approximations at the expense of increased complexity.

2.1.2.2 Information-Geometric Interpretation of Bethe Approximations

There is yet a different interpretation of the Bethe free energy approximation based upon the concept of information geometry. Interestingly, it ultimately arrives at the same BP update equations previously derived in sections 2.1.2 and 2.1.2.1. The necessary information geometry tools are presented next prior to delving into the derivation details of the BP fixed points. This exposition is largely based on [24, 25, 26].

2.1.2.2.1 Log-Coordinates, Product Distributions, and Projections

Let d_a denote the degree (i.e. the number of connected variable nodes x_a) of factor node f_a in a regular factor graph. Also let b_i represent a vector showing a single outcome $(\mathrm{i} - th)$ out of the 2^{d_a} possible outcomes of f_a assuming that the variables are defined in the $GF(2)$ domain. The $2^{d_a} \times d_a$ matrix is obtained after stacking the vectors containing all possible outcomes.

$$B_a = \begin{bmatrix} b_0 \\ \vdots \\ b_{2^{d_a}-1} \end{bmatrix}$$

Each one of these outcomes has a corresponding probability of occurrence $P(\mathbf{x}_a = b_j) = p$ which is a probability mass function. This set of probabilities are collected into a vector $\mathbf{p} = [p_0 \ \cdots \ p_{2^{d_a}-1}]^T$. Hence, the set of marginal probabilities m can be obtained from:

$$m = B_a^T \mathbf{p}$$

The negative entropy is defined as:

$$h(\mathbf{p}) = \sum_{i=0}^{2^{d_a}-1} p_i \log p_i$$

Given that $\mathbf{p}$ is a probability mass function compliant with probability axioms, the

entropy expression above can be re-written as:

$$h(\mathbf{p}) = \left(1 - \sum_{i=1}^{2^{d_a}-1} p_i\right) \log \left(1 - \sum_{i=1}^{2^{d_a}-1} p_i\right) + \sum_{i=1}^{2^{d_a}-1} p_i \log p_i$$

The expression above yields the following set of derivatives [24]:

$$\frac{\partial h(\mathbf{p})}{\partial p_i} = \log \frac{p_i}{p_0} \quad \text{for} \quad i = 0, \ldots, 2^{d_a} - 1$$

which in turn allows for the introduction of a logarithmic coordinate system:

$$\theta_i = \log \frac{p_i}{p_0} \quad \text{for} \quad i = 0, \ldots, 2^{d_a} - 1$$

Note that θ_0 is always 0. An important observation is that the probability mass function p can be obtained by [26]:

$$p_i = \exp(\theta_i - A(\boldsymbol{\theta})) \quad \text{for} \quad i = 1, \ldots, 2^{d_a} - 1$$

So $\mathbf{p}$ belongs to the family of exponential function representations (as shown in section 2.1.2.1.1) and $A(\boldsymbol{\theta})$ is the log-partition function:

$$A(\boldsymbol{\theta}) \triangleq \log \left(\sum_{i=0}^{2^{d_a}-1} \exp(\theta_i)\right)$$

Another interesting fact is that the set of derivatives of $h(\mathbf{p})$ map the distribution $\mathbf{p}$ onto the set of $\boldsymbol{\theta}$, whereas the set of derivatives of $A(\boldsymbol{\theta})$ maps $\boldsymbol{\theta}$ back to $\mathbf{p}$. The direct implication is that $A(\boldsymbol{\theta})$ and $h(\mathbf{p})$ form a convex conjugate pair (Legendre pair) such

that [25, 26]:

$$A(\boldsymbol{\theta}) + h(\mathbf{p}) = \langle \boldsymbol{\theta}, \mathbf{p} \rangle$$

For an arbitrary probability mass distribution t, the conjugate pair expression above becomes then:

$$A(\boldsymbol{\theta}) + h(\tau) = \langle \boldsymbol{\theta}, \tau \rangle + D(\mathbf{p} \parallel \mathbf{t})$$

If the probability mass distribution $\mathbf{t}$ is defined as the product of its marginals, then it is called a product distribution [24]:

$$t(x_a) = \prod_{i \in N(a)} t_i(x_i)$$

whereas the log-coordinates τ of the expression above are given by:

$$\tau(x_a) = \sum_{i=1}^{d_a} \log \frac{t_i(x_i)}{t_i(0)} = \sum_{i:x_i=1} \log \frac{t_i(1)}{t_i(0)} = B^T \boldsymbol{\lambda}$$

where $\boldsymbol{\lambda}$ contains the log-marginal ratios.

A well-known result is that the probability mass fuction $\mathbf{t}$ obtained from the product of the marginals of $\mathbf{p}$ is the closest product distribution of $\mathbf{p}$ by the Kullback-Leibler divergence (or distance) [26]:

$$\mathbf{t} = \arg\min_{t \in P_a} D(\mathbf{p} \| \mathbf{s})$$

for some arbitrary product distribution $\mathbf{s}$ belonging to the set $_a$ of all product distribu-

tions defined for $\mathbf{x}_a$ (variables connected to node f_a). Since $\mathbf{t}$ minimizes this distance then it gives the best approximation to the true set of marginals m of distribution $\mathbf{p}$. This is stated by the following equation in log-coordinates:

$$\lambda = \log B^T p - \log(1 - B^T p) = \pi(\cdot)$$

where the operator $\pi(\cdot)$ can be seen asan information projection.

Returning to the Bethe approximation, the free energy expression is now given by:

$$F_{Bethe} = \sum_{a=1}^{M} \left(\sum_{j=0}^{2^{d_a}-1} p_{a,j} \log p_{a,j} - \langle p_a, \phi_a \rangle \right) - \sum_{i=1}^{N} (d_i - 1)[t_i \log t_i + (1 - t_i) \log(1 - t_i)]$$

where the inner product summation is the average Bethe energy and the rest of the terms combined give the variational Bethe entropy [24]. Note that the inner product argument ϕ_a is the factor node f_a in log-coordinates and $\mathbf{p}_a$ is the probability mass function across all outcomes out of f_a as defined previously. Also, the t_i is the marginal at variable node i. The consistency constraint is given by:

$$B_a^T \mathbf{p}_a = \mathbf{t}_a \quad \text{for} \quad a = 1, \ldots, M$$

where B_a is also defined as before. The Lagrangian equation appears now as:

$$\begin{aligned}\mathcal{L}_{Bethe} &= \sum_{a=1}^{M}\left(\sum_{j-0}^{2^{d_a}-1} p_{a,j}\log p_{a,j} - \langle p_a, \phi_a\rangle\right) \\ &\quad - \sum_{i=1}^{N}(d_i-1)[t_i\log t_i + (1-t_i)\log(1-t_i)] + \sum_{a=1}^{M}\langle t_a - B_a^T p_a, \lambda_a\rangle \\ &= \sum_{a=1}^{M}\left(\sum_{j=0}^{2^{d_a}-1} p_{a,j}\log p_{a,j} - \langle p_a, B_a^T\lambda_a + \phi_a\rangle\right) \\ &\quad - \sum_{i=1}^{N}(d_i-1)[t_i\log t_i + (1-t_i)\log(1-t_i)] + \sum_{a=1}^{M}\langle t_a, \lambda_a\rangle\end{aligned}$$

where λ_a is the vector of Lagrange multipliers. The following log-coordinate expressions are obtained after taking the partial derivatives with respect to both $\mathbf{p}_a$ and t_i and setting them to zero:

$$\tau_i^* = \frac{1}{d_i - 1}\sum_{a\,\in\,N(i)}\lambda_{i\to a} \qquad \theta_a^* = B_a\lambda_a + \phi_a$$

where τ_i is the log-coordinate of the (belief) marginal at variable node i and $\boldsymbol{\theta}_a$ is the log-coordinate vector representation of the marginal distributions of $\mathbf{p}_a$. Note that in the expressions above the asterisks distinguish the actual critical points that effectively null out the partial derivatives based on the pseudo-dual function of the Lagrangian instead of the Lagrangian itself [24]. The $\lambda_{i\to a}$ can be seen as the messages collected from all nodes surrounding node i. Thus, the formulation on the left above represents one of the BP updates at variable node i in log-coordinates. On the other hand, the equation on the right represents the projection or mapping of the distribution $\mathbf{p}_a$ onto the set of product distributions $\boldsymbol{\theta}_a$ and constitutes the other

BP update equation from factor node f_a back to its surrounding variable nodes. The actual stationary points are the marginal consistency constraints obtained from substituting the critical points back into the pseudo-dual Lagrangian expression.

2.2 Survey Propagation

Survey propagation (SP) is a message-passing technique similar to belief propagation proposed to address satisfiability problems. It was originally developed in the context of the statistical physics of disordered systems [27, 28]. These physical systems have very close ties to constraint satisfaction problems. Constraint satisfaction is a classic problem in combinatorial optimization.

These problems generally consist of N Boolean variables and M constraints. Each constraint (or clause) is connected to a subset of the N Boolean variables. The objective is to find an assignment of variables that simultaneously satisfy all the constraints. Satisfiability problems can also be represented by factor graphs as shown in Figure 2.2. Most of the attention in combinatorial optimization has focused on the random K-SAT problem. For K $>$ 2, this problem has been found to be NP complete [27]. The K-SAT problem is described in much the same way as the general constraint satisfaction problem except that the clauses are OR functions of K randomly chosen variables. The statistical physics literature suggests the existence of a phase diagram describing the entire space of variable assignments [28]. The ratio $\alpha = M/N$ determines the various phases of the diagram. The critical parameter α_c

represents the threshold that splits the region into a SAT phase and an UNSAT phase. The SAT phase contains all the satisfying variable assignments while the UNSAT phase has no satisfying assignments. It has also been conjectured that the SAT phase can be further segregated into an easy SAT portion and a hard SAT portion by a parameter α_d [28]. The threshold α_c has only been found for K = 2. It has been conjectured via non-rigorous methods that $\alpha_c \approx 4.267$ and $\alpha_d \approx 3.921$ for K = 3.

Virtually any heuristic search algorithm is able to converge on a satisfying assignment in the easy SAT phase. Most of the interest in the K-SAT problem arises from the fact that very difficult instances are generated around the threshold α_c (SAT/UNSAT boundary) and conventional heuristic algorithms often fail to find those instances.

There have been attempts to explain the reasons behind the apparent difficulty in finding a solution in the hard SAT part. Statistical physicists have proposed the concept of clusters [27]. In the easy SAT part, the satisfying assignments form a single cluster in the phase diagram. This means that solutions only differ from one another by a finite number of positions. On the other hand, in the hard SAT part, the single cluster of solutions breaks up into multiple clusters that are often widely separated. This phenomenon is known in the physics literature as 1-step replica symmetry breakup (1-RSB) [28]. Therefore, a local heuristic such as belief propagation that falls in a cluster not containing the global valid assignment would struggle to converge or yield a reasonable solution. Figure 2.3 shows this phenomenon.

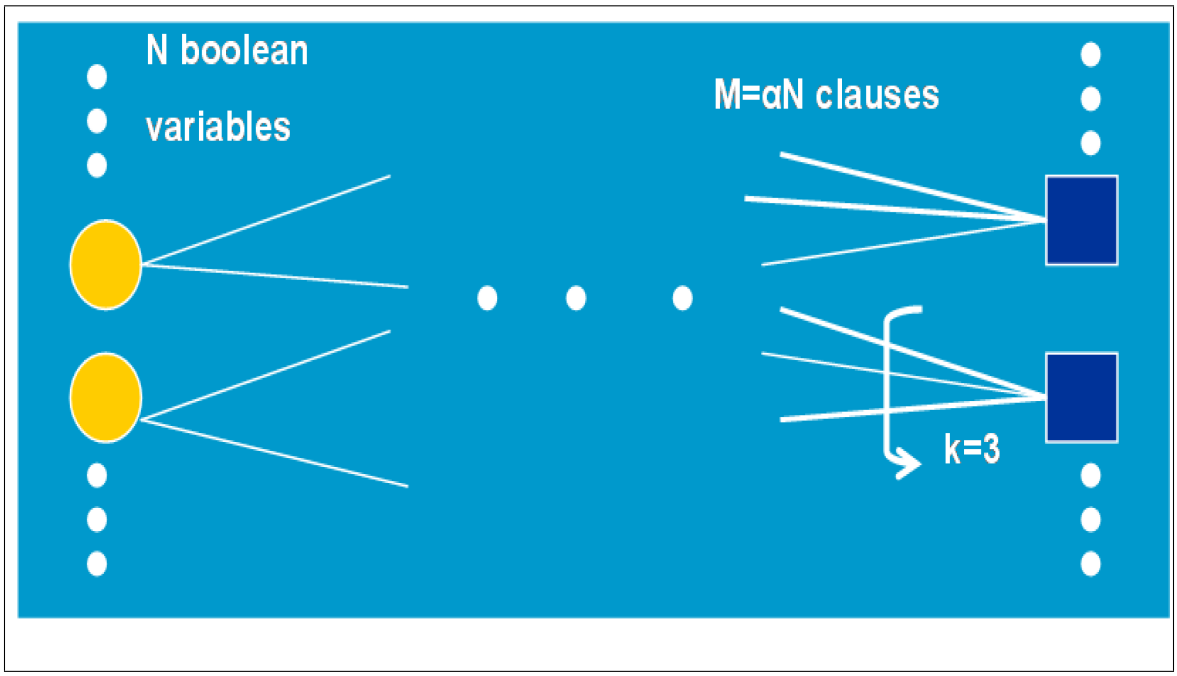

Figure 2.2: Factor Graph Representation of the K-SAT Problem

The K-SAT problem in the difficult hard SAT region manifests as a factor graph with cycles. The survey propagation algorithm has shown better empirical performance in these types of problems than the standard belief propagation algorithm [27, 28]. However, survey propagation is also a heuristic whose convergence is not guaranteed either. The messages being passed along the graph are surveys of the clusters in the SAT region.

In a factor graph, the messages called surveys are real numbers defined between zero and one. If the survey from clause a to node i is named $\eta_{a\to i}$ then [27]:

$$\eta_{a\to i} = \prod_{j \in V(a)\setminus i} \frac{\gamma^u_{j\to a}}{\gamma^u_{j\to a} + \gamma^s_{j\to a} + \gamma^0_{j\to a}}$$

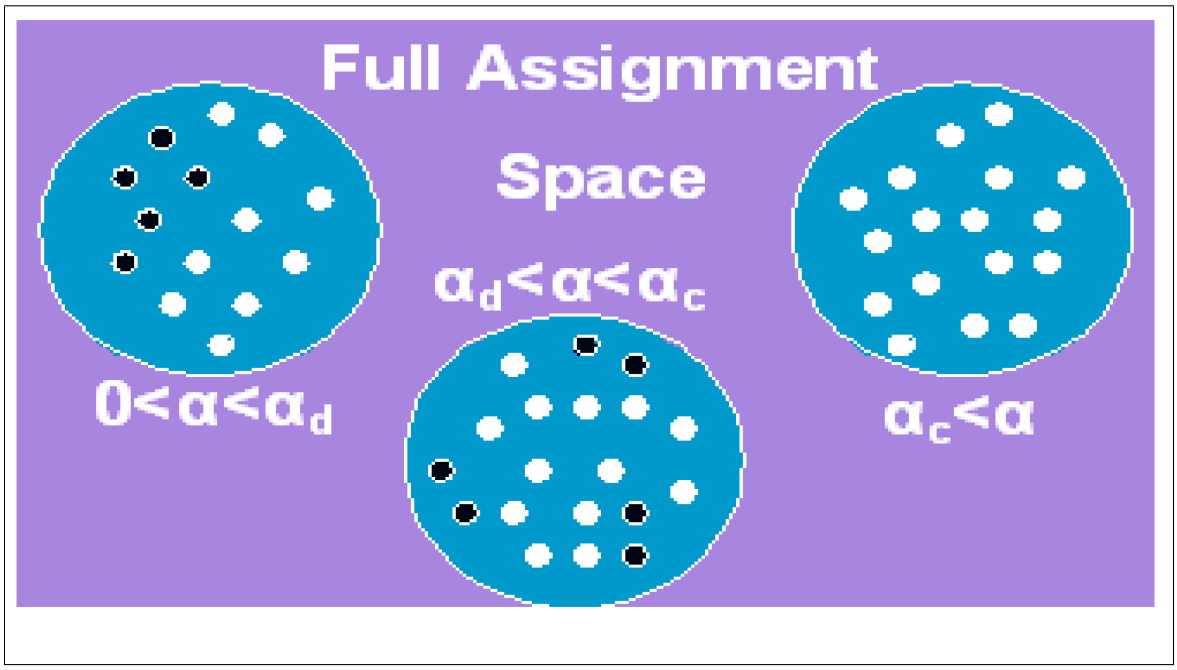

Figure 2.3: Factor Graph Representation of the K-SAT Problem

where the three real quantities in the denominator are given by:

$$\gamma^u_{j\to a} = \left[1 - \prod_{b\in V^u(a)\backslash j}(1-\eta_{b\to j})\right] \prod_{b\in V^s(a)\backslash j}(1-\eta_{b\to j})$$

$$\gamma^s_{j\to a} = \left[1 - \prod_{b\in V^s(a)\backslash j}(1-\eta_{b\to j})\right] \prod_{b\in V^u(a)\backslash j}(1-\eta_{b\to j})$$

$$\gamma^0_{j\to a} = \prod_{b\in V(a)\backslash j}(1-\eta_{b\to j})$$

where $V(a)\backslash j$ means all the nodes connected to clause a except for j and the superscripts s and u represent the subset of $V(\cdot)$ that either satisfies or does not satisfy the constraint. The fixed points $\eta^*_{a\to i}$ can be used in a decimation process in order to find the satisfying assignment. This decimation procedure is the second step in the survey propagation algorithm and consists of eliminating the clauses that were

satisfied during the last iteration of SP [28]. The first step propagates the messages along the graph much like belief propagation would. After decimation is performed the messages are propagated again along the remaining clauses in the graph. After convergence, the third step is to use any local heuristic search in order to find the full valid assignment.

The ability of survey propagation to find valid assignments more efficiently than belief propagation for hard K-SAT formulas has inspired applications beyond the realms of statistical physics and computer science [1]. Nevertheless, the vast majority of the evidence produced to demonstrate the performance of survey propagation is empirical and the theoretical arguments that explain its effectiveness remain largely elusive.

2.2.1 Connections to Belief Propagation

A remarkable link between SP and BP has been found recently [29]. This relationship establishes that SP is nothing more than an instantiation of BP over an extended MRF. The extended MRF is an expansion of the definition of the MRF underlying a graph representing the random K-SAT instance. Furthermore, this modified definition of the MRF gives rise to a new family of generalized SP(ρ) algorithms where $\rho = 1$ yields the original SP algorithm. Thus, the SP(ρ) updates are defined by the equations

below [29]:

Message from clause a to variable t :

$$\eta_{a\to i} = \prod_{j \in V(a)\backslash i} \frac{\gamma^u_{j\to a}}{\gamma^u_{j\to a} + \gamma^s_{j\to a} + \gamma^0_{j\to a}}$$

Message from clause j to variable a :

$$\gamma^u_{j\to a} = \left[1 - \rho \prod_{b\in V^u(a)\backslash j} (1-\eta_{b\to j})\right] \prod_{b\in V^s(a)\backslash j} (1-\eta_{b\to j})$$

$$\gamma^s_{j\to a} = \left[1 - \prod_{b\in V^s(a)\backslash j} (1-\eta_{b\to j})\right] \prod_{b\in V^u(a)\backslash j} (1-\eta_{b\to j})$$

$$\gamma^0_{j\to a} = \prod_{b\in V(a)\backslash j} (1-\eta_{b\to j})$$

where $\rho \in [0,1]$. The marginals at each variable are computed as follows:

$$\mu_j(1) \propto \left[1 - \rho \prod_{b\in C^+(j)} (1-\eta_{b\to j})\right] \prod_{b\in C^-(j)} (1-\eta_{b\to j})$$

$$\mu_j(0) \propto \left[1 - \rho \prod_{b\in C^-(j)} (1-\eta_{b\to j})\right] \prod_{b\in C^+(j)} (1-\eta_{b\to j})$$

$$\mu_j(*) \propto \prod_{b\in C^+(j)} (1-\eta_{b\to j}) \prod_{b\in C^-(j)} (1-\eta_{b\to j})$$

The basis for this new parameter ρ was originated by extending the MRF to deal with partial satisfiability assignments typically associated with any K-SAT formula. Thus, the two key insights are that SP behaves just like BP over a modified MRF and that a whole family of message passing algorithms could be developed to

tackle hard K-SAT problems.

2.2.1.1 Extended Markov Random Field

The SP(ρ) family of algorithms are attached to special-purpose MRFs with partially defined sets of variables (assignments) in their domain space [29]. A partial assignment basically means allowing the variables $X = \{X_1, \ldots, X_N\}$ to take on values in the set $\{0, 1, *\}$ where the symbol $*$ represents free variables allowed to take on either 0 or 1. As such, the validity of a partial assignment X to a clause labeled a is dictated by the following two rules:

1. The assignment is invalid if all variables X are unsatisfied.
2. The assignment is invalid if all variables are unsatisfying except for one.

The validity of a partial assignment in clause a is denoted by $VAL_a(X_{v(a)})$ following the notation in [29]. A variable node X_i is constrained by clause a if it is the uniquely satisfying variable in this clause. This condition is indicated by $CON_{i,a}(X_{v(a)})$. This constraint condition defines three distinct sets:

1. $S_*(X) = \{i \in V : X_i = *\}$
2. $S_c(X) = \{i \in V : X_i \text{ constrained}\}$
3. $S_o(X) = \{i \in V : X_i \text{ unconstrained}\}$

The size of the sets above is given by the variables $n_*(X), n_c(X), \text{and } n_o(X)$ respectively. Several probability distributions could be defined by providing weights to the various variables belonging to each of these sets. These weights are restricted to the interval $[0, 1]$ and are denoted by $\omega_*(X), \omega_c(X), \text{and } \omega_o(X)$ respectively. Lastly, the parent set of variable X_i, labeled P_i , constitutes the set of clauses for which X_i is the uniquely satisfying variable. Consequently, the following conditions apply to this set:

1. If$X_i = 0, \text{then } P_i \subseteq C^-(i)$
2. If$X_i = 1, \text{then } P_i \subseteq C^+(i)$
3. When$X_i = *, \text{then } P_i = \varnothing$

Hence, the extended MRF can be formulated as [29]:

$$p_{EMRF}(X, P) = \prod_{i \in V} (\Psi_i(X_i, P_i) \prod_{a \in C} \Psi_a(X_{V_a}, P_{V(a)})$$

defined on the Cartesian space $\chi_1 \times \chi_2 \times \cdots \times \chi_n$ where $\chi_i = \{0, 1, *\} \times P_i$. The joint probability distribution above is composed of the product of both variable and

clause compatibility functions ($\Psi_i$and Ψ_a) defined below:

$$\Psi_i(x_i, P_i) := \begin{cases} \omega_o \ : \ P_i = \varnothing, x_i \neq * \\ \omega_* \ : \ P_i = \varnothing, x_i = * \\ 1 \ : \ \text{for any other valid } (x_i, P_i) \end{cases}$$

$$\Psi_a(x_{V(a)}, P_{V(a)}) := VAL_a(x_{V(a)}) \prod_{i \in V(a)} \delta(\text{Ind}[a \ \in \ P_i], CON_{a,i}(x_{V(a)}))$$

where the variable compatibility function Ψ_i assigns the proper weighting to the partial assignments according to the number of unconstrained and free variables. The clause compatibility function Ψ_a is used to ensure that the partial assignments are valid and that the parent sets are consistent with the assignments in the neighborhood of clause a.

2.2.1.2 Belief Propagation Recursions over the Extended Markov Random Field

The BP recursions over the expanded MRF involve passing messages among two sets of triplets: $(M^s_{a\to i}, M^u_{a\to i}, M^*_{a\to i})$and $(R^s_{i\to a}, R^u_{i\to a}, R^*_{i\to a})$ [29]. The first set of triplets are messages from clause a to variable i. The second set constitutes the messages from the variable node i to clause a. The actual recursion equations are displayed

below:

$$
\begin{aligned}
M^s_{a\to i} &= \prod_{j\,\in\, V(a)\setminus i} R^s_{j\to a} \\
M^u_{a\to i} &= \prod_{j\in V(a)\setminus i} (R^u_{j\to a} + R^*_{j\to a}) + \sum_{k\in V(a)\setminus i} (R^s_{k\to a} - R^*_{k\to a}) \prod_{j\in V(a)\setminus i,k} R^u_{j\to a} \\
&\quad - \prod_{j\in V(a)\setminus i} R^u_{j\to a} \\
M^*_{a\to i} &= \prod_{j\in V(a)\setminus i} (R^u_{j\to a} + R^*_{j\to a}) - \prod_{j\in V(a)\setminus i} R^u_{j\to a} \\
R^s_{i\to a} &= \prod_{b\in C^u_a(i)} M^u_{b\to i} \left[\prod_{b\in C^s_a(i)} (R^s_{b\to i} + R^*_{b\to i}) \right] \\
R^u_{i\to a} &= \prod_{b\in C^s_a(i)} M^u_{b\to i} \left[\prod_{b\in C^u_a(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o) \prod_{b\in C^u_a(i)} M^*_{b\to i} \right] \\
R^*_{i\to a} &= \prod_{b\in C^u_a(i)} M^u_{b\to i} \left[\prod_{b\in C^s_a(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o) \prod_{b\in C^s_a(i)} M^*_{b\to i} \right] \\
&\quad + \omega_* \prod_{b\in C^s_a(i)\cup C^u_a(i)} M^*_{b\to i}
\end{aligned}
$$

where the marginals at any point during the iteration are given by:

$$
\begin{aligned}
p_i(0) &\propto \prod_{b\in C^+(i)} M^u_{b\to i} \left[\prod_{b\in C^-(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o) \prod_{b\in C^-(i)} M^*_{b\to i} \right] \\
p_i(1) &\propto \prod_{b\in C^-(i)} M^u_{b\to i} \left[\prod_{b\in C^+(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o) \prod_{b\in C^+(i)} M^*_{b\to i} \right] \\
p_i(*) &\propto \omega_* \prod_{b\in C(i)} M^*_{b\to i}
\end{aligned}
$$

Thus, the BP updates on the extended MRF are equivalent to the generalized SP(ω_*) algorithms if $\omega_o + \omega_* = 1$ given $\omega_* \in [0, 1]$ and that $M^u_{a\to i}$ is initialized to $M^*_{a\to i}$ [29]. This last assertion opens the door for SP to be applied to more general statistical inference problems such as source coding since it can be treated like any other type of message passing algorithms, yet it appears to be more suitable than other schemes in dealing with factor graphs with cycles.

2.2.2 Alternate Survey Propagation Interpretation

The importance of the MRF formalism introduced in section 2.2.1 along with its connection to BP cannot be overstated. The unifying framework has provided key insight into the SP algorithm and the structure of the underlying satisfiability problem. Nevertheless, a more flexible interpretation of SP has surfaced recently and will be described next.

A different formulation of the K-SAT problem has been presented based on the concept of normal realizations (Forney graphs) via generalized state variables [30]. First, under this MRF construction the variable sets are extended to what is known as power sets. Secondly, the generalized state variables are denoted as left states and right states on the graph depending on the direction of the messages. Thirdly, a state de-coupling condition is applied. The resulting MRF yields a family of SP algorithms whose update rules are akin to BP. Therefore, the SP algorithm is a special case of BP only under the set of conditions stated above. Additional details are found in [30].

On the other hand, it is important to note that the MRF formalism presented in section 2.2.1 is limited to random K-SAT problems. The idea of generalizing this particular MRF concept to other types of constraint satisfaction problems (i.e. graph coloring, etc.) might lead to contradictory conclusions. More specifically, that it is not possible to reduce the BP algorithm to SP. Nonetheless, the MRF structure based on normal realizations is more amenable to general constraint problems (i.e. satisfiability problems with arbitrary finite variable alphabets and clauses of arbitrary form) but inevitably leads to the notion that for general constraint satisfaction problems SP is not an instantiation of BP [31]. Fortunately, many relevant source coding problems are equivalent to random instantiations of the K-SAT problem represented in a factor graph.

2.3 Other Approximate Inference Algorithms

A fairly large number of algorithms have been proposed to address the challenge imposed by approximate statistical inference methods. The vast majority of the proposed algorithms in the literature share a common thread which is that they could be considered special cases of the two main families of message-passing algorithms already discussed, namely BP and SP. Aside from the fact that they are important in their own right, some of the most relevant schemes in the context of source and channel coding to be highlighted next are also important in the sense that they provide additional insight and context to the new developments to be unveiled subsequently.

2.3.1 Thoules-Anderson-Palmer Algorithm

The Thoules-Anderson-Palmer (TAP) algorithm is a lossy compression scheme developed using a particular statistical physics approach to build a dynamic model based on the Markov process assumption on prior beliefs [32]. More specifically, mean field behavior is assumed for statistical dependencies between codeword and source sequences and a first-order time dependence between current and previous spin glass states. The encoding step entails translating the Boolean alphabets from the binary source into Ising alphabets (more natural in statistical physics) and running the two update equations (with suitably chosen $\alpha, \beta,$ and γ parameters) presented below recursively until convergence is attained.

$$\hat{m}_{ui'}(t+1) = \tanh(\beta J_u) \prod_{i' \in L(u) \setminus i} m_{ui'}(t)$$

$$m_{ui}(t+1) = \tanh\left(\sum_{u' \in M(i) \setminus u} \tanh^{-1}[\hat{m}_{u'i}(t)] + \alpha + \tanh^{-1}[\gamma m_i(t)] \right)$$

Then the pseudo-posterior marginals are given by:

$$m_i(t) = \tanh\left(\sum_{u \in M(i)} \tanh^{-1}[\hat{m}_{ui}(t)] + \alpha + \tanh^{-1}[\gamma m_i(t)] \right)$$

and the optimal (in the Bayes sense) encoding is finally obtained from $\hat{m}_i(t) = \text{sgn}(m_i(t))$ in the Ising representation [32]. The algorithm applies the inverse to the decoding problem described by Sourlas in [33]. The published results indicate

the effectiveness of TAP in approaching the rate-distortion theoretical bounds for relatively long block lengths.

2.3.2 Bias Propagation

This algorithm is essentially a straightforward BP procedure with the added step of decimation [34]. It performs bit-wise maximum a-posteriori estimation in order to find a binary vector that represents a source sequence according to a distortion measure. The bias messages $B_{i\to a}$ (difference in marginals) and the source bit messages $B_{s_a\to a}$ are sent to the check nodes. The check nodes return satisfaction messages $S_{a\to i}$. These iterative terms are defined by the following update equations [34]:

$$B_{i\to a} = \frac{1 - R_{i\to a}}{1 + R_{i\to a}}$$
$$S_{a\to i} = \prod_{j\in\bar{V}(a)\setminus\{i\}} B_{j\to a}$$

where $R_{i\to a} = M_{i\to a}(1)/M_{i\to a}(0) = \prod_{b\in C(i)\setminus\{a\}} R_{b\to i}$ and $M_{i\to a}$ is the message from node i to check node a. The source bit messages are given by:

$$B_{s_a\to a} = (-1)^{s_a}\tanh(\gamma_a)$$

where the variable γ_a is part of a fixed vector γ which represents the strength of the check nodes. After the maximum number of iterations is reached, the final bias

equation is shown below:

$$B_i = \frac{1 - \prod_{b \in C(i)} R_{b \to i}}{1 + \prod_{b \in C(i)} R_{b \to i}}$$

The most likely bits (i.e. those with the largest bias) are fixed and removed from the graph. The process continues until all the information bits (binary vector) are fixed or until the maximum number of iterations is reached. The Bias Propagation (BiP) technique is very appealing for many source coding applications but the decimation step adds undesirable complexity. It is also worth mentioning that a connection between the BiP algorithm and the TAP algorithm has been found [34].

2.3.3 Fractional Belief Propagation

Fractional BP is a generalization of the standard BP that avoids resorting to large clusters (i.e. Kikuchi methods) by introducing a scale parameter c_α that attempts to capture the effects of cycles in factor graphs [22]. The fixed points are given by the expressions below:

$$\begin{aligned} Q_\alpha(X_\alpha) &\propto \psi_\alpha(X_\alpha)^{1/c_\alpha} \prod_{i \in N_\alpha} \prod_{\beta \in N_i \setminus \alpha} m_{\beta i}(X_i) m_{\alpha i}(X_i)^{1-1/c_\alpha} \\ Q_i(X_i) &\propto \prod_{\alpha} m_{\alpha i}(X_i) \\ m_{\alpha i} &= \frac{Q_\alpha(X_\alpha)}{Q_i(X_i)} m_{\alpha i}(X_i) \end{aligned}$$

where Q_a and Q_i are the marginal distributions on factor and variable nodes respectively and the $m_{\alpha i}$ constitute the updates. The scale parameter c_α is usually bounded

according to $0 \leq c_\alpha \leq 1$. The standard BP could be recovered by setting $c_\alpha = 1$. No methods are known to exist for optimally tuning this parameter for any given situation. Also, these update equations have only been shown to converge in simple channel decoding problems. The algorithms to be exposed in chapter 3 are geared specifically towards codeword quantization problems, which are known to be more difficult than traditional channel decoding problems.

2.3.4 Multilevel Belief Propagation

The multi-level or multi-grid BP algorithm has been developed to deal with large-scale graphs frequently encountered in data mining and computer vision applications in which standard BP appears to be notoriously slow. The graph of interest is coarsened to reduce its scale. Then the standard BP is run and a coarse result is obtained. The coarse results are refined back, level by level until the solution to the original problem is obtained.

The approach is closely related to the algebraic multi-grid technique and reduces run time without seemingly compromising the accuracy of the solutions [35]. The coarsening procedure iteratively selects nodes that strongly influences others and splits them into fine and coarse sets. Typically about half the nodes make it into the next iteration of the graph. The scale of the graph is reduced exponentially and can be accomplished in linear time with respect to the total size of the original graph [35].

2.3.5 Normalized and Offset Belief Propagation

Normalized and offset BP are two simple modifications to the standard BP algorithm which are believed to help in factor graphs with cycles. These modifications compensate for the over-estimation of the reliability of the messages in standard BP due to the presence of cycles in the factor graph. In normalized BP the message from a variable node v to factor node c is modified by a multiplicative correction factor [36]:

$$m'_{v\to c} = \alpha m_{v\to c}$$

whereas in offset BP the message going from variable node v to factor node c is modified instead by an additive correction factor [36]:

$$|m'_{v\to c}| = \begin{cases} |m_{v\to c}| - \beta, & m_{v\to c} > \beta \\ |m_{v\to c}|, & m_{v\to c} \leq \beta \end{cases}$$

Both correction factors are also constrained to values in the interval $[0, 1]$. Both algorithms appear to perform similarly in channel decoding cases and their complexity is very comparable to the original BP algorithm. Their performance in codeword quantization problems remains questionable.

2.3.6 Sequential Auxiliary Belief Propagation

This algorithmic variant combines BP with importance sampling (particle filter) techniques in a framework that extends the applicability of BP to non-linear, non-Gaussian graph models such that the computational complexity only increases linearly with the number of samples (particles) [37]. The structure of the optimal importance function for both messages and belief updates is given by:

$$q_{i,j}^{\text{opt}}(x_j|y_j)\frac{p_j(y_j|x_j)\prod_{k\in N(j)\setminus i}\hat{m}_{k,j}^{n-1}(x_j)}{\int p_j(y_j|x_j)\prod_{k\in N(j)\setminus i}\hat{m}_{k,j}^{n-1}(x_j)\mathrm{d}x_j}$$

The structure above is computationally intractable for dense graphs and large number of particles. Hence, an auxiliary variable is needed to circumvent this difficulty. This auxiliary variable represents a simpler importance function which is just a product of mixture distributions:

$$q(\theta_{1:K}) = \prod_{i=1}^{K} q_i(\theta_i)$$

which yields a computationally feasible approach to approximate the posterior distribution. An unscented approximation is used to sample the auxiliary variable and the state as detailed in [37]. Then the BP weights are updated and the steps are repeated. This method enables the study of temporally evolving graphs. It was developed and applied in the context of distributed fusion problems.

2.3.7 Residual Belief Propagation

Residual BP is essentially a serial dynamic scheduling method (as opposed to a flooding scheme) whereby residuals are calculated from the differences in messages before and after an update [38]. These residuals are then used to determine which messages to propagate first. The rationale is that the residuals decrease to zero as the BP algorithm converges. Therefore, the messages yielding the largest residuals should be prioritized. This has the net effect of yielding faster convergence times than the standard BP algorithm. The messages used to compute residuals could be either variable-to-check or check-to-variable. For any check node c_i connected to a variable node v_j the update equations are given by:

$$m_{v_j \to c_i} = \sum_{c_a \in N(v_j \backslash c_i)} m_{c_a \to v_j} + C_{v_j}$$

$$m_{c_i \to v_j} = 2 \operatorname{arctanh} \left(\prod_{v_b \in N(c_i \backslash v_j)} \tanh \left(\frac{m_{v_b \to c_i}}{2} \right) \right)$$

where $C_{v_j} = \log((p(y_j v_j) = 1)/(p(y_j|v_j) = 0))$ and y_j is the received signal. There are differences in terms of both performance and complexity depending on which set of messages is used to compute the residuals [39, 40]. Additional serial dynamic scheduling schemes for fast decoding of LDPC codes at high data rates have been developed and can be found in [41, 42, 43, 44, 45, 46].

2.3.8 Oscillation-based Belief Propagation

In the context of LDPC decoding, oscillations refer to the change in reliability of the bits from iteration to iteration. More specifically, an oscillation has occurred if the sign of the Log-Likelihood Ratio (LLR) changes from the previous iteration [47].

$$\text{sign}(z_{mn}^{t}) \neq \text{sign}(z_{mn}^{t-1})$$

$$\text{sign}(z_{mn}^{t}) = \begin{cases} 1, & z_{mn}^{t} > 0 \\ -1, & z_{mn}^{t} \leq 0 \end{cases}$$

where t is the iteration number and z_{mn} is the extrinsic LLR propagated from the n$-th$ variable node to the m$-th$ check node. The oscillations are mainly induced by the presence of cycles in the underlying graph. The oscillation-based BP seeks to overcome this obstacle by adding together the previous extrinsic LLR value from the bit node to the check node with the current one [47]:

$$z_{mn}^{t} = F_n + \sum_{m' \in M(n) \backslash m} L_{m'n}$$

$$z_{mn}^{'t} = \begin{cases} z_{mn}^{t} + z_{mn}^{'t-1}, & \text{sign}(z_{mn}^{t}) \neq \text{sign}(z_{mn}^{'t-1}) \\ z_{mn}^{t}, & \text{otherwise} \end{cases}$$

where $F_n = 2y_n/\sigma^2$ and L_{mn} is given by:

$$L_{mn} = 2 \tanh^{-1} \left(\prod_{n' \in M(m) \backslash n} \tanh \left(\frac{z_{mn'}^{t-1}}{2} \right) \right)$$

The variable y_n is just the n$-th$ received bit and σ^2 is the noise variance. This appears to reduce the impact of oscillations and improve decoding performance over the conventional BP algorithm [47].

2.3.9 Expectation Propagation

This method uses the expectation propagation (EP) algorithm to approximate a given joint probability distribution of arbitrary structure with a tree of known structure [48]. The tree is essentially an approximation to the actual joint density based on the product of pair-wise factors along a tree $\mathcal{T}$:

$$q(x) = \frac{\prod_{(j,k)\in\mathcal{T}} q(x_j, x_k)}{\prod_{s\in\mathcal{S}} q(x_s)}$$

Each one of these factors is approximated by EP individually (one by one). The final approximation is the product of the approximate factors (including a division by the appropriate single-node factors to avoid over-counting terms). Presently, no method is available to select the optimal tree structure that would give the best approximation to the actual joint density function [48].

2.3.10 Survey Propagation with random gates

Even though SP has met with great success when applied to the channel decoding problem, a similar statement cannot be made when it comes to lossy source coding

applications. The SP method using random gates changes the nature of the regular parity check nodes from a linear to a non-linear operation on their inputs. The details about how the random checks are chosen for each constraint as well as their behavioral properties when the number of constraints increase are contained in [49]. These random gates approach the theoretical capacity of the parity check nodes and appear to produce better source coding results when combined with a generalized SP scheme. Nonetheless, the computational complexity due to non-linearity in the new random checks as well as the decimation steps can be problematic for practical implementations.

2.3.11 Re-weighted Sum-Product Algorithm

The family of re-weighted sum-product algorithms is a class of message-passing scheme where messages are modified with edge-based weights depending on the structure of the underlying graph [21, 22, 50]. Note that BP is a special case where all the weights are equal to one. These algorithms have been shown to have a unique fixed point regardless of the structural properties of the graph and better stability than BP. More recently, a set of sufficient conditions for convergence was established for this set of approximation schemes [51]. For reference purposes, both the update and marginal

equations, respectively, are presented below:

$$M_{ts}(x_s) \leftarrow \sum_{x'_t} \exp\left\{\frac{\theta_{st}(x_s, x'_t)}{\rho_{st}} + \theta_t(x'_t)\right\} \frac{\prod_{u\in N(t)\setminus s} M_{ut}(x'_t)^{\rho_{ut}}}{M_{st}(x'_t)^{\rho_{st}}}$$

$$\tau_s(x_s) \propto \exp\{\theta_s(x_s)\} \prod_{t\in N(s)} M_{ts}(x_s)^{\rho_{st}}$$

for edge weight values $\rho = [0, 1]$. One of the main drawbacks of this technique is the determination of the optimal set of weights for a given graph.

2.3.12 Consensus Propagation

Consensus propagation is an asynchronous distributed averaging protocol shown to have better convergence properties than BP [52]. It is considered a special case of BP under certain conditions. This method assumes that the underlying probability (MRF) is Gaussian and appears to be useful in the context of wireless sensor network nodes attempting to compute aggregate statistics [52]. Suppose that a number of singly connected nodes belong to the set S_{ij}. In order for this network to estimate an average $\bar{y}$, the nodes need an estimate of the average μ^*_{ij} and the cardinality K^*_{ij}. These quantities are computed recursively as [52]:

$$\mu^t_{ij} = \frac{y_i + \sum_{u\in M(i)\setminus j} K^{t-1}_{ui}\mu^{t-1}_{ui}}{1 + \sum_{u\in N(i)\setminus j} K^{t-1}_{ui}}, \quad \forall\{i, j\}$$

$$K^t_{ij} = 1 + \sum_{u\in N(i)\setminus j} K^{t-1}_{ui}, \quad \forall\{i, j\}$$

where t is the iteration number. For a large number of iterations, the estimated average μ_{mn}^t converges to $\bar{y}$. It is important to note that this algorithm has been proven to converge for the Gaussian case and its convergence rate scales with the total number of nodes. It is related to BP in the sense that it attempts to estimate in a distributed fashion the conditional distributions in a Gaussian MRF.

2.4 Channel and Source Coding

Let a binary linear block code $\mathbb{C}$ be defined by the set of vectors $x = \{0,1\}^N$ that give the null space in H:

$$\mathbb{C} := \{x = \{0,1\}^N Hx = 0\}$$

where H is the $\{0,1\}^{(N-K)\times N}$ parity check matrix operating in the $GF(2)$ domain. This setup is shown in Figure 2.4 and yields a code rate of $R = K/N$

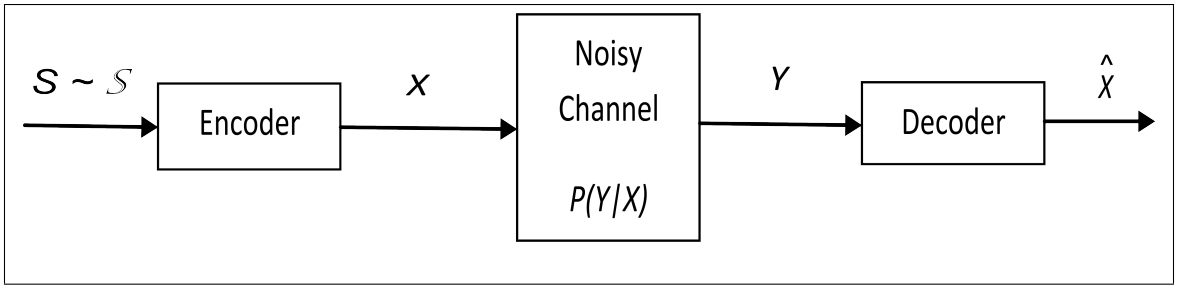

Figure 2.4: Channel Coding

In the channel coding case, the encoder selects a codeword $X \in \mathbb{C}$ and sends it across a noisy channel. The decoder gets a corrupted version of X, named Y. The

channel is characterized by the conditional probability $P(YX)$ which is the probability of observing a particular received sequence given the sequence transmitted originally. Hence, the objective in the channel coding problem is to estimate the most likely transmitted codeword:

$$\hat{X} = \arg\max_{X \in \mathbb{C}} P(Y|Y)$$

The Shannon capacity C of a channel specifies an upper (theoretical) limit on the rate R achieved by any code $X \in \mathbb{C}$ which guarantees error-free transmission [53]. This upper limit is stated formally by the following expression:

$$C = \max_{p(X)} I(X;Y)$$

where $I(X;Y)$ is the mutual information between the sent and received codewords (X and Y respectively) and is defined mathematically for two discrete sources as:

$$\begin{aligned} I(X;Y) &= \sum_{Y}\sum_{X} p(X,Y) \log \frac{p(X,Y)}{p(X)p(Y)} \\ &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned}$$

where $H(X)$ and $H(Y)$ are marginal entropies, $H(X|Y)$ and $H(Y|X)$ are conditional entropies, and $H(X,Y)$ is the joint entropy [54]. The entropy $H(X)$ is defined for a single discrete source as:

$$H(X) = -\sum_{i=1}^{N} p(x_i) \log p(x_i)$$

As remarkable as the channel capacity result is, Shannon did not allude to specific and realizable code constructions that could achieve this capacity limit. More recent developments have shown that promising codes with underlying sparse structure are able to attain rates very close to the Shannon limit when combined with message-passing decoding schemes [55, 56].

On the other hand, in the lossy source coding problem the encoder takes a sequence realization S of length N generated by a random source with distribution $\mathbb{S}$ and attempts to compress it by representing each sequence with a codeword $X \in C$ of length M, where $M < N$. In principle, this compression procedure achieves a rate $R = M/N$ when each source sequence is mapped to a binary code of $2^M = 2^{NR}$ elements. The codeword X is taken by the decoder to reconstruct an estimate $\hat{S}(X)$ of the source sequence S. This situation is depicted in Figure 2.5.

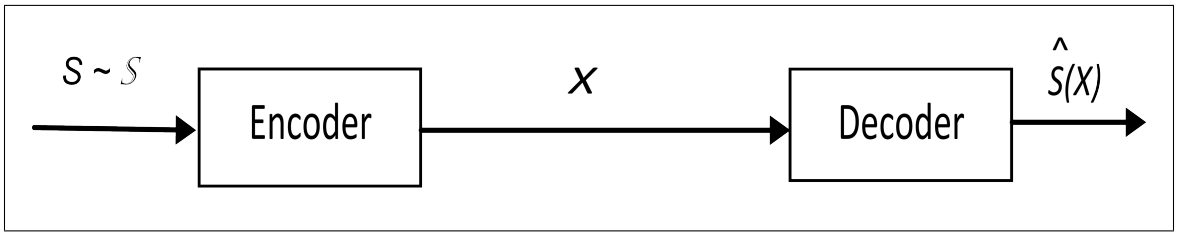

Figure 2.5: Source Coding

The quality of this reconstruction is typically measured by a distortion criterion d. Thus, the objective in source coding is to find a codeword that yields the minimum distortion according to:

$$\hat{X} = \arg\min_{X \in \mathbb{C}} d\left(\hat{S}(X), S\right)$$

where the tradeoff between achievable compression rates R and their corresponding average distortion $D = E[d(\hat{S}(X), S)]$ is described by rate-distortion theory [54]. Given the setup above, the rate-distortion function is:

$$R(D) = \min_{p(\hat{S}(X)|S)} (I(S; \hat{S}(X)) \quad \text{such that} \quad E[d(\hat{S}(X), S)] \leq D$$

for any $D \geq 0$. In other words, the rate $R(D)$ above characterizes the theoretical lower bound that can be achieved in compressing a source so that its subsequent reconstruction does not exceed a given amount of distortion. Nonetheless, as with the channel coding case, no coding strategy has been specified to achieve this objective either.

It is important to note that both channel and source coding are not only duals of each other but also belong to the class of NP-complete problems [54, 57, 58].

2.5 Lossy Source Coding with Side Information

A related problem that is more relevant to the applications that will be examined subsequently is that of lossy source coding with side information. In this case, the objective is still to recover S within a certain distortion bound but the decoder now has side information about S readily available as shown in Figure 2.6: where S and Y are correlated random sequences with joint distribution $p(\mathbb{S}, \mathbb{Y})$. The optimum rate

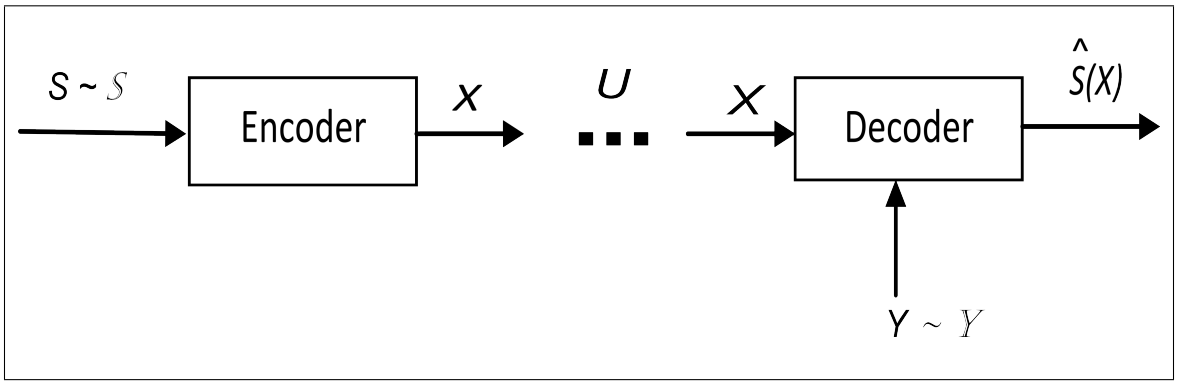

Figure 2.6: Source Coding with Side Information at the Decoder

in this particular setup is given by the following expression [59]:

$$R(D) = \min_{p(U|S),p(\hat{S}|U,Y)} [I(U;S) - I(U;Y)] \quad \text{such that} \quad E[d(S,\hat{S},Y)] \leq D$$

where U is an auxiliary random sequence with finite alphabet $\mathbb{U}$. The rate-distortion function could also be expressed as:

$$R(D) = \min_{p(U|S),p(\hat{S}|U,Y)} [I(S;U|Y)] \quad \text{such that} \quad E[d(S,\hat{S},Y)] \leq D$$

where the identity $I(U;S) - I(U;Y) = I(S;U|Y)$ is derived from the following Markov chains: $Y \rightarrow S \rightarrow U$ and $S \rightarrow (U,Y) \rightarrow \hat{S}$. The first Markov chain represents the fact that only the decoder has access to the side information. The second Markov chain implies that the decoder does not have access to the source. Another important result is that a duality has been established between source coding with side information at the decoder and channel coding with side information at the encoder [60]. Similar results have extended this duality to quantization problems such as information hiding [61].

2.6 Codeword Quantization

Given the dualities exposed in the section 2.5, the codeword quantization (i.e. lossy source coding with side information) problem simply boils down to finding a minimum-weight error pattern consistent with a given error syndrome [62]. Recall that the set of vectors $c \in \mathbb{C}$ defines a code which produces a null space in the parity check matrix H. Let the coset of this linear block code be the set of vectors y of length N in $GF(2)$ defined as:

$$\mathbb{C}(s) = y \in GF(2)^N \quad \text{such that} \quad s = Hy$$

where s is the error syndrome produced by this coset. The challenge is to obtain a y that minimizes the Hamming distance (e.g. distortion measure) $d(w, y)$ subject to the constraints specified by the parity check matrix H.

This can be stated formally as:

$$\min_{y} d(w, y) \quad \text{subject to} \quad s = Hy$$

which in turn is equivalent to finding the minimum-weight vector e (error pattern) according to the following [62]:

$$\min_{e} d(e, 0) \quad \text{subject to} \quad s = He$$

where $y = w + e$ and $w \in GF(2)^N$ is an arbitrary vector. Note that the error syndrome s is treated as the side information in this framework while w is considered

the vector received at the decoder and y could be seen as the recovered information of interest. In many applications, the error syndrome could be interpreted as a set of given constraints and be incorporated on a factor graph as shown in Figure 2.7. where the set $q_N \in GF(2)^N$ represents the constraints imposed on the factor nodes

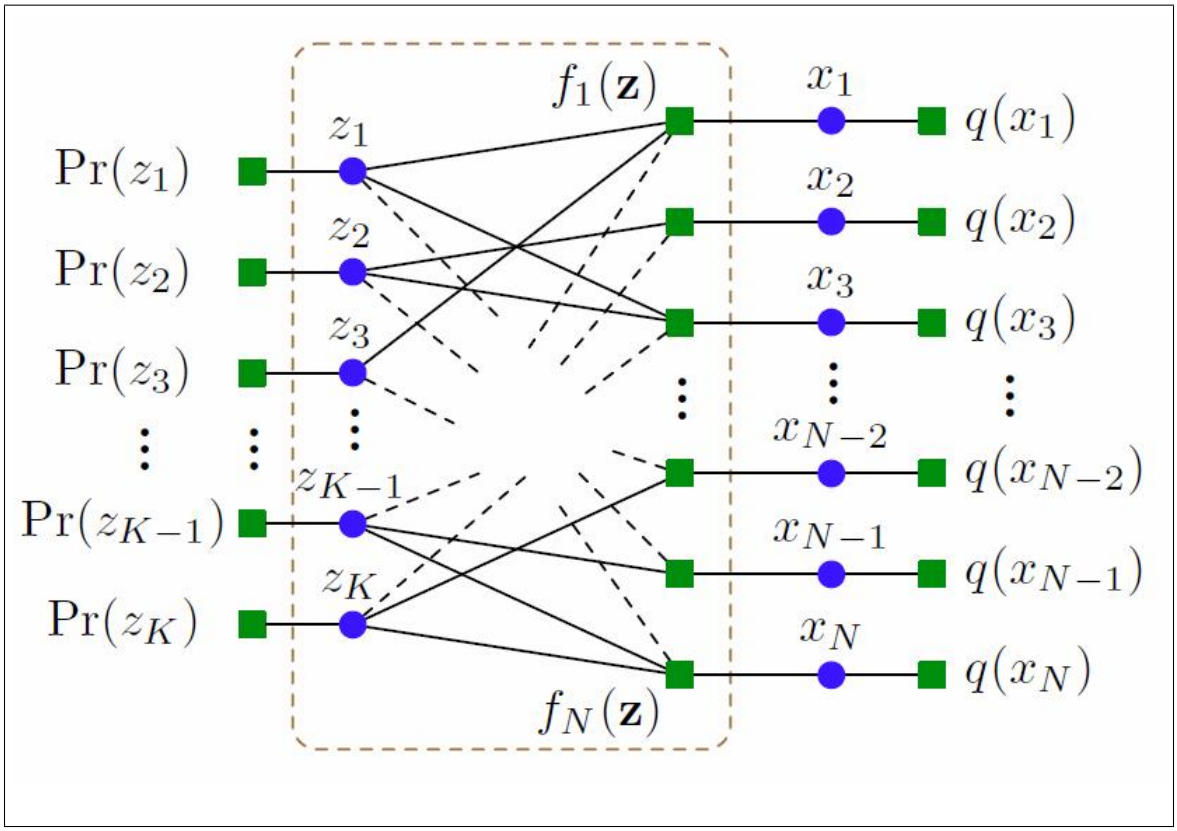

Figure 2.7: Factor Graph for a Low-Density Generator Matrix [62]

f_N.

It is also important to note that the factor graph above does not capture the typical linear block code structure embedded in H in the traditional sense. Instead, it constitutes a factor graph adaptated to codeword quantization based on the concept of a Low-Density Generator Matrix (LDGM) [63]. LDGM codes used for source

coding are the duals of LDPC codes used for channel coding [1, 2]. Hence, the factor graph structure in Figure ?? represents the following relationship:

$$x = Gz$$

where the generator matrix $G \in 0, 1^{N \times K}$ defines the linear code $c \in \mathbb{C}$ and is the dual of the parity check matrix H, such that $HG = 0$. If x satisfies the parity constraints given by q(i.e. $s = Hx$) then the codeword quantization problem can be re-formulated by the expression below:

$$\min_{z} d(x, Gz)$$

The effective use of LDGM codes also requires the implementation of new message passing algorithms since the standard BP has proven to be inadequate in codeword quantization applications [1, 2]. Two novel message-passing algorithms will thus be introduced next to address this challenge.

3

New Iterative Source Coding Algorithms

Belief Propagation has become a very attractive approach to multiple statistical inference problems, including channel decoding, since its approximations often yield astonishing results with relatively low computational complexity even for factor graphs with cycles [19]. Despite the fact that Belief Propagation has met with great success when used iteratively for channel decoding, many theoretical and practical challenges still remain in other important areas of communications and signal processing which are trying to benefit from advances in sparse graphical codes and message-passing algorithms. For instance, lossy source coding is a prime example of a problem where standard Belief Propagation and other local heuristic schemes do not yield good results [2]. Hence, there is a dire need for new efficient and practical algorithms that can be brought to bear to tackle this enormous challenge.

Nonetheless, the BP algorithm continues to provide fertile ground for the de-

velopment of new instantiations applicable to problems where the standard version of the algorithm typically fails. Moreover, a fairly large number of message-passing algorithms based on modifications to the BP algorithm have already been proposed over the years, including many that were reviewed in chapter 2. Thus, BP is the starting point for the developments detailed herein. Specifically, two novel message-passing procedures are formally introduced and their rate-distortion performance is subsequently assessed.

3.1 Truthiness Propagation

A new practical approach inspired by the lossy source coding (with side information) problem has been recently proposed [64]. This method, dubbed Truthiness Propagation (TP), is closely related to the standard Belief Propagation algorithm and involves negligible overhead compared to other recent techniques such as Survey Propagation.

The Truthiness Propagation algorithm is based on a relatively simple modification to the belief propagation update equations. It is well-known that Belief Propagation has deficiencies when used to perform codeword quantization [1, 2]. A factor graph captures the source coding problem by adding constraints to each factor node as shown in Figure 2.7. These factor graphs will invariably have multiple cycles. Since the standard Belief Propagation is still effective in many instances for decoding of error-control codes, it is a natural choice for the purposes of lossy source coding. However, the structure of the posterior channel distribution function in source coding

is quite different from that of channel coding. In channel coding, the distribution is uni-modal and a noisy measurement of the codeword sent by the transmitter would lie in the vicinity of the center of this distribution. On the contrary, in source coding the posterior distribution is multi-modal and a source sequence may lie equidistant to more than one likely codeword with equal amount of distortion [1]. Figure 3.1 shows a zoomed-in version of the typical message flow between variable nodes, a factor node and their respective constraint. A message generated by f_j could be interpreted as the probability that bits z_k connected to f_j produce an odd parity. If the constraint q_j is equal to the variable x_j, conventional BP does not converge. If the constraint q_j is equal to a probability based on x_j, conventional Belief Propagation yields meaningless probabilities.

If, on the other hand, the constraint q_j is replaced by a combination of the hard constraint and the message from f_j the Truthiness Propagation algorithm emerges [64]:

$$x_j \to f_j = \alpha x_j + (1-\alpha) q_j \to x_j, \quad \text{for} \quad 0 < \alpha < 1$$

where $x_j \to f_j$ and $q_j \to x_j$ are the messages from variable node x_j to function node f_j and from the constraint q_j to the variable node x_j, respectively.

These messages are blended via the factor α dubbed the truthiness factor. The term truthiness is used to draw a parallel to the situation where the information propagated is the one desired to be true rather than the actual truth. The factor α is bounded on the $[0, 1]$ interval. The general idea is to maintain (and feedback) the parity of the check f_j if it coincides with the parity of q_j or to allow further

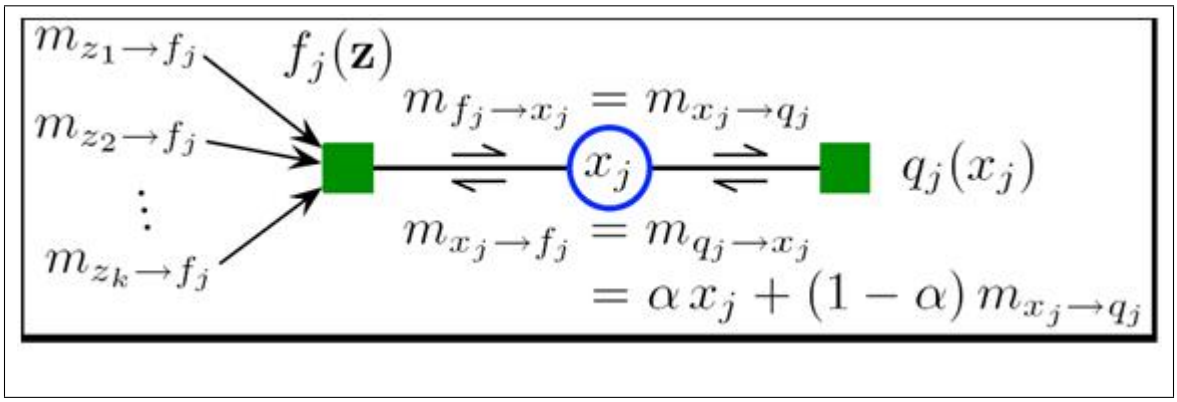

Figure 3.1: Message Flow in the "Truthiness" Propagation Algorithm [62]

updates if it does not. Even though the algorithm appears to converge to a quantizing solution (even for short block lengths), the reasons behind its apparent superiority to conventional Belief Propagation (at least for codeword quantization) remain unknown.

3.1.1 Information-Geometric Interpretation

The information-geometric description of both the BP algorithm and the Bethe free energy stationary points was treated in chapter 2. A possible extension to TP was considered in [65] and is explored here. Let the function nodes $f_a(\mathbf{z}_a)$ in Figure 2.7 be replaced with $f_a(\mathbf{z}_a, x_a)$ which takes into account the constraint x_a in the parity computation. Applying the notation from chapter 2, the average Bethe energy expression can be modified as follows:

$$U'_{Bethe} = -\sum_{a=1}^{M}(\langle p_a, \phi_a \rangle + t_x p_x)$$

where the terms in the summation above are interpreted as:

$$\phi_a(\mathbf{z}_a, x_a) = \log[f_a(\mathbf{z}_a, x_a)/f_a(0,0)]$$
$$p_x = \log[p(x_a = 1)/p(x_a = 0)] = \log[t_x/1 - t_x]$$

The point mass functions $\mathbf{p}_a$ and p_x are defined over all potential outcomes of the combined variables $(\mathbf{z}_a, x_a)$ and the constraint x_a, respectively. Likewise, the terms in the Bethe entropy can be re-arranged as shown below using a mutual information term $I(\mathbf{z}_a, x_a)$ between the variables connected to factor node a, including the constraint variable x_a. This term is defined as the Kullback-Leibler distance between the factor node belief $\mathbf{p}_a$ and the product distribution of the single node beliefs connected to factor node a (i.e. $\prod_{i \in a} t_i$) [65].

The modified Bethe entropy is expressed as:

$$\begin{aligned}
H'_{Bethe} &= -\sum_{i=1}^{N}[t_i \log t_i + (1 - t_i)\log(1 - t_i)] \\
&\quad - \alpha \sum_{a=1}^{M}[t_x \log t_x + (1 - t_x)\log(1 - t_x)] - (1 - \alpha)\sum_{a=1}^{M} I(\mathbf{z}_a, x_a) \\
&= -(1 - \alpha)\sum_{a=1}^{M}\left(\sum_{j=0}^{2^{d_a}-1} p_{a,j} \log p_{a,j}\right) \\
&\quad + \sum_{i=1}^{N}(d_i(1 - \alpha) - 1)[t_i \log t_i + (1 - t_i)\log(1 - t_i)] \\
&\quad + (1 - 2\alpha)\sum_{a=1}^{M}[t_x \log t_x + (1 - t_x)\log(1 - t_x)]
\end{aligned}$$

where the entropy introduced by the constraint node x_a is accounted for in a separate term and scaled by α consistent with the definition of TP in section 3.1. In turn, since the constraints $\mathbf{x}$ are present in every factor node f_a along the graph, the mutual information terms $I(\mathbf{z}_a, x_a)$ above are scaled by $1-\alpha$, which is also consistent with the original definition of TP. Thus, the newly-defined Bethe free energy approximation is given by:

$$\begin{aligned} F'_{Bethe} &= U'_{Bethe} + H'_{Bethe} \\ &= -\sum_{a=1}^{M} \Bigg(\alpha \sum_{j=0}^{2^{d_a}-1} p_{a,j} \log p_{a,j} + \langle p_a, \phi_a \rangle + t_x p_x \\ &\quad + (2\alpha - 1)[t_x \log t_x + (1 - t_x) \log(1 - t_x)] \Bigg) \\ &\quad + \sum_{i=1}^{N} (d_i - 1)[t_i \log t_i + (1 - t_i) \log(1 - t_i)] \end{aligned}$$

The consistency constraints also need to be slightly modified to accommodate the augmented set of variables at the factor nodes $(\mathbf{z}_a, x_a)$:

$$B_a^T \mathbf{p}_a = \begin{bmatrix} \mathbf{t}_a \\ t_x \end{bmatrix} \quad \text{for} \quad a = 1, \cdots, M$$

where B_a is a matrix collecting all possible binary outcomes of $(\mathbf{z}_a, x_a)$, $\mathbf{t}_a$ are the candidate marginals of the variables connected to factor node a, and t_x the candidate marginal of the constraint x_a as defined above. The Lagrangian equation can now be

constructed by incorporating the modified consistency constraints as shown below:

$$\begin{aligned}\mathcal{L}'_{Bethe} = -\sum_{a=1}^{M} \Bigg(& (1-\alpha) \sum_{j=0}^{2^{d_a}-1} p_{a,j} \log p_{a,j} + \langle p_a, \phi_a \rangle + t_x p_x \\ & + (2\alpha - 1)[t_x \log t_x + (1-t_x)\log(1-t_x)] \Bigg) \\ & + \sum_{i=1}^{N} (d_i(1-\alpha) - 1)[t_i \log t_i + (1-t_i)\log(1-t_i)] \\ & + \sum_{a=1}^{M} \langle \begin{bmatrix} t_a \\ t_x \end{bmatrix} - B_a^T p_a, \lambda_a \rangle + \sum_{a=1}^{M} \lambda_x \left(\log \frac{t_x}{1-t_x} - p_x \right)\end{aligned}$$

The critical points of the objective function are found from the following set of partial derivatives:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial t_i} &= -(d_i(1-\alpha) - 1) \log \frac{t_i}{1-t_i} + \sum_{a \in N(i)} \lambda_{i \to a} \\ \frac{\partial \mathcal{L}}{\partial t_x} &= (1-2\alpha) \log \frac{t_x}{1-t_x} + \lambda_{x_a \to a} \\ \frac{\partial \mathcal{L}}{\partial p_a} &= (1-\alpha)\theta_a - B_a \lambda_a - \phi_a\end{aligned}$$

Setting these partial derivatives to zero and solving for the appropriate vari-

ables gives the following update equations:

$$\tau_i \triangleq \log \frac{t_i}{1-t_i} = \frac{1}{d_i(1-\alpha)-1} \sum_{a \in N(i)} \lambda_{i \to a}$$
$$p_x \triangleq \log \frac{t_x}{1-t_x} = \frac{1}{2\alpha - 1} \lambda_{x_a \to a}$$
$$\theta_a = \frac{B_a \lambda_a + \phi_a}{1-\alpha}$$

where the log belief ratios given in the first two equations above are the messages relayed back to the factor node from the variables and constraint nodes respectively, and the third expression is the outgoing message from the factor node. These expressions would correspond to the proposed fixed points of the TP algorithm, or equivalently its Bethe free energy stationary points. The Bethe free energy stationary points are again characterized by the marginal consistency equation shown earlier with the augmented variable space $(\mathbf{z}_a, x_a)$.

The TP algorithm computes marginals and returns a projection of these marginals onto the set of product distributions, but unlike BP the projection is scaled by $1-\alpha$. It is also noteworthy that in this framework the BP fixed points can no longer be recovered by just setting the truthiness factor α equal to 1. The TP fixed points derived above are only valid in the context of lossy source coding with side information. This particular formulation is precluded in generalized coding instances due to the presence and intervention of the constraints on every factor node. Furthermore, as with other recursive algorithms covered previously, only limited empirical guidance exist for setting the value of the truthiness factor α for optimal results [64]. The TP algorithm is summarized in table 3.1 above.

Initialization	$m^{(0)}_{z_i \to f_j}(1) = 0.5 \pm \text{dither}$
Natural Parities	$m^{(n)}_{f_j \to x_j}(1) = \frac{1}{2}\left\{1 - \prod_{k \in V(j)} \left(1 - 2m^{(n)}_{z_k \to f_j}(1)\right)\right\}$ $m_{x_j \to f_j(1)} = \alpha x_j + (1-\alpha) m^{(n)}_{f_j \to x_j}(1)$
Check Nodes	$m^{(n)}_{f_j \to z_i}(z_i) =$ $\frac{1}{2}\left\{1 + (-1)^{z_i}\left(1 - 2m^{(n)}_{x_j \to f_j}(1)\right)\prod_{k \in V(j)\setminus i}\left(1 - 2m^{(n)}_{z_k \to f_j}(1)\right)\right\}$
Variable Nodes	$m^{(n+1)}_{z_i \to f_j}(z_i) = \zeta_{ij} \prod_{k \in F(i)\setminus j} m^{(n)}_{f_k \to z_i}(z_i)$
Beliefs	$b^{(n+1)}_i(z_i) = \zeta_i \prod_{k \in F(i)} m^{(n)}_{f_k \to z_i}(z_i)$

Table 3.1: Summary of Truthiness Propagation Recursive Equations [62]

3.2 Modified Truthiness Propagation

The equivalence between Bethe free energy stationary points and BP fixed points is fundamental in understanding the effectiveness of BP. Moreover, it also provides powerful insight to develop new algorithms that could be applied to solve difficult problems such as codeword quantization. The second algorithm presented in this chapter is called Modified Truthiness Propagation (MTP). The Modified Truthiness Propagation algorithm is based on modifications to the Bethe free energy approximation inspired by the original Truthiness Propagation formulation. Detailed derivations are illustrated in sections 3.2.1 and 3.2.2.

3.2.1 Bethe Free Energy-Based Derivation

Let the return message from factor node a be defined as follows:

$$m_{a\to j} = \beta_j \sum_{x_a \setminus x_j} \sum_{\varepsilon=0}^{1} f_a(x_a, \varepsilon_a) m_{\varepsilon\to a}(\varepsilon_a) \prod_{i\in N(a)\setminus j} m_{i\to a}(x_i)$$

where β_j is a constant to ensure normality and the factor node a is given by:

$$f_a(x_a, \epsilon_a) = \begin{cases} 1, & \text{if } (x_a, \varepsilon_a) \text{ has even parity} \\ 0, & \text{if } (x_a, \varepsilon_a) \text{ has odd parity} \end{cases}$$

The message $m_{\varepsilon\to a}(\varepsilon_a)$ is generated from the following convex combination:

$$m_{\varepsilon\to a}(\varepsilon_a) = \alpha\varepsilon_a + (1-\alpha) m_{a\to\varepsilon}(\varepsilon_a)$$

where the first term represents the hard constraint and the remaining term constitutes the natural parity from the incoming messages into factor node a. This convex sum of the messages (probabilities) is the principle behind TP as exposed in section 3.1.1.

The basic strategy to develop the fixed points of the MTP algorithm is to incorporate the hard constraint beliefs ε impinging on the factor nodes into the Bethe approximation via a convex combination using the truthiness factor α. The set of variables impinging on factor node a is expanded to include the value of the desired parity bit ε_a and will be denoted by $\mathbf{z}_a = (x_a, x_\varepsilon)$. The MTP Bethe free energy

approximation then becomes:

$$\begin{aligned}\bar{F}_{Bethe} = &- \sum_{a=1}^{M} \sum_{z_a} b_a(z_a) \ln f_a(x_a) + \sum_{a=1}^{M} \sum_{z_a} b_a(z_a) \ln b_a(z_a) \\ &- \sum_{i=1}^{N} (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i) + \alpha \sum_{\varepsilon=1}^{M} \sum_{x_\varepsilon} b_\varepsilon(x_\varepsilon) \ln b_\varepsilon(x_\varepsilon) \\ &+ (1-\alpha) \sum_{a=1}^{M} \sum_{z_a} b_\varepsilon(x_\varepsilon) \ln f_a(z_a)\end{aligned}$$

The last two terms of the equation above constitute the main modification to the Bethe approximation. The first is the negative entropy of the constraining node ε_a modulated by α. The second is the contribution of the hard constraints to the average energy modulated by $1 - \alpha$. An additional modification is necessary to the marginalization conditions imposed on the hard constraint beliefs. The new Lagrangian equation is defined as:

$$\begin{aligned}\bar{L}(b, \lambda) = \bar{F}_{Bethe} &+ \sum_i \lambda_i \left[\sum_{x_i} b_i(x_i) - 1 \right] + \sum_a \lambda_a \left[\sum_a b_a(x_a) - 1 \right] \\ &+ \sum_i \sum_{a \in N(i)} \sum_{x_i} \lambda_{ai} \left[b_i(x_i) - \sum_{x_a \setminus x_i} b_a(x_a) \right] + \sum_\varepsilon \lambda_\varepsilon \left[\sum_{x_\varepsilon} b_\varepsilon(x_\varepsilon) - 1 \right] \\ &+ \sum_\varepsilon \sum_{a \in N(\varepsilon)} \sum_{x_\varepsilon} \lambda_{a\varepsilon} \left[(1-\alpha) b_\varepsilon(x_\varepsilon) - \sum_{z_a \setminus x_\varepsilon} b_a(z_a) \right]\end{aligned}$$

The previous two mathematical expressions imply that MTP may be seen as a generalization of Truthiness Propagation in the sense that the truthiness factor is used to scale the interactions between constraint nodes and factor nodes. It is also

a generalization of BP since the fixed points of the standard algorithm are recovered if α is set to 1. Another observation is that the definition of the marginalization property for the constraint nodes could potentially yield a finer approximation to the marginal polytope $MARG(G)$ than the local consistency rules typically enforced for the BP fixed points would.

The next step is to perform constrained optimization on the modified Lagrangian equation. The partial derivative is taken with respect to the single node belief $b_i(x_i)$ and set to zero to get:

$$\nabla_{b_i} \bar{L}(b, \lambda) = -(d_i - 1) \ln b_i(x_i) - (d_i - 1) + \lambda_i + \sum_{a \in N(i)} \lambda_{ai} = 0$$

Solving for $\ln b_i(x_i)$ we have:

$$\ln b_i(x_i) = -1 + \frac{1}{d_i - 1} \lambda_i + \frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}$$

The Lagrangian multiplier λ_{ai} is replaced by the following expression:

$$\lambda_{ai} = \ln \prod_{c \in N(a) \backslash i} m_{ca}(z_a)$$

where m_{ca} is the collection of messages impinging on node a, except the message from node i, to obtain:

$$\ln b_i(x_i) = -1 + \frac{1}{d_i - 1} \sum_{a \in N(i)} \ln \prod_{c \in N(a) \backslash i} m_{ca}(z_a)$$

Applying the natural exponent to the whole expression results in:

$$b_i(x_i) \propto \frac{1}{d_i - 1} \prod_{a \in N(i)} \prod_{c \in N(a) \setminus i} m_{ca}(z_a)$$

which is one of the fixed points of the Modified Truthiness Propagation algorithm. Note that the Modified Truthiness Propagation single node update equation above is the same as both BP and TP (up to a scale factor).

The partial derivative with respect to $b_a(z_a)$ is now taken to get:

$$\nabla_{b_a} \bar{L}(b, \lambda) = -\ln f_a(z_a) + \ln b_a(z_a) + 1 + \lambda_a - \sum_{i \in N(a) \setminus \varepsilon} \lambda_{ia} - \sum_{\varepsilon \in N(a) \setminus i} \lambda_{\varepsilon a} = 0$$

After solving for $\ln b_a(z_a)$ the following is obtained:

$$\ln b_a(z_a) = \ln f_a(z_a) - 1 - \lambda_a + \sum_{i \in N(a) \setminus \varepsilon} \lambda_{ia} + \sum_{\varepsilon \in N(a) \setminus i} \lambda_{\varepsilon a}$$

Replacing the Lagrange multipliers λ_{ia} and $\lambda_{\varepsilon a}$ with the following,

$$\begin{aligned} \lambda_{ia} &= \ln \prod_{c \in N(i)} m_{ci}(x_i) \\ \lambda_{\varepsilon a} &= m_{\varepsilon a}(x_\varepsilon) = \ln \prod_{a \in N(\varepsilon)} m_{a\varepsilon}(z_a) \end{aligned}$$

gives:

$$\ln b_a(z_a) = \ln f_a(z_a) - 1 + \sum_{i \in N(a) \setminus \varepsilon} \ln \prod_{c \in N(i)} m_{ci}(x_i) + \sum_{\varepsilon \in N(a) \setminus i} m_{\varepsilon a}(x_\varepsilon)$$

After applying the natural exponent the following is obtained:

$$b_a(z_a) \propto f_a(z_a) \prod_{i \in N(a) \setminus \varepsilon} \prod_{c \in N(i)} m_{ci}(x_i) \sum_{\varepsilon \in N(a) \setminus i} m_{\varepsilon a}(x_\varepsilon)$$

This expression is very similar to the Belief Propagation factor node fixed point equation shown earlier, except for the additional term $m_{\varepsilon a}$ representing the message coming from the hard constraint ε_a. This term will be formally defined once the last partial derivative is taken. To this end, the partial derivative with respect to the hard constraint belief is:

$$\nabla_{b_\varepsilon} \bar{L}(b, \lambda) = (1 - \alpha) \ln f_a(z_a) + \alpha \ln b_\varepsilon(x_\varepsilon) + \alpha + \lambda_\varepsilon - (1 - \alpha) \sum_{\varepsilon \in N(a) \setminus i} \lambda_{\varepsilon a} = 0$$

In this case, the interest is to solve for the message coming from the hard constraint ε represented by the Lagrangian multiplier λ_ε:

$$\lambda_\varepsilon = -(1 - \alpha) \ln f_a(z_a) - \alpha \ln b_\varepsilon(x_\varepsilon) - \alpha + (1 - \alpha) \sum_{\varepsilon \in N(a) \setminus i} \lambda_{\varepsilon a}$$

Replacing $\lambda_{\varepsilon a}$ with:

$$\lambda_{\varepsilon a} = \ln \prod_{a \in N(\varepsilon)} m_{a\varepsilon}(z_a)$$

and applying the natural exponent the following is obtained:

$$\lambda_\varepsilon \propto b_\varepsilon(x_\varepsilon)^\alpha \sum_{z_a} \left(f_a(z_a) \prod_{\varepsilon \in N(a) \setminus i} \prod_{a \in N(\varepsilon)} m_{a\varepsilon}(z_a) \right)^{1-\alpha}$$

where $b_\varepsilon(x_\varepsilon)$ is either a known probability (soft constraint) or a binary hard constraint.

This last expression constitutes the core of the MTP algorithm. At first glance it appears similar to the original TP factor node update equation but it differs in that the convex combination of the hard constraint value, the incoming messages to the factor node, and the factor node parity is taken in the log-probability space. In contrast, the TP algorithm applies this convex combination in the probability space directly. Equivalently, the MTP update could be seen as a product of probabilities scaled exponentially by α and $1 - \alpha$. This constitutes the only difference between TP and MTP. Both iterative algorithms share the same update equations shown in table 3.1, except that MTP has a different update rule for the message going from the constraint x_j to the factor node f_j. Also, the peculiar marginalization definition for the constraint nodes in MTP appears to suggest that only a partial set of constraints (i.e. side information) is active during the message iterations. This feature could be used to replace pruning in future extensions of the algorithm since it would be easier to slightly change the value of α every few iterations depending on the staleness of the messages rather than the multiple pauses and re-runs involved with pruning. Furthermore, it alludes to more profound connections to SP (and possibly BiP) and could also pave the way for possible use in certain constraint-satisfaction problems [27, 34].

3.2.2 Log-Partition Function-Based Derivation

The derivation presented in section 3.2.1 followed the notation used by Yedidia, et al. [20]. The objective in this section is to re-formulate the MTP fixed points using the representation of exponential families and partition functions introduced by Wainwright, et al. [7] and presented in chapter 2.

As such, the probability distribution that describes a pair-wise Markov random field is:

$$p(x) = \frac{1}{A} \prod_{c \in C} \Psi_c(x_c)$$

where Ψ_c are potential functions restricted to single or two-node cliques in C present throughout the undirected graph, and A is the normalization or partition function. Since all graphical models on discrete variables can be represented as exponential family distributions, let an undirected graph $G(V, E)$ be described by the following probability distribution:

$$p(\mathbf{x}; \theta) = \exp\left(\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t - A(\theta) \right)$$

The third term in the exponent above is the well-known log partition function defined by the expression:

$$A(\theta) = \log \int_{\chi^n} \exp\langle \theta, \phi(x) \rangle v(\mathrm{d}x)$$

where $\phi(x)$ represents a collection of potential functions delineating the mapping $\chi^n \to \mathbb{R}^+$ on the base measure v defined via $\mathrm{d}v = h(\mathbf{x})\mathrm{d}\mathbf{x}$, with arbitrary $h(\mathbf{x})$ and $\mathrm{d}\mathbf{x} = \prod \mathrm{d}x_s$ being the counting measure with respect to the mapping above [7, 14].

A closed-form description of $p(\mathbf{x};\theta)$ is desirable but unattainable due to the inherent complexity of the terms in the expressions defined above. The variational principle appears to be a suitable alternative. This principle was exposed in chapter 2 and is repeated next for convenience and clarity in the exposition.

The log partition function A is the solution to the optimization problem below:

$$A = \max_{q \in \mathcal{Q}} \left\{ \sum_x q(x) \left[\sum_c \Psi_c(x) \right] - \sum_{x \in \chi^n} q(x) \log q(x) \right\}$$

where $\Psi_c(x)$ again represents the clique functions (or potentials) along the graph. This expression is uniquely maximized when $q = p(\mathbf{x};\theta)$. The probability q belongs to the set of all distributions on the chi^n discrete space denoted by $\mathcal{Q}$. Hence, the rationale of the variational approach is to obtain a $q \cong p$ by approximating the entropy term in the optimization expression above and choosing a suitable set $\mathcal{Q}$ to maximize over. Returning to the log partition function $A(\theta)$, a well-known result of this function is that it is a conjugate dual of itself [7, 14]:

$$A(\mu) = \sup_{\theta \in \mathbb{R}^d} \{\langle \mu, \theta \rangle - A^*(\theta)\}$$

where $\mu = E_\theta[\phi(x)]$ maintains the expression above bounded as long as it belongs to the relative interior of $MARG(G)$. The set $MARG(G)$ is the marginal polytope

defined by the collection of potentials $\phi(x)$ belonging to the graph $G(V, E)$. The polytope contains the set of realizable μ vectors that validates the conjugate duality of $A(\theta)$. In other words, μ vectors lying outside of this polytope force the supremum expression above to be unbounded [14].

Contrasting the conjugate duality expression for $A()$ above with the classical variational principle presented earlier, in the former the optimization takes places over a different space (μ vectors in $MARG(G)$) rather than the space of all distributions as in the latter. One challenging aspect about the optimization expression is that the size of the marginal polytope grows very quickly with increasing graph size making it intractable to compute this set exactly. Another problem is that the dual log partition function is available in closed form for cycle-free graphs only. For an acyclic graph (tree) A is given by the following:

$$A(\mu) = H_{Bethe} = \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} I_{st}(\mu_{st})$$

where the H_s terms represent the singleton entropies and $I_{st}(\mu_{st})$ is the edgewise mutual information [7].

The Bethe approximation assumes that the log-partition function above applies to graphs with cycles and that it is well defined for any $\mu \in MARG(G)$. Nonetheless, defining the marginal polytope is very challenging. The Bethe approximation circumvents this difficulty by defining a set of necessary constraints on the marginals. These constraints are exact for acyclic graphs and are summarized in the expression

below:

$$MARG(G) = LOCAL(G) = \left\{ \tau \geq 0 \quad \middle| \quad \sum_{x_s} \tau_s(x_s) = 1 \quad \middle| \quad \sum_{x_{st}} \tau_{st}(x_{st}) = \tau_t(x_t) \right\}$$

where τ_s and τ_{st} are known as pseudo-marginals. For cyclic graphs, the Bethe approximation asserts that the true marginal polytope is approximated by a convex outer bound defined by the local consistency equations above. In other words, a candidate marginal τ may belong to $LOCAL(G)$, but not necessarily to $MARG(G)$. Hence, the expression for the Bethe variational problem (BVP) is:

$$\max_{\tau \in LOCAL(G)} \left\{ \langle \theta, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st}) \right\}$$

The sum-product updates yield the stationary points of the optimization expression above. However, (except for trees) the BP algorithm can lead to globally inconsistent marginals. This phenomenon manifests itself as the belief propagation fixed points falling into local minima instead of the unique global minimum.

Both Truthiness Propagation and MTP attempt a tighter approximation to the marginal polytope and the partition function. It is possible to obtain an equivalent expression of the modified Bethe approximation derived in section 3.2.1 using exponential families and the partition function approximation. The partition function, or negative entropy A^*, typically does not have a closed form. However, for

acyclic graphs the negative entropy decomposes into the following terms:

$$H_s(x_s) = -\sum_{x_s} \mu_s(x_s) \ln \mu_s(x_s)$$
$$I_{st}(\mu_{st}) = \sum_{x_s,x_t} \mu_{st}(x_s,x_t) \ln \frac{\mu_{st}(x_s,x_t)}{\mu_s(x_s)\mu_t(x_t)} = H_s(\mu_s) + H_t(\mu_t) - H_{st}(\mu_{st})$$

The sum of these two terms constitutes the Bethe entropy approximation to a graph with cycles:

$$A(\mu) \approx H_{Bethe} = \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} I_{st}(\mu_{st})$$

This approximation is exact for acyclic graphs. This is an alternate form of the Bethe approximation presented in section 3.2.1. It is a well-known fact that optimizing the BVP yields the Belief Propagation fixed points. Thus, we proceed to construct the corresponding Lagrangian equation as follows:

$$\begin{aligned} L(\tau,\lambda) = \langle \theta, \tau \rangle &+ \sum_{s \in V \setminus \varepsilon} H_s(\tau_s) - \sum_{(s,t) \in E \setminus (s,\varepsilon)} I_{st}(\tau_{st}) + \sum_{\varepsilon \in V} H_\varepsilon(\tau_\varepsilon) - \sum_{(s,\varepsilon) \in E} I_{s\varepsilon}(\tau_{s\varepsilon}) \\ &+ \sum_{(s,t) \in E \setminus (s,\varepsilon)} \left[\sum_{x_s} \lambda_{ts}(x_s) C_{ts}(x_s) + \sum_{x_t} \lambda_{st}(x_t) C_{st}(x_t) \right] \\ &+ \sum_{(s,\varepsilon) \in E} \left[\sum_{x_s} \lambda_{\varepsilon s}(x_s) C_{\varepsilon s}(x_s) + \sum_{x_\varepsilon} \lambda_{s\varepsilon}(x_\varepsilon) C_{s\varepsilon}(x_\varepsilon) \right] \end{aligned}$$

A clear distinction is drawn between the ε nodes (hard constraints) and their relationship with the other nodes (i.e. edges involving the ε nodes). It is of crucial

importance to distinguish and expand some of the terms in this Lagrangian expression. The inner product term is specifically defined as:

$$\begin{aligned}\langle \theta, \tau \rangle &= \theta_s(x_s)\tau_s(x_s) + \theta_t(x_t)\tau_t(x_t) + \alpha\theta_\varepsilon(x_\varepsilon)\tau_\varepsilon(x_\varepsilon) \\ &+ \theta_{st}(x_s, x_t)\tau_{st}(x_s, x_t) + (1-\alpha)\theta_{s\varepsilon}(x_s, x_\varepsilon)\tau_{s\varepsilon}(x_s, x_\varepsilon)\end{aligned}$$

The constraints are defined below. Note that the ones involving the ε nodes, namely $C_{\varepsilon s}(x_s)$ and $C_{s\varepsilon}(x_\varepsilon)$, are defined differently than the other constraints:

$$\begin{aligned}C_{ts}(x_s) &= \tau_s(x_s) - \sum_{x_t} \tau_{st}(x_s, x_t) = 0 \\ C_{st}(x_t) &= \tau_t(x_t) - \sum_{x_s} \tau_{ts}(x_s, x_t) = 0 \\ C_{\varepsilon s}(x_s) &= \tau_s(x_s) - \alpha \sum_{x_\varepsilon} \tau_{s\varepsilon}(x_s, x_\varepsilon) = 0 \\ C_{s\varepsilon}(x_\varepsilon) &= \alpha\tau_\varepsilon(x_\varepsilon) - \sum_{x_s} \tau_{\varepsilon s}(x_s, x_\varepsilon) = 0\end{aligned}$$

Taking the partial derivative with respect to τ_s and setting it to zero gives:

$$\begin{aligned}\nabla_{\tau_s} L(\tau, \lambda) &= \theta_s + 1 + \ln \tau_s(x_s) + \lambda_{\varepsilon s}(x_\varepsilon) + \sum_{t \in N(s) \setminus \varepsilon} \lambda_{ts}(x_s) = 0 \\ \ln \tau_s(x_s) &= -\theta_s - 1 - \lambda_{\varepsilon s}(x_s) - \sum_{t \in N(s) \setminus \varepsilon} \lambda_{ts}(x_s)\end{aligned}$$

The partial derivative with respect to τ_ε and setting it to zero gives:

$$\nabla_{\tau_\varepsilon} L(\tau, \lambda) = \alpha\theta_\varepsilon(x_\varepsilon) + 1 + \ln \tau_\varepsilon(x_\varepsilon) + \alpha \sum_{s \in N(\varepsilon)} \lambda_{s\varepsilon}(x_\varepsilon) = 0$$

$$\ln \tau_\varepsilon(x_\varepsilon) = -\alpha\theta_\varepsilon(x_\varepsilon) - 1 - \alpha \sum_{s \in N(\varepsilon)} \lambda_{s\varepsilon}(x_\varepsilon)$$

The expressions above define the single node beliefs. The two-node belief equation is obtained by taking the partial derivative and substituting the single node belief equations as follows:

$$\nabla_{\tau_{st}} L(\tau, \lambda) = \theta_{st}(x_s, x_t) - 1 - \ln \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\tau_t(x_t)} - \lambda_{ts}(x_s) - \lambda_{st}(x_t) = 0$$

$$\ln \tau_{st}(x_s, x_t) = \theta_{st}(x_s, x_t) - 1 - \ln \tau_s(x_s) - \ln \tau_t(x_t) - \lambda_{ts}(x_s) - \lambda_{st}(x_t)$$

$$\ln \tau_{st}(x_s, x_t) = \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t) + 1 + \sum_{t \in N(s) \setminus \varepsilon} \lambda_{ts}(x_s)$$

$$+ \sum_{s \in N(t) \setminus \varepsilon} \lambda_{st}(x_t) - \lambda_{ts}(x_s) - \lambda_{st}(x_t) + \lambda_{\varepsilon s}(x_s) + \lambda_{\varepsilon t}(x_t)$$

$$\ln \tau_{st}(x_s, x_t) = \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t) + 1 + \sum_{u \in N(s) \setminus \varepsilon, t} \lambda_{us}(x_s)$$

$$+ \sum_{u \in N(t) \setminus \varepsilon, s} \lambda_{ut}(x_t) + \lambda_{\varepsilon s}(x_s) + \lambda_{\varepsilon t}(x_t)$$

The next step is to introduce an auxiliary two-node belief equation for edges

involving the hard constraint node ε:

$$\begin{aligned}
\nabla_{\tau_{s\varepsilon}}(L(\tau,\lambda) &= \alpha + (1-\alpha)\theta_{s\varepsilon}(x_s,x_\varepsilon) + \ln\frac{\tau_{s\varepsilon}(x_s,x_t)}{\tau_s(x_s)\tau_\varepsilon(x_\varepsilon)} - \alpha\sum_{\varepsilon\in N(s)}\lambda_{\varepsilon s}(x_s) \\
&\quad - \alpha\sum_{s\in N(\varepsilon)}\lambda_{s\varepsilon}(x_\varepsilon) = 0 \\
\ln\tau_{s\varepsilon}(x_s,x_\varepsilon) &= -\alpha - (1-\alpha)\theta_{s\varepsilon}(x_s,x_\varepsilon) + \ln\tau_s(x_s) + \ln\tau_\varepsilon(x_\varepsilon) \\
&\quad + \alpha\sum_{\varepsilon\in N(s)}\lambda_{\varepsilon s}(x_s) + \alpha\sum_{s\in N(\varepsilon)}\lambda_{s\varepsilon}(x_\varepsilon) \\
\ln\tau_{s\varepsilon}(x_s,x_\varepsilon) &= -\alpha - (1-\alpha)\theta_{s\varepsilon}(x_s,x_\varepsilon) - \theta_s(x_s) + \ln\tau_\varepsilon(x_\varepsilon) - \lambda_{\varepsilon s}(x_s) \\
&\quad + \sum_{s\in N(\varepsilon)}\lambda_{s\varepsilon}(x_\varepsilon) + \alpha\sum_{\varepsilon\in N(s)}\lambda_{\varepsilon s}(x_s) - \alpha\sum_{s\in N(\varepsilon)}\lambda_{s\varepsilon}(x_\varepsilon) \\
\ln\tau_{s\varepsilon}(x_s,x_\varepsilon) &= -\alpha - (1-\alpha)\theta_{s\varepsilon}(x_s,x_\varepsilon) - \theta_s(x_s) - \alpha\theta_\varepsilon(x_\varepsilon) \\
&\quad - \sum_{t\in N(s)\setminus\varepsilon}\lambda_{ts}(x_s) + \alpha\sum_{\varepsilon\in N(s)}\lambda_{\varepsilon s}(x_s) + (1-\alpha)\sum_{s\in N(\varepsilon)}\lambda_{s\varepsilon}(x_\varepsilon)
\end{aligned}$$

The equation above unveils the essence of the Modified Truthiness Propagation algorithm since it showcases the peculiar relationship between the hard constraint nodes and their surrounding nodes. More specifically, it shows that the message coming from the hard constraint node is a convex combination (in the log-probability space) of the hard constraint value and the messages arriving at the hard constraint. As in section 3.2.1, this convex combination is modulated by the truthiness parameter α.

The Lagrangian multiplier $\lambda_{us}(x_s)$ is replaced by the following expression:

$$\lambda_{us}(x_s) = \ln \prod_{t \in N(u) \setminus s} M_{tu}(x_u)$$

where $M_{tu}(x_u)$ is the collection of messages impinging on node u, except the message from node s. After this substitution the following expressions for the single and pairwise node beliefs are obtained:

$$\begin{aligned}
\tau_s(x_s) &\propto \exp \theta_s(x_s) M_{\varepsilon s}(x_s) \prod_{t \in N(s) \setminus \varepsilon} M_{ts}(x_s) \\
\tau_\varepsilon(x_\varepsilon) &\propto \exp \alpha \theta_\varepsilon(x_\varepsilon) \left(\prod_{s \in N(\varepsilon)} M_{s\varepsilon}(x_\varepsilon) \right)^\alpha \\
\tau_{st}(x_s, x_t) &\propto \exp(\theta_{st}(x_s, x_t) + \theta_s(x_s) \\
&\quad + \theta_t(x_t)) M_{\varepsilon s}(x_s) M_{\varepsilon t}(x_t) \prod_{u \in N(s) \setminus \varepsilon, t} M_{us}(x_s) \prod_{u \in N(t) \setminus \varepsilon, s} M_{ut}(x_t) \\
\tau_{s\varepsilon}(x_s, x_\varepsilon) &\propto \exp((1-\alpha)\theta_{s\varepsilon}(x_s, x_\varepsilon) - \theta_s(x_s) \\
&\quad - \alpha \theta_\varepsilon(x_\varepsilon)) \prod_{t \in N(s) \setminus \varepsilon} M_{ts}(x_s) \left(\prod_{s \in N(\varepsilon)} M_{s\varepsilon}(x_\varepsilon) \right)^{1-\alpha} \left(\prod_{\varepsilon \in N(s)} M_{\varepsilon s}(x_s) \right)^\alpha
\end{aligned}$$

The Modified Truthiness Propagation fixed points appear after some algebraic manipulation:

$$M_{ts}(x_s) \propto \sum_{x_t} \exp(\theta_{st}(x_s, x_t) - \theta_t(x_t)) M_{\varepsilon t}(x_t) \prod_{u \in N(t) \setminus s, \varepsilon} M_{ut}(x_t)$$

The message coming from the hard constraint ε denoted by $M_{\varepsilon t}(x_t)$ is defined by:

$$M_{\varepsilon t}(x_t) \propto \sum_{x_\varepsilon} \exp((1-\alpha)\theta_{t\varepsilon}(x_t, x_\varepsilon) - \alpha\theta_\varepsilon(x_\varepsilon))(M_\varepsilon(x_\varepsilon))^\alpha \left(\prod_{u \in N(\varepsilon)} M_{u\varepsilon}(x_\varepsilon) \right)^{1-\alpha}$$

where the message $M_\varepsilon(x_\varepsilon)$ is either a fixed value or a known a priori probability.

The preceding derivation shows that both Belief Propagation and Modified Truthiness Propagation follow a similar strategy for estimating the partition function $A(\theta)$. This strategy is heavily based on the Bethe Variational Problem. Where they differ is in the manner which the marginal consistency constraints are defined. The immediate implication is that the convex approximation to the marginal polytope will be different in both instances. The quality of the convex hull approximation proposed by the Modified Truthiness Propagation algorithm will vary depending on the specific graph structure and the chosen value of α. For codeword quantization applications where cyclic graphs and short block lengths are often encountered, both Truthiness Propagation and MTP appear to converge and perform better than Belief Propagation. In other applications, however, performance might be roughly equivalent across these schemes. Some of these considerations are examined in section 3.3.

3.3 Rate-Distortion Performance

Some of the basics of rate distortion theory were introduced in sections 2.4, 2.5, and 2.6. These concepts will be expanded upon next in order to add context and enhance the understanding of the results presented herein. Nonetheless, the focus of this section shall be to demonstrate the rate-distortion performance of MTP.

3.3.1 Binary Symmetric Channel

In most modern coding applications the data of interest is in binary form. A number of standard communications channels are often used to analyze performance in the presence of noise [66]. The Binary Symmetric Channel (BSC) is arguably the most commonly used due to its simplicity and the fact that many complex channels can be reduced to this framework. The BSC is a simple communications channel with binary inputs and outputs and a probability of error (or crossover) p associated with the receiver getting the wrong bit value. The BSC provides a convenient way to benchmark performance since it is rather easy to compute its Shannon capacity (or its rate-distortion function) and compare it to what could actually be achieved. A BSC is shown in Figure 3.2 below

The rate-distortion function $R(D)$ for this channel is equivalent to that obtained for a Bernoulli(p) source with a maximum amount of errors below or equal to

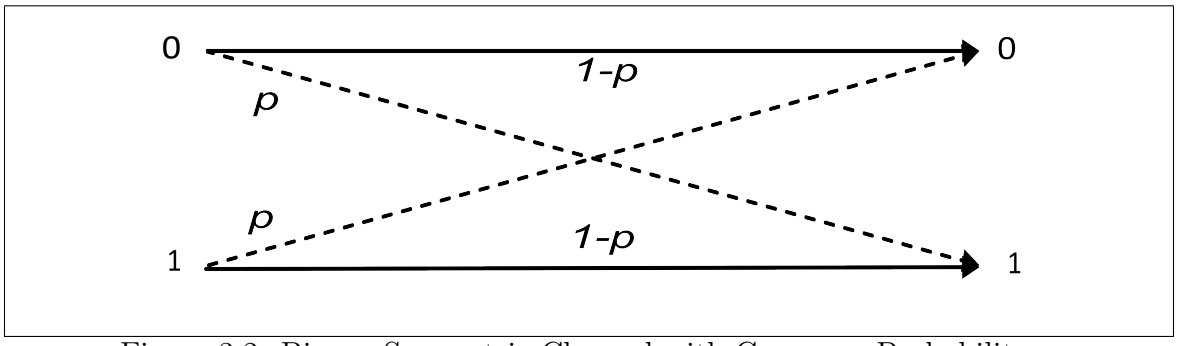

Figure 3.2: Binary Symmetric Channel with Crossover Probability p

D [54]. This is given by the following expression:

$$R(D) = \begin{cases} H(p) - H(D), & 0 \leq D \leq \min(p, 1-p) \\ 0, & D > \min(p, 1-p) \end{cases}$$

where the amount of errors is defined by the Hamming distortion measure,

$$d(x, \hat{x}) = \begin{cases} 0, & \text{if } x = \hat{x} \\ 1, & \text{if } x \neq \hat{x} \end{cases}$$

which is associated with the probability of error $P(X \neq \hat{X}) = E[d(x, \hat{x})]$. For binary sequences, the distortion measure is the average Hamming distortion given by:

$$d(x^n, \hat{x}^n) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)$$

where the distortion is computed on a per-bit basis and averaged over the length of the sequence. Note that the BSC rate-distortion function shown earlier is maximized when the crossover probability $p = 0.5$ [54]. This constitutes the theoretical lower bound in rate-distortion performance used to benchmark source coding algorithms.

3.3.2 Low-Density Generator Matrix Codes

The message-passing algorithms discussed thus far are only efficient when operating over sparse graph structures. This fact underscores the importance of practical code constructions such as LDGM codes for lossy source coding. LDGM codes have been shown to be the other basic ingredient needed in order to attain the elusive rate-distortion bound [1, 2]. LDGM codes are the duals of LDPC codes. Their factor graph encapsulates the following relationship as shown in Figure 2.7:

$$x = Gz$$

where the generator matrix $G \in \{0,1\}^{N \times K}$ defines a linear code $c \in \mathbb{C}$ and is the dual of the parity check matrix H, such that $HG = 0$. Just like LDPC codes, LDGM codes are low-density because both the variable degree and the check degree remain bounded as the block length increases. The degree in this context means the number of edges coming out of either a variable or check node. There are two types of LDGMs. The first is the regular LDGM whose factor graph has uniform (fixed) degree across all variable and/or parity check nodes. Irregular LDGM are those with varying degrees across either or both variable and check nodes. Performance differences among the two could be significant, consistent with what has been demonstrated for LDPC codes [56, 63].

3.3.3 Results

The results shown in this section were generated using a MATLAB® model of the TP and the MTP algorithm. The tests were set up in the following manner:

1. Fixed code block length of 300 bits.
2. 10 randomly-generated LDGMs per code rate using the method outlined in [67].
3. 1000 runs (100 repetitions over 10 LDGMs) per code rate.
4. Hamming distortion $D = E[d(x, Gz)]/N$ computed as ensemble average ($N = 1000$) for each code rate.
5. 300 TP and MTP iterations allowed for each repetition.
6. An optimized α determined for each code rate.

The plot shown in Figure 3.3 corresponds to the rate-distortion function obtained by applying the MTP algorithm in conjunction with a pseudo-random set of 10 regular LDGMs over a BSC per code rate.

The MTP rate-distortion function shown in Figure 3.3 is in the order of 1 dB above the Shannon limit. This rate-distortion performance is pretty remarkable considering that, in general, coding performance tends to get worse with shorter block length sequences as demonstrated with channel coding [19, 56, 67, 68, 69].

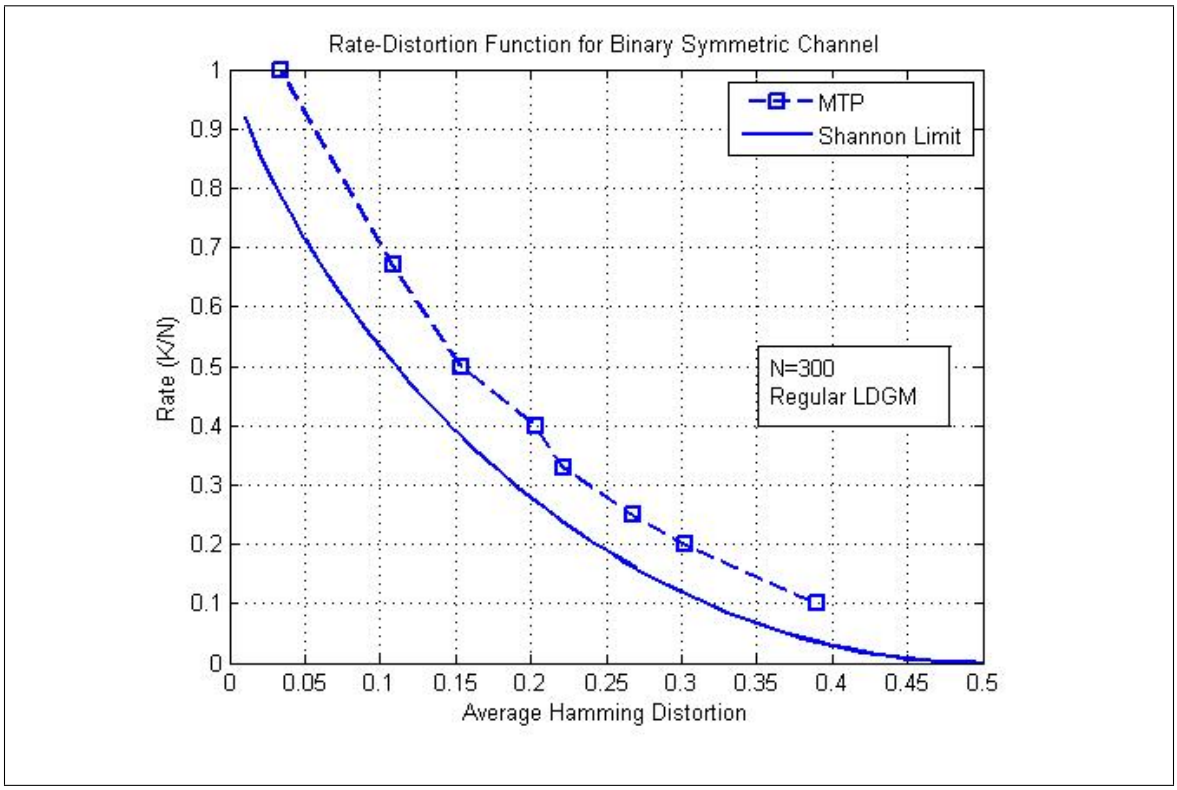

Figure 3.3: MTP Rate-Distortion Function with Regular LDGM

This feature could be very advantageous in high data rate applications where the required throughput makes it nearly impossible to use long block codes. As expected, the performance shown improves (i.e. get closer to the lower bound) as the sequence length increases in accordance with [53]. Another impressive fact is that regular LDGMs were used to generate these results. Some of the techniques presented in chapter 2 tend to perform poorly or even fail to converge altogether when used with regular LDGM codes.

The performance of TP (not shown in Figure 3.3) appears to be slightly better than MTP as seen on [62]. Nonetheless, the MTP behavior seems to be consistently better than the TAP algorithm results reported in [32, 62]. No direct comparisons

were made against the SP algorithm over LDGM codes since their source coding results appeared to be very similar to those obtained with the TAP algorithm as documented in [2, 32, 62].

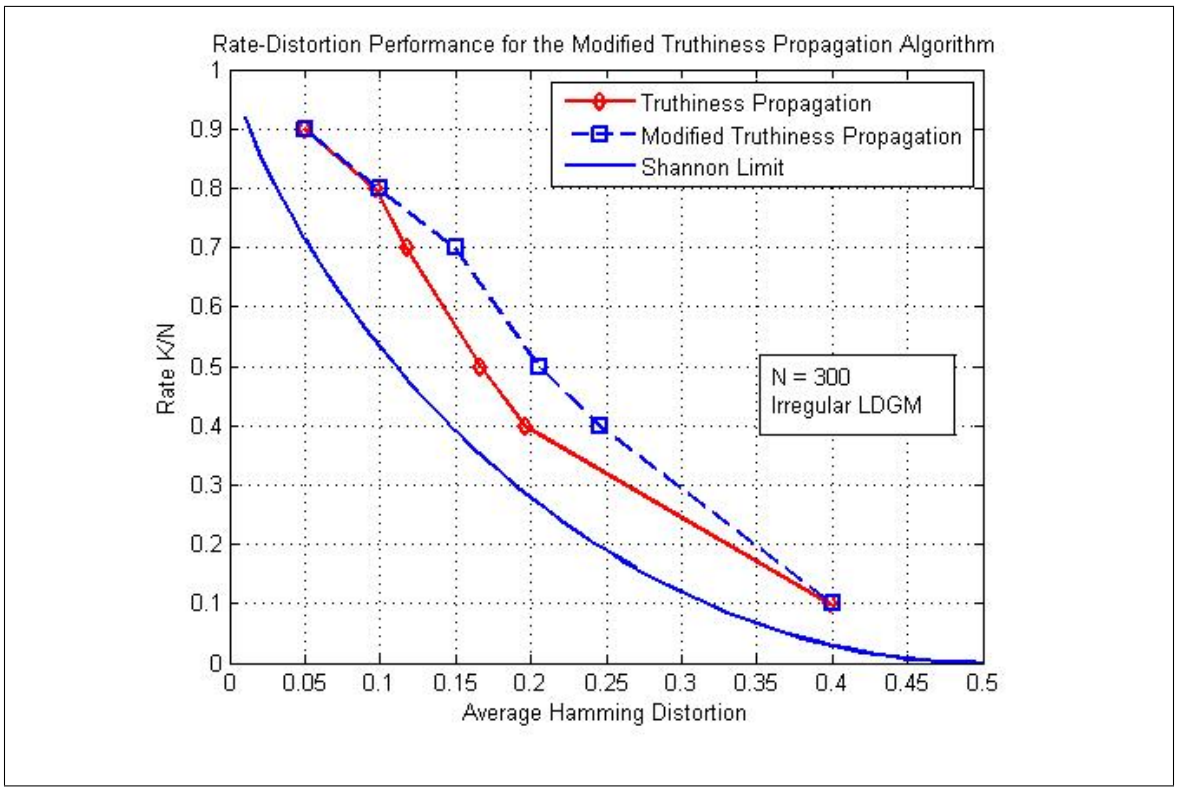

Figure 3.4: MTP Rate-Distortion Function with Irregular LDGM

The results using irregular LDGMs are shown in Figure 3.4. The rate-distortion performance between TP and MTP is comparable under the same set of conditions, again with TP having a slight edge. It is interesting to note that in this case the TAP algorithm is inadequate (i.e. non-convergence) for irregular LDGMs with degree greater than two. This could possibly be due to its close ties to the Ising spin model framework which only accounts for pairwise node relationships [32].

4

Applications of New Source Coding Algorithms

The growing demand for faster wireless devices and services is faced with the reality of finite bandwidth resources. From the viewpoint of the designer, the need to accommodate multiple users in a spectrally-efficient manner is further exacerbated by decreased levels of security/privacy due to the nature of these open high-occupancy channels. These two prevalent aspects in modern communication systems are explored here in the context of the modified BP source coding concepts developed in chapter 3.

The first aspect alludes to challenges associated with multi-user communications. In this setting, the main interest is on enabling technologies and algorithms aimed at attaining the promised levels of capacity while maximizing the number of

users, in particular multi-user Multiple-Input Multiple-Output (MIMO) networks. Dirty Paper Coding (DPC) has recently emerged as a possible solution to this problem [70]. Nevertheless, DPC only lays out a generic strategy to achieve capacity without specifying any algorithms to help attain it.

The second aspect is related to steganography (data hiding). The focus of modern steganography is to conceal information in digital media (i.e. images, audio, etc.) so that only the sender and recipient of the information are aware of its existence [71]. By contrast, cryptography only seeks to protect the contents of the message but not its existence nor that of the communicating parties. Effective message concealment requires sophisticated quantization in order to reduce the amount of perceptual distortion in the carrier media.

4.1 Dirty Paper Coding

One of the most relevant applications of codeword quantization methods is in Dirty Paper Coding. DPC is a theoretical data transmission scheme for channels subjected to interference. This remarkable technique only requires a priori knowledge of the interference at the encoder and not the decoder. The intended message is pre-coded (known interference cancellation) in a way which avoids exceeding power limitations. The name DPC is derived from the title of the seminal paper where the analogy is made between judicious coding of data with a priori knowledge of the interference and writing on a dirty piece of paper with prior knowledge of where the dirt is located [72].

This pioneering work by Costa demonstrated that the capacity of such a channel is the same as if the interference did not exist. More concisely, given an Additive White Gaussian Noise (AWGN) channel with known interference also characterized by a zero-mean Gaussian distribution as shown in Figure 4.1, the capacity C is given by the following expression:

$$C = \frac{1}{2}\log\left[1 + \frac{P}{N}\right], \quad \text{independent of } Q$$

where P is the signal input power, N is the channel noise power, and Q is the interference power [72]. Note in Figure 4.1 that the AWGN channel is operating under the power constraint $\frac{1}{n}\sum_{i=1}^{n} X_i^2 \leq P$, therefore rendering the naive approach of canceling out the interference by over-powering it practically useless.

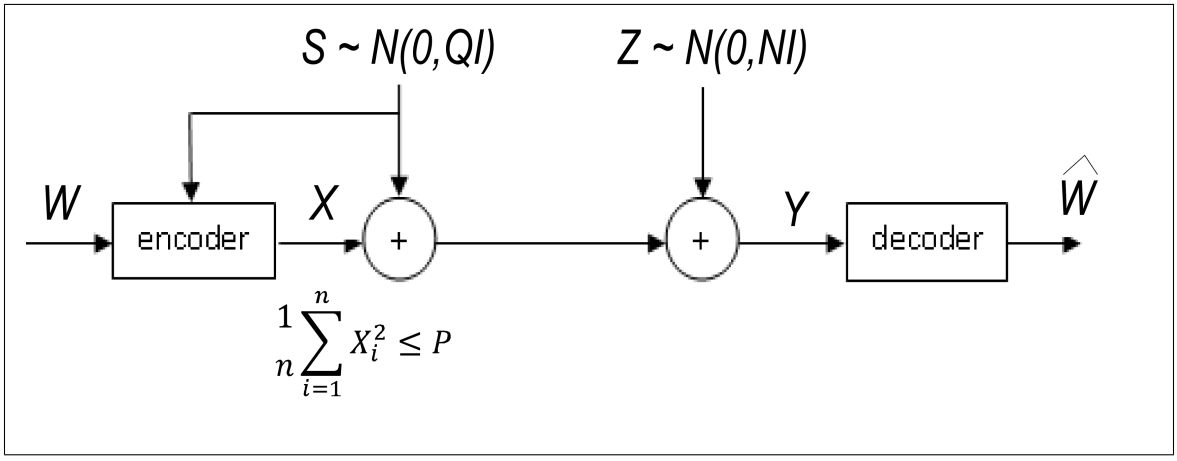

Figure 4.1: AWGN Channel with Interference known to the Encoder [72]

4.1.1 MIMO Channels

DPC has become the focal point in recent efforts to find the potential capacity of multi-user MIMO channels [70]. Traditional communication systems consisted of a single transmit and a single receive antenna. These are referred to as Single-Input Single-Output (SISO) systems. A MIMO system refers to one in which there are multiple transmit and multiple receive antennas. These systems can be used between single users and/or multiple users on both transmit and receive ends. There is an extensive body of literature about these systems and their predicted capacity [73, 74, 75]. The interest in MIMO communications aroused due to the following two factors. First, the potential capacity gains that could be attained compared to Single-Input Single Output (SISO) systems. Second, the increasingly stringent constraints of power, bandwidth, and complexity virtually leave this technology as the only viable option to accommodate the demand for higher data rates. The capacity of single user MIMO systems has been characterized to scale linearly according to $\min(M, N)$ from that of SISO systems, where M and N are the number of antennas at the transmitter and receiver respectively, under certain assumptions about the channel statistical behavior, the spatial correlation of antenna elements, and the availability of Channel State Information (CSI) at the encoder, the decoder, or both [76]. It is conjectured that the relatively large capacity gains are due to the dense scattering environment that provides multiple independent paths between the transmitter and receiver antennas.

The case of multi-user MIMO channels presents a different story. There are two

types of multi-user MIMO channels, namely the Multiple Access Channel (MAC) and the Broadcast Channel (BC). Although several capacity results exist for the MIMO MAC, very little is known about the capacity of the MIMO BC. The capacity of this type of channel will be the focus of section 4.1.2.

4.1.2 MIMO Gaussian Broadcast Channel Capacity

The MIMO BC (or downlink) is described following the notation in [76]. Consider a wireless network where a base station (or access point) has M antennas and K users have N antennas each as shown in Figure 4.2. Then user k receives the signal

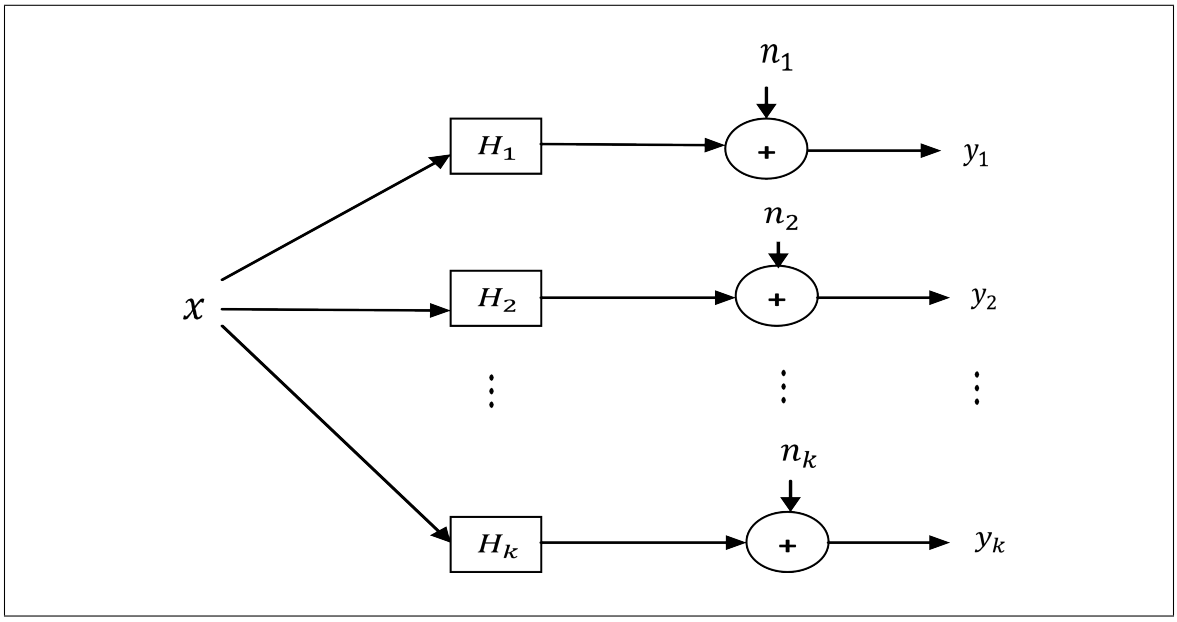

Figure 4.2: MIMO BC System Description [76]

according to the following expression:

$$y_k = H_k x + n_k$$

where x is an $M \times 1$ vector of the transmitted signal, y_k is an $N \times 1$ vector of the signal received by user k, and n_k is an $N \times 1$ noise vector assumed to be circularly symmetric complex Gaussian with identity covariance such that $N(0, \mathbf{I})$. The H_k denotes the channel matrix from the base station to user k. The transmit covariance is $\sum_x = E[xx^T]$ and the base station is subject to an average power constraint $P \geq \text{Trace}(\sum_x)$.

When the transmitter has only one antenna ($M = 1$) the users can be unambiguously ordered by their respective channel strengths, which are assumed to be constant (non-fading) for the present purpose. This is called a degraded Gaussian Broadcast Channel and its capacity is known [54]. When the transmitter has more than one antenna ($M > 1$) the BC is in general non-degraded and a closed-form solution for the capacity is still an open problem. Nonetheless, much progress has been made towards this goal. Inner and outer bounds of this capacity have been found [77, 78]. An achievable capacity region was recently found for the $N = 1$ case, and later extended to the general case $N > 1$ using the DPC concept [70, 79]. The sum-rate capacity using DPC was also proven to be optimal [80, 81, 82]. This DPC achievable capacity region $C_{DPC}(P, \mathbf{H})$ is given by the convex hull of the union of rate vectors $\mathcal{R}(u, \sum_i)$ over all permutations of users $u_1, \cdots, u_k$ and all semi-definite covariance matrices $\sum_1, \cdots, \sum_k$ such that $\text{Trace}(\sum_1 + \cdots + \sum_k) = \text{Trace}(\sum_x) \leq P$:

$$C_{DPC}(P, \mathbf{H}) = Co\left(\bigcup_{u, \sum_i} \mathcal{R}(u, \sum_i)\right)$$

and the rate vectors $\mathcal{R}(u, \sum_i)$ are given as follows:

$$\mathcal{R}_{u_i} = \frac{1}{2} \log \left| \frac{\mathbf{I} + \mathbf{H}_{u_i} \left(\sum_{j \geq i} \sum_{u_j} \right) \mathbf{H}_{u_i}^T}{\mathbf{I} + \mathbf{H}_{u_i} \left(\sum_{j > i} \sum_{u_j} \right) \mathbf{H}_{u_i}^T} \right| \quad i = 1, \ldots, k$$

An important duality has been established between the MAC and the BC [83]. More specifically, the dirty paper region of the MIMO BC is equal to the union of MIMO MAC capacity regions as follows:

$$C_{DPC}(P, \mathbf{H}) = \bigcup_{\sum_{i=1}^{k} P_i = P} C_{MAC}(P_1, \ldots, P_k, \mathbf{H}^T)$$

The importance of this duality is two-fold. First, the DPC region covariances are non-concave functions. However, the MAC covariances yield concave functions, thus using them can considerably ease the capacity computations of the DPC region [76]. Second, it also suggests that the DPC capacity region above could indeed be the elusive MIMO BC capacity region, although this notion has yet to be proven rigorously.

Another important consideration is that even if the dirty paper region is proven to be the overall capacity region for the MIMO BC, the inherent complexities involved in the implementation of the DPC strategy at the encoder renders the scheme very impractical. Hence, this is precisely where the codeword quantization techniques developed previously can contribute to dramatically reduce complexity and help attain capacity. Section 4.1.3 demonstrates the feasibility of this approach via a simple example.

4.1.3 Two-User Dirty Paper Coding Example

The DPC strategy typically involves the use of nested lattice codes [84, 85]. Lattice codes are often employed in Trellis-Coded Quantization (TCQ) schemes in conjunction with the Viterbi algorithm [86]. However, the complexity of lattice codes grows exponentially with the constraint length thereby limiting its practical application. Recent research efforts proposed the use of nested LDGM/LDPC codes (binning) for source quantization with side information with very favorable results [1, 87]. Therefore, the approach for this example is to use nested LDGM/LDPC codes along with the new iterative source quantization procedures in an attempt to attain the DPC capacity for a simple two-user BC. The DPC transmission scheme for a two-user BC is described next following the exposition in [88].

Consider the situation where two users on the transmit side wish to send separate messages, u_1 and u_2 of lengths K_1 and K_2 respectively, via a common access point (base station) to two remote users on the receive side. The encoded messages are represented by the codewords x_1 and x_2, both of length N which yield the following rates $R_1 = K_1/N$ and $R_2 = K_2/N$ for each user. Let H be a $K \times N$ parity check matrix with good (capacity approaching) source coding properties be defined as follows:

$$H = \begin{bmatrix} H_\Delta \\ H_2 \end{bmatrix}$$

where H_Δ is the $K_\Delta \times N$ parity check matrix of a capacity approaching error control code and H_2 the $K_2 \times N$ parity check matrix of a capacity-achieving source code.

Once the message from user 1 is encoded (via a procedure to be described later) to create x_1, the message from user 2 is encoded by introducing a small amount of distortion to x_1. The resulting codeword x_2 is then transmitted over the channel. The BC connecting the transmitter to receiver 2 is described by a memoryless BSC with cross-over probability p_2. To be precise, the codeword x_2 is obtained from the following constrained minimum distance problem:

$$x_2 = \arg\min_{\varepsilon} d(x_1, \varepsilon) \quad \text{subject to} \quad \begin{bmatrix} H_\Delta \\ H_2 \end{bmatrix} \varepsilon = \begin{bmatrix} 0 \\ u_2 \end{bmatrix}$$

On the receive (decoder 2) side, the received signal is denoted by $\mathbf{z} = [z_1, \ldots, z_N]$. The estimated codeword is obtained as follows:

$$\hat{x}_2 = \arg\min_{\varepsilon} d(\mathbf{z}, \varepsilon) \quad \text{subject to} \quad H_\Delta \varepsilon = 0$$

an the approximate message is recovered using:

$$\hat{u}_2 = H_2 \hat{x}_2$$

In principle, the estimate of codeword x_1 could be obtained as follows:

$$\hat{x}_1 = \arg\min_{\varepsilon} d(\mathbf{y}, \varepsilon) \quad \text{subject to} \quad H_1 \varepsilon = 0$$

where the signal received at decoder 1 is given by $\mathbf{y} = [y, \ldots, y_N]$ and H_1 is the dual of the generator matrix G_1 used to produce the original codeword x_1(i.e. $x_1 = G_1 u_1$).

However, the transmitted codeword is x_2, not x_1. This seemingly insurmountable difficulty is circumvented by the fact that a good codeword quantizer would make x_2 very close to x_1. Since H was assumed to be a good source code, x_1 may be quantized with an average Hamming distortion D consistent with the rate-distortion function below:

$$R_H = (N - K_2 - K_\Delta)/N \approx 1 - h_2(D)$$

where h_2 is the binary entropy function. From the point of view of receiver 1, codeword x_1 undergoes two transformations. The first occurs when the encoder attempts to insert the message from user 2. This quantizing step could be modeled as a BSC with cross-over probability D. The second step is when codeword x_2 is sent over the actual BSC connecting the transmitter (base station) to receiver 1 with cross-over probability p_1. Therefore, these two transformations can be modeled as two cascaded BSC with an overall cross-over probability given by $q = p_1(1 - D) + D(1 - p_1)$.

Thus, the DPC achievable capacity region is delimited by the following rate inequalities:

$$\begin{aligned}
R_1 &= \frac{K_1}{N} \leq 1 - h_2(q) \\
R_\Delta &= \frac{(N - K_\Delta)}{N} \leq 1 - h_2(p_2) \\
R_H &= \frac{(N - K_2 - K_\Delta)}{N} \geq 1 - h_2(D) \\
R_2 &= \frac{K_2}{N} \leq h_2(D) - \frac{K_\Delta}{N} \leq h_2(D) - h_2(p_2)
\end{aligned}$$

where the cross-over probability q is again given by $p_1(1 - D) + D(1 - p_1)$.

A MATLAB® model was used to represent the two-user DPC setting described above. The first step is to build the low-density graph codes based on degree distribution polynomial methods [67]. H_1 is a LDPC matrix representing a good error-control code of approximate rate 0.1 which produces a 1000-bit codeword x_1 via its corresponding generator matrix G_1. H is the dual of a LDGM G corresponding to a good source code of approximate rate 0.8 built from nesting two independently generated LDGM (H_2, H_Δ) of (approximate) rate 0.9 each. Note that the rate of the error-control code was chosen to be low in order to tolerate both the channel noise and the distortion introduced by the quantizer. On the other hand, the rate of the quantizing code was chosen to be high in order to minimize the amount of distortion. The next step is to encode the message from user 2. This message is treated by the encoder as side information and incorporated into the scheme as a constraint. By an abuse of notation, if ε is any particular solution of the constraint presented earlier:

$$\begin{bmatrix} H_\Delta \\ H_2 \end{bmatrix} \varepsilon = \begin{bmatrix} 0 \\ u_2 \end{bmatrix}$$

where all candidate codewords for x_2 that satisfy the constraint can be formulated as $x_2 = \varepsilon + Gw$. Then the objective is to minimize the Hamming weight of $x_1 + \varepsilon + Gw$, which is equivalent to minimizing $d(x_1, \varepsilon + Gw)$.

Once the global minimum is reached ($w \approx w_{min}$) via the MTP algorithm then the codeword x_2 (of length 1000) is generated according to $x_2 = \varepsilon + Gw_{\min}$. In the DPC model, the message from user 2 is also used to form a syndrome vector. This syndrome vector is then used to find a particular solution ε that satisfies the

constraint via Gaussian elimination in the $GF(2)$. The codeword x_2 is transmitted to their two recipients via a BSC with cross-over probabilities p_1 and p_2 respectively. The message at receiver 1 is recovered with nominal BP decoding using H_1. The message at receiver 2 is obtained first by regular BP decoding using H_Δ and then calculating $u_2 = H_2 x_2$. This strategy can easily be extended beyond two users by observing that a third user, for instance, would be accommodated by transmitting $x_3 = \varepsilon + Gw_{\min}$ based upon minimizing $d(x_2, \varepsilon + Gw)$ and so forth.

The achieved sum-rate capacity is calculated by setting the distortion D to the appropriate value according to the achieved rate R_H and gradually increasing (or decreasing) the cross-over probabilities p_1 and p_2 until MTP convergence is lost. The set of collected rate pairs constitutes the sum-rate capacity, which delineates the achieved capacity region.

The set of achieved rate pairs in this example are:

$$R_1 = [0, 0.0619, 0.0725, 0.0828, 0.0894, 0.0951]$$
$$R_2 = [0.8768, 0.7658, 0.7278, 0.7060, 0.3216, 0]$$

with the corresponding maximum cross-over probabilities $p_1 = 0.3132$ and $p_2 = 0.0492$.

The capacity region obtained from this set is highlighted in Figure 4.3. The dash line bordering the shaded area (i.e. capacity region) denotes the sum-rate capacity achieved with DPC using the MTP algorithm for codeword quantization. The

achieved sum-rate is compared to the approximate corner points of the achievable DPC sum-rate capacity. The achievable DPC sum-rate capacity is labeled as optimal in the figure for simplicity. In reality, however, the optimal DPC capacity would be a smooth curve through the corner points upper-bounded by the Sato bound.

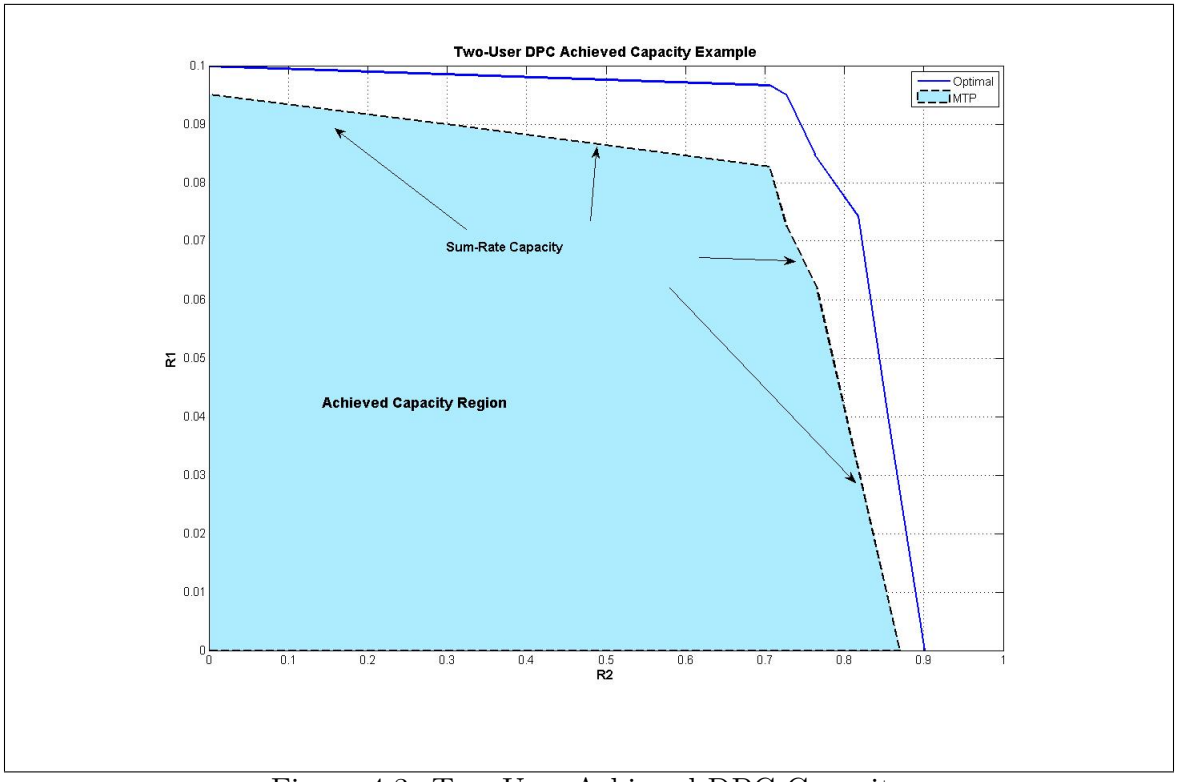

Figure 4.3: Two-User Achieved DPC Capacity

There are a number of reasons why the capacity region obtained above does not saturate the optimal bound. First, a constant and non-optimized α parameter in the MTP algorithm was used with only 50 iterations. Second, much larger block lengths ($N \gg 1000$) would be needed in order to reach capacity. Third, the code constructions used in this example do not yield the exact intended code rates. This particular problem is compounded when the resulting codes are nested, thus giving

a certain element of unpredictability to the actual rates. In addition, when a LDPC matrix or a LDGM is generated using degree distributions their corresponding dual matrices typically have high girth (i.e. not low density). This underscores the need for further advances in those code constructions in order to approach the capacity limits.

4.2 Information Embedding

The process of embedding information generally consists of placing data into a different set of data for the purpose of asserting data ownership, protecting data content, or hiding data transmission. Information embedding is a fairly broad subject, related to but different than cryptography, which includes the popular topic of digital watermarking as well as the lesser-known digital steganography [89]. The general setup for information embedding is represented by the communication channel shown in Figure 4.4.

The data to be embedded, or hidden, is the message M to be communicated secretly and it is sometimes referred to as the payload. The medium in which the payload is to be inserted is known as the cover object X (or cover image, host signal, carrier sequence, etc.) and is available only to the hider. After insertion, the modified object S is called a stego (or stego image, cipher sequence, etc.). The stego key K is only known to the sender and recipient and used to aid in the encoding and decoding process. The stego object S is subjected to possible detection and/or alterations

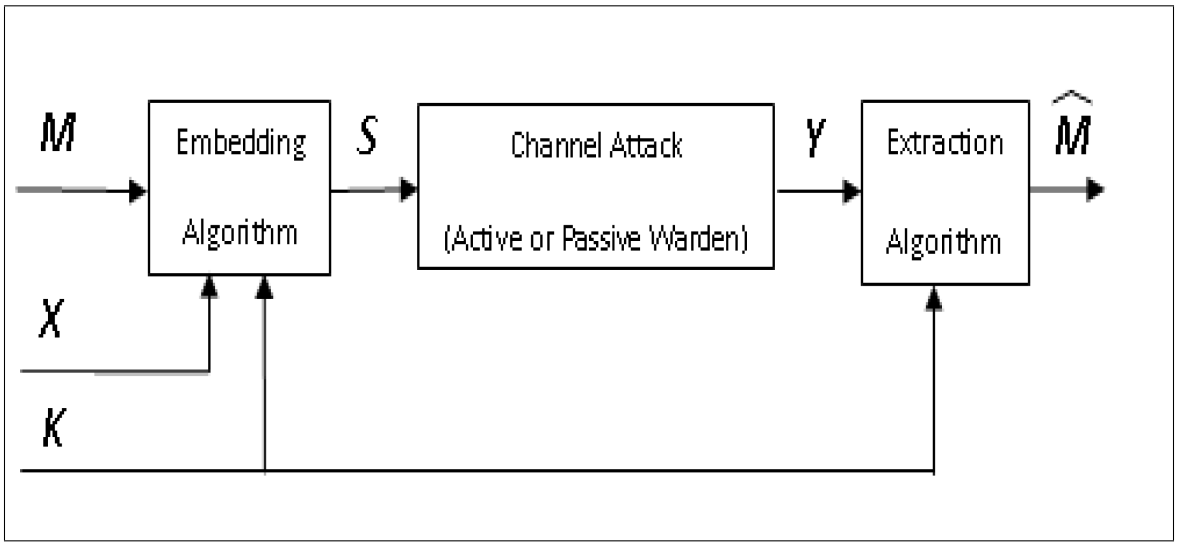

Figure 4.4: General Information Embedding Setup

(channel noise) from a presumptive attacker as it makes its way to its destination. The (possibly altered) stego object $Y \cong S$ is received by the decoder which proceeds to extract the intended message.

Many parallels between multi-user communications, DPC in particular, and information embedding, particularly digital steganography, have been identified [61]. This opens up the possibility of using codeword quantization techniques to enhance the embedding process and attain capacity. Thus, the focus in the subsequent example is on source coding algorithms applied to digital steganography.

4.2.1 Steganography

Steganography refers to the covert transmission of information via overt communication channels [71]. It differs from both watermarking and cryptography in the sense

that the main priority is to conceal the existence of hidden information rather than to protect the hidden information itself. The advent of personal computers has shifted the attention from classical steganography, known and used for many centuries, to modern steganographic techniques used to hide data in digital media such as audio, video, or image files (i.e. digital steganography) [71]. There is a large variety of both free and commercially available software to perform data-hiding functions on virtually any kind of digital media [90]. Nevertheless, theoretical insights about information embedding capacity and its fundamental limits as well as capacity achieving techniques are not yet widely understood. This not only pose risks to well-mounted attacks that exploit system vulnerabilities but also underscores the need for more effective embedding techniques attuned to recent information-theoretic advances applicable to this challenge.

The famous article by G.J. Simmons regarding the Prisoners Problem marked the turning point of steganography as a scientific discipline [91]. In that setting, Alice and Bob are in prison and poised to plan an escape. Their communications are being monitored by the warden, Willie. If Willie sees any suspicious (possibly encrypted) messages, they will be placed in solitary confinement and their escape plan becomes thwarted. Hence, they need to devise a strategy to secretly communicate with each other about their plan through seemingly harmless messages. It is assumed that Willie has knowledge about the strategy but not about a message security-enabling key that Alice and Bob managed to share prior to being imprisoned.

The analogy posed by the prisoners problem helped to re-shape the steganography problem in the framework necessary to develop a more comprehensive under-

standing about its fundamental limits and underlying assumptions [71]. For instance, the use of a public key versus a private key can have a significant impact on the embedding capacity. So does the behavior of the warden (i.e. passive or active). Another important assumption is that the warden has full knowledge of the embedding mechanism, but not the key. This is in concert with the well-known Kerckhoffs principle from cryptology where true security does not lay with the particular encryption method but with the knowledge about the key used to encrypt and decrypt the message [92].

4.2.2 Embedding Capacity

A crucial aspect of information embedding is the question of how much data can be hidden in a cover object. The answer came from the remarkable observation that a duality exists between information embedding and source coding with side information [61]. Referring back to the block diagram depicted in Figure 4.4, if the distortion between the cover object X and the stego object S is constrained to d or less, then the specific question becomes: what is the maximum rate of reliable communication supported by the embedding algorithm for a particular transmission channel characterization?

The characterization of the information embedding capacity is essentially drawn from a generalization of the informed encoder channel case with defective memory [93, 94]. This capacity relationship is modified in this case to add an arbitrary distortion constraint and a metric [61, 95]. The resulting information embedding (IE)

capacity expression $C^{IE}(d)$ is given below following the notation in [71]:

$$C^{IE}(d) = \sup[I(Y;U) - I(U;X)]$$

where the supremum is taken over all probability distributions $p_{UX}(ux)$ and functions $f : \mathcal{U} \times \mathcal{X} \rightarrow \mathcal{S}$ which satisfy $E[D(X,S)] \leq d$ for $S = f(U,X)$ where $D(\cdot,\cdot)$ is an arbitrary distortion metric and U is an auxiliary random variable. The relationships among the different variables is illustrated by the fact that the auxiliary random variable U forms the following Markov chain: $U \rightarrow (X,S) \rightarrow Y$.

The general information embedding capacity expression shown above can be further specified in the context of the BSC. In this case, the distortion function d is defined to be the Hamming metric and the cover object is modeled as a Bernoulli source with cross-over probability of 0.5. Under these conditions, the capacity expression becomes [61]:

$$C^{IE}(d) = \begin{cases} \frac{c_p^{IE}(d_p)}{d_p} d, & \text{if } 0 \leq d \leq d_p \\ c_p^{IE}(d), & \text{if } d_p \leq d \leq 1/2 \end{cases}$$

where $d_p = 1 - 2^{-h(p)}$, $h(\cdot)$ is the binary entropy, and $c_p^{IE}(d)$ is an upper concave envelope function given by:

$$c_p^{IE}(d) = \begin{cases} 0, & \text{if } 0 \leq d \leq p \\ h(d) - h(p), & \text{if } p \leq d \leq 1/2 \end{cases}$$

Therefore, the information embedding capacity on a BSC is upper-bounded by the concave function defined above. Thus, it sets the fundamental theoretical limits on the amount of data that can be inserted in a digital cover file under specified distortion constraints. Acceptable distortion constraints cannot be strictly defined on a mathematical basis alone. Perceptual constraints are perhaps of more interest in the case of digital imagery or audio (as seen in the subsequent example), though subjective in the sense that it may not always correlate well with mathematical constraints.

4.2.3 Embedding Techniques

A number of embedding techniques have been developed over time drawing from the rich parallels with multi-user communication schemes such as spread spectrum [96]. These approaches appear attractive due to their relative effectiveness and implementation simplicity. Nevertheless, they seem to fall short of attaining the predicted maximum embedding capacity. As such, the attention has recently turned to alternate strategies such as those inspired by DPC. As mentioned in section 4.1, DPC-based techniques have the distinct advantage of being capacity-achieving; however, their implementation could be cumbersome. Two practical techniques, namely wet paper coding and matrix embedding, are discussed next whose aim is to maximize the embedding capacity.

4.2.3.1 Wet Paper Coding

Wet paper codes are analogous to the dirty paper codes invoked by M. Costa in the sense that the embedding process is akin to writing on the dry spots of a wet piece of paper [72, 97]. Once the wet paper dries up, it is difficult to tell which spots were actually altered. This covertness feature is very attractive from a steganographic point of view. More specifically, the sender is constrained to insert the message into a subset of the cover object. This subset is chosen according to an arbitrary selection rule which is only known to the sender and not shared with the intended recipient. The recipient proceeds to extract the message without knowledge of the location of the altered spots in the object or how they were chosen. This communication setup represents an instantiation of a channel with defective memory cells [98].

Given a cover object X with n elements the selection rule picks (either deterministically or randomly) q elements which could be modified by the embedding procedure out of the set of indexed elements $[1 \ldots n]$, where $q < n$ [97]. The cover object X (or host sequence) is passed through a publicly-known parity function to obtain the corresponding n-element carrier sequence. This sequence becomes the stego object S after message insertion and sent over the channel to the recipient. The recipient re-applies the parity function to obtain the sequence of altered bits and then uses the secret key K to recover the message M according to $M = KS \cong KY \pmod 2$, per Figure 4.4. Again, this secret key is only known to the sender and recipient.

The capacity obtained with this approach is essentially q/n and the average distortion is given by $0.5q/n$ [98]. The wet paper coding strategy boasts enhanced

security against passive warden detections but it comes at the expense of lower embedding capacity. A slightly modified approach allows for the selection of a greater number of cover object elements to alter such that $q \leq l \leq n$ [99]. This, in general, yields an embedding capacity that approaches the theoretical limits.

4.2.3.2 Matrix Embedding

The matrix embedding technique is very similar to the wet paper coding technique but instead it uses nested codes and additional constraints. Starting with a cover object X of length n and a message M of length q, with $q < n$, the key K is a $q \times n$ parity check matrix chosen such that the cipher sequence S satisfies the constraint $M \equiv KS \pmod 2$. Thus, only the recipient is able to recover the message since it is the only other party that knows the unique secret key. Also, suppose that V_0^K is the set of null vectors V defined in the $GF(2)$ for matrix K such that:

$$V_0^K = \{\mathbf{0} \equiv KV \pmod 2\}$$

and the coset of vectors that produces the syndrome M (by intentional abuse of notation) V_M^K is defined by:

$$V_M^K = \{M \equiv KV \pmod 2\}$$

Hence, in the $GF(2)$ domain, the matrix embedding approach attempts to find a stego object S such that the Hamming distance $d(X, S)$ between the cover object X and the stego object itself is minimized subject to the constraint $M \equiv KS \pmod 2$

[61, 95, 100].

The immediate observation is that the structure of the problem presented by this approach is strikingly similar to DPC, exposed earlier in section 4.1. In fact, the problem reduces to performing binary quantization on the carrier sequence X to get it as close as possible to a vector in the coset V_M^K. If the vector $v_K(M)$ is the coset leader of V_M^K, then:

$$S \equiv X + v_K(M)$$

The host signal X is quantized to the rate $R_X \geq H(X) - H_2(D)$, where $H(X)$ represents the per-bit entropy and $H_2(D)$ is the binary entropy of the average distortion D. If the chosen parity function is able to generate a uniformly-distributed sequence X, then the rate equation above simplifies to:

$$R_X \geq 1 - H_2(D)$$

which is the traditional rate-distortion function expression presented before. The matrix embedding method could attain, in principle, the maximum embedding rate of q/n by carefully selecting capacity-approaching low-density source and error-correcting codes (i.e. LDGM and LDPC codes) and using them in a nested structure as described in the DPC example earlier. By the same token, the same codeword quantization techniques applied to the DPC example before can be applied to information embedding as shown in the example in section 4.2.4.

4.2.4 Digital Image Steganography Example

This example involves the succinct modification of a still image in order to embed a message. The input image, shown in Figure 4.5, is the famous cameraman photograph. This image is represented by a 256×256 array of pixels with an 8-bit gray-scale per pixel stored in the Tagged Image File (TIF) format.

Figure 4.5: Original Cameraman Image

The first step is to extract the chosen bit plane in which to store the message

among the eight bit planes available in the image. The Least Significant Bit (LSB) plane is typically selected and constitutes the simplest parity function available. However, this choice appears to be more succeptible to detection since the attacker might also be expecting the LSB to be chosen for hiding data [102][103]. Thus, the second LSB plane was chosen for this example for two reasons. The first is to reduce the risk of payload detection by either a passive and/or an active warden. The second is to demonstrate that the employed embedding technique is still able to imperceptibly insert data in more important bit locations throughout the image than just the LSB.

The second step is to build a code composed of a low-rate error-correction code nested in a high-rate source code using the identical procedure discussed in the DPC example in section 4.1. The rate of the nested error-control code is 0.2 and produces codewords of 534 bits in length. The rate of the overall source code is 0.6.

The third step is to generate a 534-bit carrier sequence via a carefully-selected parity function with the purpose of creating a bit sequence with high entropy [71]. This has the effect of reducing the rate even further as shown in section 4.2.3.2. The parity function used in the example essentially computes each bit in the carrier sequence as the XOR of three spatially-separated, randomly chosen bits along the selected bit plane.

The MTP algorithm attempts to quantize the carrier sequence into a minimum weight vector which is then used to produce a stego sequence that meets the same constraint of $M \equiv KS \pmod 2$ but is also part of the coset of the source code generated in the second step above. This binary vector is the 534-bit cipher sequence

which contains the message to be inserted in the image. The message is simply a 107-bit sequence randomly generated from a Bernoulli source of equally probable bits. Figure 4.6 shows the cameraman image containing the stego object.

The cipher sequence is embedded into the second LSB plane by randomly chosing the pixel locations (indexes) along the plane. This is sometimes called the inverse parity check function. Again, these random pixel locations are shared between the sender and the recipient. After the embedding process is complete, the selected bit plane is placed back into the image and the modified image is sent over the channel.

Note that the differences between the images in Figures 4.5 and 4.6 are imperceptible to the naked eye which underlines the early success of the embedding algorithm. The robustness of the embedding procedure is put to the test by subjecting the modified image to an attack mounted by an active warden. It is assumed that the warden is able to detect the possible presence of hidden data in the second LSB plane but not the specific locations of the altered bits. Hence, the attack is modeled as passing the entire second LSB plane through a BSC with bit flip probability of 0.1. The post-attack image is shown in Figure 4.7. Note that the image now shows a few scattered white spots across the photograph.

The message recovery process begins by collecting the modified bits from the pixel locations chosen by the encoder from the second LSB plane. The message is easily obtained using the regular BP algorithm to decode the low-density code generated in the second step above.

Figure 4.6: Modified Cameraman Image

The question that remains is to determine how close the embedding procedure gets to achieving the presumptive theoretical embedding rate of $q/n \approx 0.2$ for this example. The average distortion was computed by replacing it with the empirical mean of the distortion between the host and cipher sequences over 100 iterations. The calculated mean was approximately 0.1511. To determine the rate, the message length was lowered bit by bit, from a maximum of 107 bits until no discernible differences between the original and modified images were observed. The achieved embedding rate was $95/534 \cong 0.1779$. The plot in Figure 4.8 shows the specific attained rate

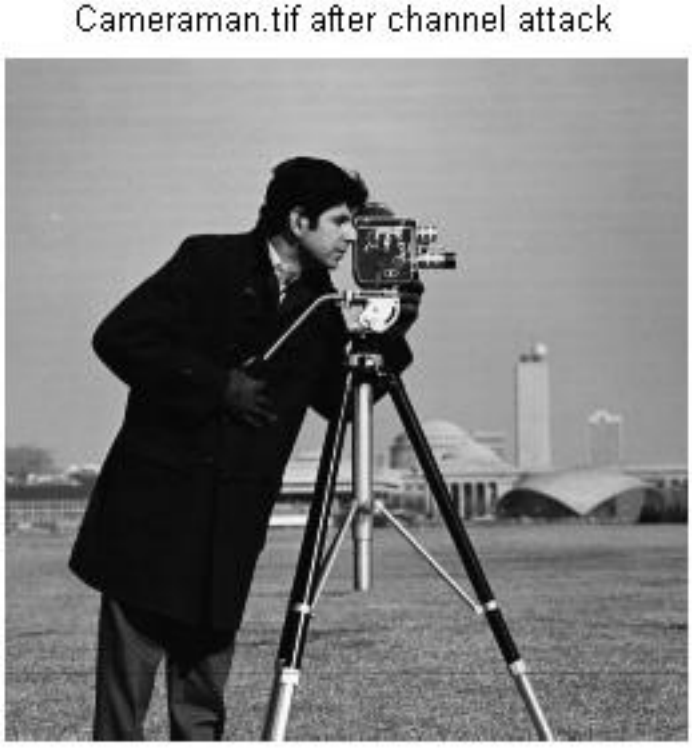

Figure 4.7: Modified Cameraman Image after the Chanel Attack

compared to the theoretical bound. This is very good performance considering the relatively short length of the cover signal in the example and the fact that it is more difficult to achieve capacity when low distortion is desired. Also, the polynomial degree distributions used to generate low-density codes do not yield codes with the exact desired rate as seen in the previous DPC example.

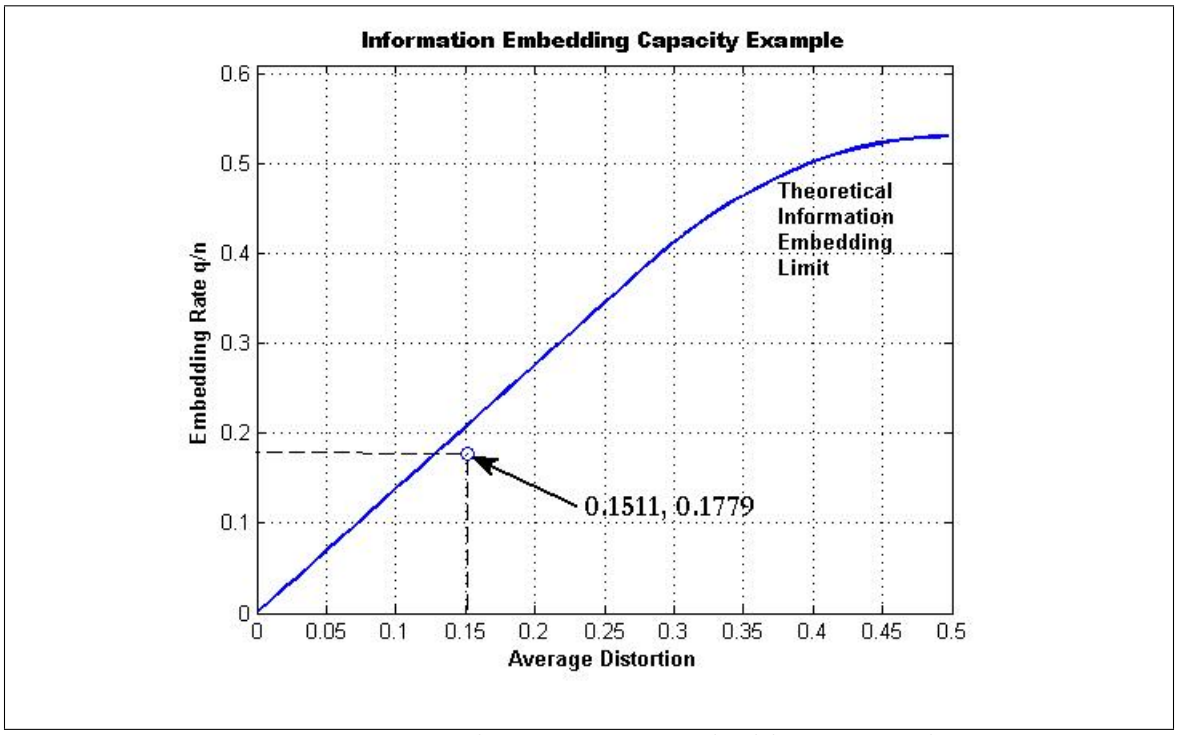

Figure 4.8: Digital Image Data Embedding Example

4.3 Information Secrecy

In many ways, information secrecy (also known as information-theoretic secrecy, information-theoretic security, or perfect secrecy) could be seen as an extension of the information embedding case conditioned on intercepted messages between Alice and Bob. An eavesdropper named Eve has tapped into their communication channel and detected their messages. The challenge for Eve, though, is to successfully decipher (i.e. decode) these messages.

In contrast to information embedding, where the main question is how much data can be covertly hidden given a maximum level of allowed distortion, information

secrecy is concerned with how much data can be securely transmitted while guaranteeing that an eavesdropper will not have more than a random chance of successfully decoding the data. More specifically, once Eve intercepts a message between Alice and Bob the probability that Eve decodes the message should ideally be uniform across all possible (right and wrong) outcomes [101].

This concept differs from the notion of security afforded by cryptographic means which rely heavily on computational intractability in order to curtail the ability of an eavesdropper to decrypt the data [102, 103]. The level of security provided by information secrecy goes beyond intractability, which can become obsolete by eventual advances in algorithms and/or computer hardware. In fact, information secrecy involves looking at the problem posed by cryptography from an information-theoretic point of view rather than an algorithmic complexity point of view.

The example shown subsequently in section 5.2.4 is synergistic with the renewed interest in this problem motivated by the widespread awareness of the inherent vulnerabilities of wireless networks due to their open nature and the increasing need to integrate information security as a fundamental aspect in system design [102, 103].

4.3.1 Perfect Secrecy and Equivocation

The basic tenets of perfect secrecy were laid out by C.E. Shannon in 1949 [101]. In this seminal work, Shannon defined perfect secrecy as the requirement that the a posteriori probabilities of a cipher (denoted by E) be equal to the a priori probabilities

of the underlying message (denoted by M). This could be expressed as follows:

$$P_E(M) = P(M)$$

where $P_E(M)$ is the a posteriori probability of message M if the encrypted message E is intercepted and $P(M)$ is the a priori probability of the message M. Shannon also further developed the concept of equivocation, defined in terms of the following (conditional) entropies [53, 101]:

$$H_E(K) = \sum_{E,K} P(E,K) \log P_E(K) = H(M) + H(K) - H(E)$$
$$H_E(M) = \sum_{E,M} P(E,M) \log P_E(M) = H(K) + H_M(E) - H(E)$$

where $H_E(K)$ and $H_E(M)$ are the conditional entropies of the key and message respectively, $P(E,K)$ and $P(E,M)$ are the joint probabilities of the cipher and key, and the cipher and message respectively, and finally $P_E(K)$ is the a posteriori probability of the key. Note that zero equivocation implies that one of the received (intercepted) messages (or key) has unit probability of occurrence while the others have zero probability of occurrence. Typically, the equivocation function tends to decrease as the number of intercepted ciphers increase. Nonetheless, perfect secrecy ideally means that:

$$H(M|E) = H(M)$$
$$H(K|E) = H(K)$$

Therefore, Eve has learned the same information about the message by intercepting the cipher as if the cipher had never been intercepted in the first place. In other words, Eves chances of deciphering the message did not improve by intercepting the cipher. The basic situation described thus far is depicted in Figure 4.9 below: The underlying

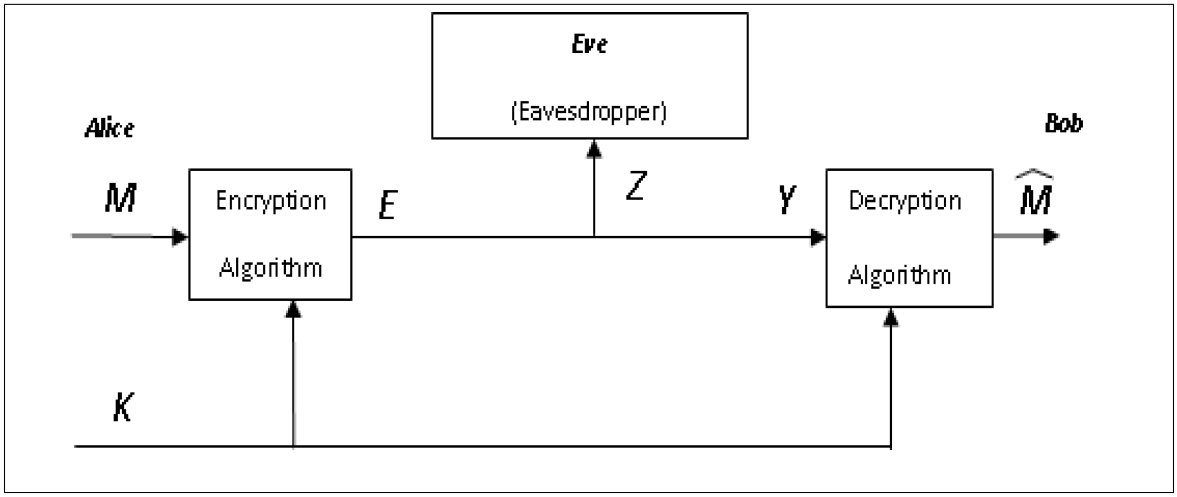

Figure 4.9: Basic Information Secrecy Setup

assumption (as with information embedding) is that Eve has full knowledge about the encryption/decryption mechanism but knows nothing about the key shared between Alice and Bob. Further assuming that the message consists of n binary elements, then the encryption scheme chooses a cipher (codeword) E among 2^{nR} possibilities according to the chosen code rate R. The key K is then selected among 2^{nR_0} choices. Hence, the fundamental result by Shannon states that the message length is limited by the length of the key. Moreover, the chosen rate for the key R_0 (and hence R) has to be greater than the entropy of the message M in order to ensure perfect secrecy, such that [101]:

$$R_0 \geq R = H(M)$$

4.3.2 Secrecy Capacity

The perfect secrecy criterion developed by Shannon and presented in section 4.3.1 establishes the fundamental rate limits for any type of crypto (secrecy) system. However, just like in his mathematical theory of communication systems, no practical code constructions were given to achieve these rate bounds. A big step towards developing a more specific secrecy capacity expression was made by the introduction of the wire-tap channel [104, 105]. The setup shown in Figure 4.9 assumes no alternate channel for key sharing. The single-letter capacity expression is then given by [104, 105]:

$$C = \max_{U \to E \to (Y,Z)} [I(U;Y) - I(U;Z)]$$

where U is the auxiliary variable forming the Markov chain: $U \to E \to (Y, Z)$. An important simplification occurs if the eavesdropper channel is assumed to be degraded compared to the receiver channel. The resulting equation is:

$$C = \max_{P_E(E)} [I(E;Y) - I(E;Z)]$$

where the mutual information difference is maximized over the probability of the channel input (i.e. cipher). The capacity expression above constitutes the rate limit at which Alice (or Bob) can communicate with Bob (or Alice) under perfect secrecy conditions (i.e. high equivocation by Eve). Multiple extensions of this capacity result have since been developed [102, 103, 106, 107].

The expressions for secrecy capacity and embedding capacity appear to be

strikingly similar. This peculiarity feeds the earlier notion about information secrecy being an extension of the information embedding given their parallel objectives of concealing/protecting information from third parties. This observation also implies that nested coding strategies similar to those employed for DPC and data hiding can be readily applied to this problem as shown in the subsequent example. The remaining concerns about key sharing-agreement along with how to induce a degraded eavesdropper channel are addressed next.

4.3.3 Secrecy Coding

Among the many ways that legitimate users can induce a degraded channel for an eavesdropper the most well-known is the one-time pad [108]. The one-time pad is a classical encryption method in which modular addition is performed between the intended message and a random sequence generated from a Bernoulli source of equally probable bits ($p = 0.5$). This random sequence is available exclusively to the legitimate users (i.e. Alice and Bob only). This technique not only provokes obscurity in the intruders channel but is also proven to achieve perfect secrecy [101]. However, even though the one-time pad is able to attain theoretical perfect secrecy this technique is seldom used in practice for various reasons ranging from the impossibility of generating truly random sequences to the difficulties in exchanging both the pad and the key. Hence, other encryption alternatives must be pursued.

A breakthrough approach to circumvent these practical difficulties is based on the idea of two-way protocols over a public channel [109]. These protocols are nothing

more than secret key agreement methods over public channels to ensure that any eavesdropper ends up with worst (degraded) channel conditions than any legitimate user. Notionally, the procedure begins with Bob generating a random sequence X and sending it un-coded to Alice (and Eve). Alice receives $X + W_b$, where W_b is the channel noise between Bob and Alice. In turn, Eve gets $X + W_e$, where W_e is the noise in Eves channel. Alice proceeds to generate her own random sequence V and adds it to X. Alice now sends the coded sequence $X + V + W_b$ back to Bob (and Eve). Since Bob knows X his received sequence can be reduced to $V + W_b$ from which he can recover V after decoding. Eve, instead, receives $X + V + W_b + W_e$ which can only be reduced to $V + W_b + W_e$. By simple inspection, Eves sequence contains more noise than Bobs, thus effectively inducing a degraded channel.

This protocol can be adapted to nested codes as shown in [110, 111]. Suppose that two linear block codes with parity check matrices H_{ab} and H_e are nested such that:

$$H_e = \begin{bmatrix} H_{ab} \\ H_k \end{bmatrix}$$

for some H_k chosen so that $K = H_k V$, with K representing the key as before. Then, instead of Alice sending her coded random sequence V, she generates a syndrome $S = H_{ab}V$ and sends it to Bob with coding for error correction. Bob receives the syndrome S and attempts to decode the sequence V generated by Alice using S as

side information according to:

$$\hat{V} = \arg\min_{Y} d(Y, V + W_b) \quad \text{subject to} \quad S = H_{ab}Y$$

where Y is the sequence received by Bob. If H_{ab} is the parity check matrix of a capacity-approaching code, then there is a very high likelihood of $\hat{V} \approx V$. At this point, Bob (and Alice) may proceed to recover the key from $K = H_k V$. The secrecy assurance stems from the specific attributes of H_e. If H_e is also a capacity-approaching code, then it is able to operate very close to the following rate limit given by [109]:

$$R_e = h(p * q) - h(p)$$

where the channel between Alice and Bob is modeled as a BSC with cross-over probability p, and the channel between Alice (or Bob) and Eve is also a BSC with cross-over probability q, assuming binary variables and binary entropies. The argument $p * q$ in the first entropy term is the convolution of the two cross-over probabilities representing the two cascaded BSC as described in section 4.1.

If the crypto system is operating at or near R_e, then the following two inequalities apply [110]:

$$I(K; S, Z) \leq \delta$$

$$h(p * q) - h(p) - H(K) \leq \delta$$

for $\delta > 0$ and Z the sequence intercepted by Eve. The first inequality means that the

closer the crypto system operates to the rate limit, the smaller the mutual information between the key and the data available to Eve becomes. In other words, Eve will not be able to derive the key from the intercepted data and, thus, perfect secrecy is achieved. The second inequality shows the direct relationship between the entropy of the key and the theoretical secrecy rate limit presented earlier.

Moreover, suppose that instead of using the protocol described above to agree on the key, Alice and Bob decide to use it for sending each other actual data. Then the problem can be re-formulated as follows:

$$\begin{bmatrix} 0 \\ M \end{bmatrix} = \begin{bmatrix} H_{ab} \\ H_k \end{bmatrix} V$$

where Alice now has to generate the sequence V not arbitrarily but according to this condition. The rest of the steps are identical to the procedure already described. The end result is the retrieval of the message M according to $M = H_k V$. The rate at which Alice can communicate with Bob with unconditional secrecy under this particular setup is given by [112]:

$$R'_{ab} = [h(p * q) - h(p)][1 - h(p)]$$

where it is assumed that Eves channel is not initially degraded and the second bracketed term corresponds to the error-correcting code applied to the sequence sent by Alice.

4.3.4 Information Secrecy Example

The secrecy coding scheme leverages many aspects of the two-user DPC model described previously. H_{ab} is a 54×541 LDPC matrix with approximate rate of 0.9 produced with degree distribution polynomials [67]. H_e has an approximate rate of 0.8 and is built by nesting the LDPC matrices H_{ab} and H_k of (approximate) rate 0.9 each with the hope that H_e remains a LDPC matrix. The matrix H_k is simply a column-permutated version of H_{ab}. It is also assumed that the dual of H_e (i.e. G_e) produces a good LDGM code.

Both the sequence X received from Bob and the message M from Alice are randomly generated from a Bernoulli binary source with probability of 0.5. The message M is treated by the encoder as side information and incorporated into the scheme as a constraint. If ε is any particular solution of the constraint presented earlier [112]:

$$\begin{bmatrix} H_{ab} \\ H_k \end{bmatrix} \varepsilon = \begin{bmatrix} 0 \\ M \end{bmatrix}$$

where all candidate ciphers (codewords) for V that satisfy the constraint above can be formulated as $V = \varepsilon + G_e w$. Then the objective is to minimize the Hamming weight of $V + \varepsilon + G_e w$, which is equivalent to minimizing $d(V, \varepsilon + G_e w)$.

Once the global minimum is reached ($w \approx w_{\min}$) via the MTP algorithm then the codeword V of length 541 bits is generated according to $V = \varepsilon + G w_{\min}$. The

message M from Alice is also used to form a syndrome vector. This syndrome vector is then used to find a particular solution ε that satisfies the constraint via Gaussian elimination in the $GF(2)$ domain. The cipher V is transmitted to Bob via a BSC with cross-over probability p. Bob recovers the message first by regular BP decoding using H_{ab} and then calculating $M = H_k V$. This strategy can easily be extended beyond two legitimate users by observing that a third legitimate user, say Carol, would be accommodated by assuming that the sequence X received from Bob has already been agreed upon separately between Bob and Carol. Alice can now broadcast her message to both users in secrecy from Eve [112].

The achieved secrecy rate capacity shown in Figure 4.10 is obtained by setting the cross-over probability p of the Alice-Bob channel to the minimum value according to the actual rate achieved by the parity check matrix H_{ab} and then gradually increase the cross-over probability q until convergence with MTP is lost. The procedure is repeated by increasing p from its minimum and increasing q again until convergence is lost. The maximum probability q where convergence is lost varies with each minimum probability p.

The set of collected rate pairs $(R2, C_s(R2))$ constitutes the secrecy rate capacity, which delineates the achieved secrecy region. This secrecy capacity is a function of the attained rate in the Alice-Eve channel $(R2)$ as exposed earlier. The MTP algorithm once again shows good performance given the limitations imposed by the short block lengths and nested codes built for this example whose actual rates fall below the intended ones.

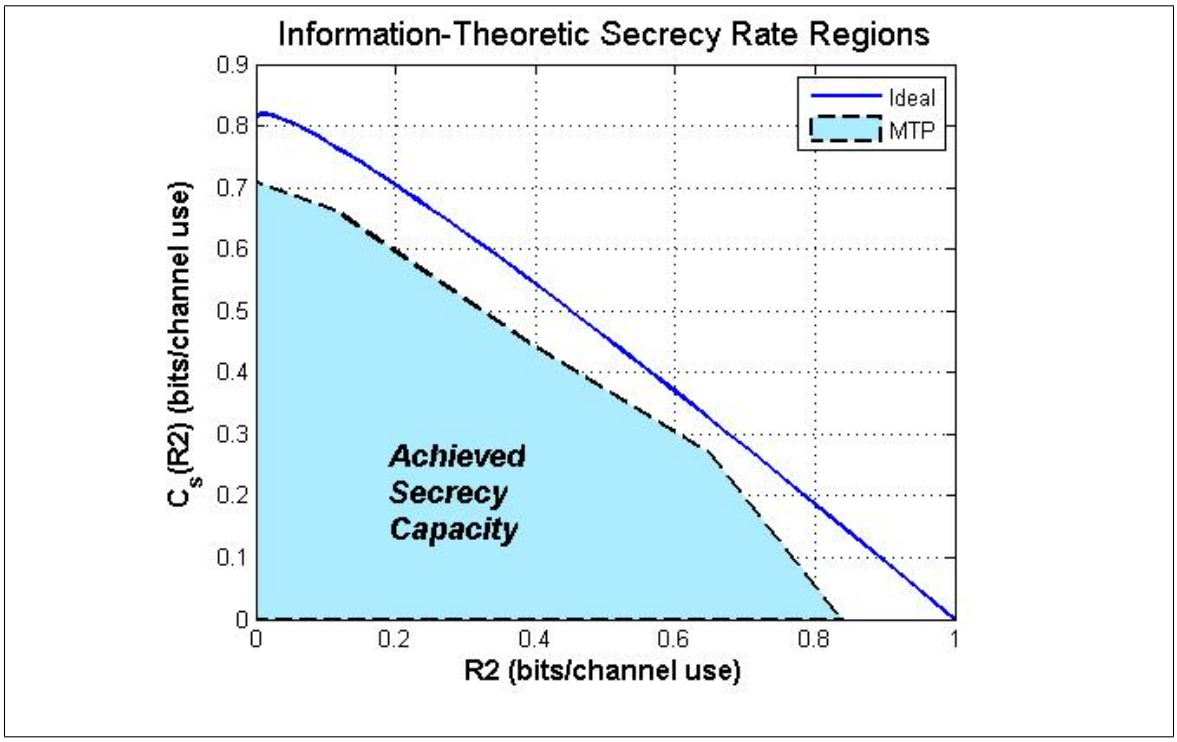

Figure 4.10: Information Secrecy Capacity Example

The set of achieved rate pairs in this example are:

$$C_s(R2) = [0, 0.2711, 0.4525, 0.6621, 0.7194]$$

$$R2 = [0.8464, 0.6581, 0.3978, 0.1260, 0]$$

with the corresponding minimum cross-over probability $p = 0.2215$.

4.4 Distributed Information Sharing

A potentially new application of the source coding techniques discussed thus far has emerged in the context of wireless sensor networks. Suppose that a wireless (ad hoc) network of sensors distributed in a relatively large space is used to monitor certain conditions across that space. Each sensor collects a fair amount of raw measurements which need to be post-processed somehow in order to extract any useful information out of the network. The naive approach would be for each sensor to haul its own data to a centralized node responsible for processing all the network data. The large amount of incoming/outgoing data could easily overwhelm the network deeming it useless for extended periods of time. A common approach to deal with this problem is to relay the data from sensor node to sensor node (multi-hop) where each sensor takes the incoming set of measurements and combines it with its own to create an aggregate set [113, 114]. This set is then sent to the next node where the process is repeated until it gets to the centralized node. For large sets of measurements, the aggregation scheme could also become problematic from a communications, computational, memory storage, and energy efficiency standpoint.

This section focuses on the dilemma of finding alternative efficient means of sharing information in a distributed (multi-terminal/multi-sensor) environment. A realizable coding scheme that allows more efficient data diffusion is identified by drawing from the structure of the previous examples. The scheme is also evaluated and compared against known information-theoretical bounds via a simple three-node example.

4.4.1 Wireless Sensor Networks

The study of wireless sensor networks (WSN) is a very broad and multi-disciplinary subject encompassing electronics, embedded systems, computer networking, communications theory, and computer science among many other scientific fields. A wireless sensor network is essentially a set of autonomous spatially-distributed devices (nodes) used to measure/monitor the conditions of their surrounding environment and collaborate to pass and/or process their collected data for meaningful information extraction [115]. The measured quantities could be temperature, pressure, vibration, or multiple others. They were originally conceived to be employed in military battlefields but additional applications are continuously on the rise such as industrial process monitoring, traffic control, and air pollution monitoring. A generic WSN is shown in Figure 4.11 composed of 12 nodes spread out over a certain area.

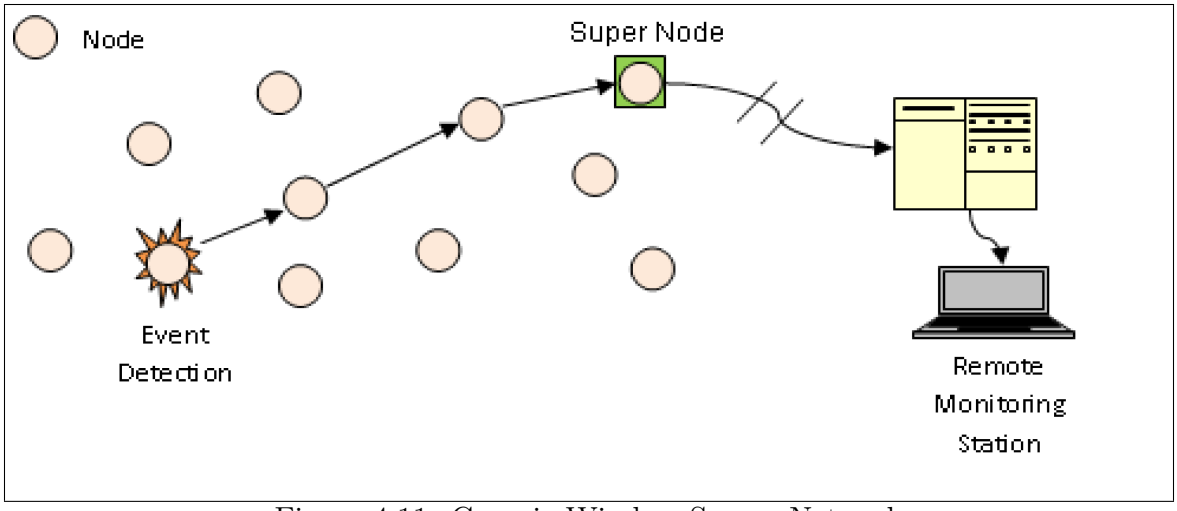

Figure 4.11: Generic Wireless Sensor Network

Once a node (or a group of nodes) detects an event or collects measurements

it proceeds to send this data to one of the nodes in the group designated as the super node. This node is distinct from the others in that it has the ability to handle external queries and perform certain post-processing functions on behalf of the group (i.e. count, sum, average, etc.) [115]. The data collected by the nodes reaches the super node by relaying it (multi-hop) through intermediate nodes. The two most common routing schemes are peer-to-peer and multicast, although many adaptive techniques have also been proposed [115]. The communications, handshake protocols, and network-sensor interface/interaction typically comply with one or more networking standards such as WLAN (IEEE 802.11), Bluetooth (IEEE 802.15), Zigbee (IEEE 802.15.4), IEEE 1451, Token Ring (IEEE 802.5), or others. The topology of these networks usually follows either one or a combination of the usual configurations: star, ring, bus, mesh, or fully connected [115]. In addition, many commercially-available WSN have the ability to self-organize and know their relative sensor positions (localization). These features are important in applications requiring ad hoc connections between sensors. Some possess aggregators (super nodes) with advanced data fusion functions and the ability to make rudimentary decisions [115, 116].

The size and cost of each sensor vary dramatically but the current trend is towards lightweight and low-cost devices such as Micro-Electro-Mechanical Systems (MEMS). These two factors have a direct impact on energy consumption, memory, computational power, and communications throughput [115, 117]. Network and sensor security and reliability have also become critical considerations as these systems are now often deployed in harsh environments and possibly subject to attack/tamper by external entities [118].

Thus far, the discussion assumed a network architecture centralized around the super node. While useful, such centralized architectures are also vulnerable to super node unavailability and failure. Distributed architectures improve reliability since any node can replace the super node in case of failure and monitoring stations are able to interrogate any node. Data fusion functions may not necessarily be exclusive of the super node. In fact, this approach is preferred since it has been shown that allowing the nodes to perform rudimentary (aggregate) operations on external data could result in reduced inter-network data traffic as well as considerable savings in sensor energy consumption [113, 114]. The coding technique exposed in the subsequent example could potentially afford further savings in the sense that even less data would need to be exchanged between nodes in order to perform these aggregation tasks.

4.4.2 Coset Encoding

The coset encoding technique is equivalent to nested coding (or binning) presented in the preceding examples. The basic ideas behind the code partitioning into coarse and fine codes remain unchanged [112, 111]. The only aspect that is different is the context. Suppose that a small network composed of only three sensors has collected some information of interest (in binary form) about the space where they are deployed as shown in Figure 4.12. Furthermore, assume that the particular application requires each node to know the measurements of the other two nodes. Aside from the obvious inefficiency, the transmission capacity of the network can be quickly overwhelmed if all the nodes attempt to send each others measurements or if the number of nodes increases. Hence, an alternate strategy would be for node 1 to take its set of N

measurements x_N and calculate the syndrome:

$$s = Hx_N$$

where H is the $(N-K) \times N$ parity check matrix of a linear block code $\mathcal{C}$ known to all the nodes.

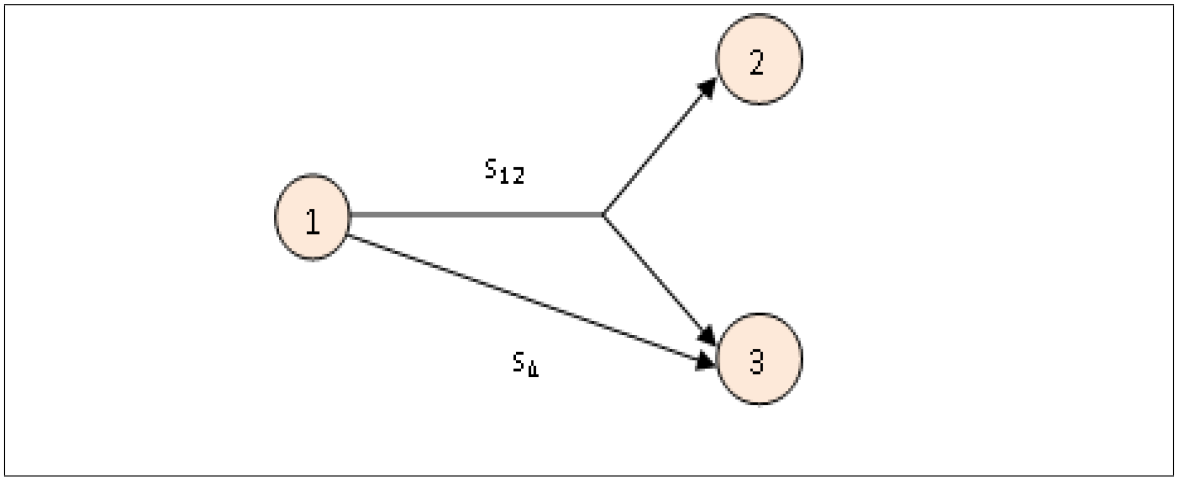

Figure 4.12: Three-Node Sensor Network

Node 1 proceeds to send just this syndrome ($N-K$ bits) to both nodes 2 and 3 rather than its entire set of measurements (N bits). Node 2 can then find an estimate of x_N according to:

$$\hat{x}_N = \arg\min_{\varepsilon} d(y_N, \varepsilon) \quad \text{subject to} \quad H\varepsilon = s$$

where ε is a binary vector candidate to belong in the coset of $\mathcal{C}$ and y_N is the set of measurements taken by node 2 under the assumption that they are correlated to x_N. If the channel characteristics between nodes 1 and 2 are the same as those between

nodes 1 and 3 (i.e. $H(x|y) = H(x|z)$, where $H(\cdot|\cdot)$ is the conditional entropy), then node 3 is able to obtain $\hat{x}_N$ via a common message (same side information s) as:

$$\hat{x}_N = \arg\min_{\varepsilon} d(z_N, \varepsilon) \quad \text{subject to} \quad H\varepsilon = s$$

where z_N is the set of node 3 measurements. On the other hand, if the channels are characterized differently, then two different code rates need to be used. This can be accomplished (just as before) by partitioning the parity check matrix H as follows:

$$H = \begin{bmatrix} H_{12} \\ H_{\Delta} \end{bmatrix}$$

assuming that $H(x|y) > H(x|z)$. As such, node 2 can recover x_N via the following:

$$\hat{x}_N = \arg\min_{\varepsilon} d(y_N, \varepsilon) \quad \text{subject to} \quad H_{12}\varepsilon = s_{12}$$

while node 3 is able to estimate x_N by:

$$\hat{x}_N = \arg\min_{\varepsilon} d(z_N, \varepsilon) \quad \text{subject to} \quad H\varepsilon = \begin{bmatrix} s_{12} \\ s_{\Delta} \end{bmatrix}$$

Note that extensions beyond the three-node case are straightforward by further par-

titions of the parity check matrix H:

$$H = \begin{bmatrix} H_{12} \\ H_{\Delta_3} \\ \vdots \\ H_{\Delta_T} \end{bmatrix}$$

and following the same steps described before. Under the three-node setup, node 1 broadcasts the vector s_{12} to both nodes 2 and 3, but unicasts (sends) the vector s_{Δ} only to node 3 as shown in Figure 4.12. The minimum theoretical rate necessary to estimate a source sequence X by using a correlated sequence Y as side information at the decoder is given by the conditional entropy $H(X|Y)$ [119]. In practice, this estimation problem is solved with high probability via message-passing decoding with side information when the linear code $\mathcal{C}$ is capacity-approaching. This alternate strategy is more efficient in terms of the total number of bits carried over the network than the original scheme of all nodes exchanging all their data.

It is important to note that the encoding technique above involves lossless measurement reconstruction. A variation from that setup comes up in applications where only an approximation of the measurements is sufficient or acceptable for post-processing. Lossy reconstruction of measurements affords an even greater reduction in communications expenditures across the sensor network and is able to achieve the same compression rates as the lossless case, up to a maximum distortion level [59, 120]. The general coset encoding approach is certainly not new in the WSN context; however, previous constructions differ from the current framework in very

notable ways. One proposed technique is to perform distributed source coding in highly-dense sensor networks with scalar quantizers and trellis-based partitions using convolutional codes [120, 121]. While this technique yields relatively simple and effective implementations, the drawback is that the achieved rates are suboptimal. Another approach is for each sensor to compress its own data blindly (without any inter-sensor communication) and send it to a super node where the all the quantized measurements are reconstructed via an adaptive algorithm that tracks the correlations among the measurements [122]. The encoding procedure is akin to the binning process but it does not use error-correcting codes. This approach provides non-trivial energy savings but a major shortfall is that the achieved rates turn out to be suboptimal as well.

The work by A. Scaglione and S. Servetto exposed the remarkable connection between routing and distributed compression in WSN [123]. More specifically, the tradeoff between bandwidth usage and decoding delay was established according to the routing strategy employed across the sensor network. Thus, their approach is a combined strategy of simple source coding, re-coding of data at intermediate nodes, and efficient routing of sensor estimates. Although the significance of the routing-compression connection cannot be overstated, one notably flawed assumption is made about the inherent complexity of vector quantizers and their inability to exploit the underlying correlations among measurements. While this assumption remained true for a long time, it appears to ignore the recent advances in capacity-approaching codes combined with the efficient quantization algorithms exposed previously. The next example described herein challenges this notion and unveils the potential application of codeword quantization to WSN problems.

4.4.3 Three-Node Wireless Sensor Network Example

Consider a relatively simple sensor network composed of just three nodes denoted x, y, and z. The main objective is for nodes y and z to attempt to estimate the measurements obtained by node x. The set of measurements X_N collected by node x is modeled as a Bernoulli sequence of 1000 independent samples with equal probability of bit occurrence. The correlation among the measurements is modeled by passing the sequence X_N through two separate BSC with cross-over probabilities p_y and p_z. A low-density linear block code of approximate rate 0.9 is created by generating a 106×1060 parity check matrix H_{12} via the polynomial degree distribution method [67]. A different low-density parity check matrix of the same rate denoted by H_Δ is created by randomly permuting the columns of H_{12}. This two matrices are nested to form a new matrix labeled H_{13}:

$$H_{13} = \begin{bmatrix} H_{12} \\ H_\Delta \end{bmatrix}$$

The set of syndrome binary vectors are produced as follows:

$$s_{12} = H_{12} X_N$$

$$s_\Delta = H_\Delta X_N$$

$$s_{13} = \begin{bmatrix} s_{12} \\ s_\Delta \end{bmatrix}$$

An approximate rather than a perfect reconstruction of the measurements X_N implies that a lesser number of bits need to be sent to nodes y and z. Since the minimum number of bits required to perfectly reconstruct X_N is given by the entropy $H(X_N)$, a lesser rate denoted by R_b would thus be acceptable [59, 124]:

$$R_b = \frac{H(X_N) - b}{N}$$

where $b > 0$. The rate R_b is associated with the average distortion D_b as follows:

$$D_b = \sum_{X_N} \Pr(X_N) \frac{d(X_N), \hat{X}_N}{N}$$

where the measurement estimates are given by the expressions:

$$\hat{x}_N = \arg\min_{\varepsilon} d(x_N, \varepsilon) \quad \text{subject to} \quad H_{12}\varepsilon = s_{12}$$

$$\hat{x}_N = \arg\min_{\varepsilon} d(x_N, \varepsilon) \quad \text{subject to} \quad H_{13}\varepsilon = s_{13}$$

The MTP algorithm is applied to the general decoding problem posed by the two expressions above. As the cross-over probabilities of the virtual BSC increase, the average of 10 X_N estimates is obtained at both nodes y and z the results are compared to the original sequences and the errors are tabulated. Figures 4.13 and 4.14 show the results compared to the results obtained with using the regular BP algorithm with side information at the decoder. It can be seen in the plots that even though the BP algorithm (with side information) maintains an error-free virtual channel for longer, once the errors appear they continue to grow very rapidly. On the other hand, the

MTP algorithm yields errors at lower cross-over probabilities. Nonetheless, the errors appear to settle around 100 bit errors until the cross-over probabilities get closer to their theoretical maximum.

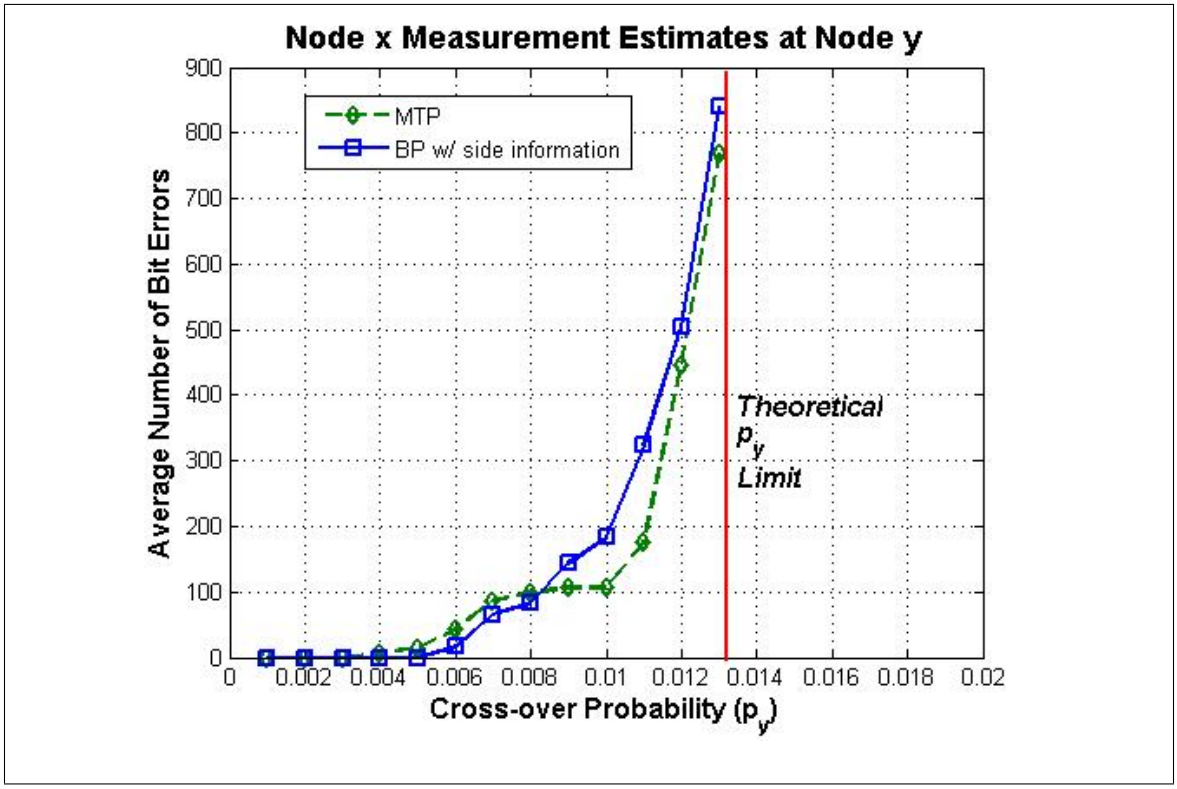

Figure 4.13: Node y Estimation Errors of Node x Measurements

The maximum values for the cross-over probabilities are $p_y \approx 0.013$ and $p_z \approx 0.031$, which are consistent with the nested code rates of 0.9 and 0.8 according to:

$$R_{12} = 1 - h_2(p_y)$$

$$R_{13} = 1 - h_2(p_z)$$

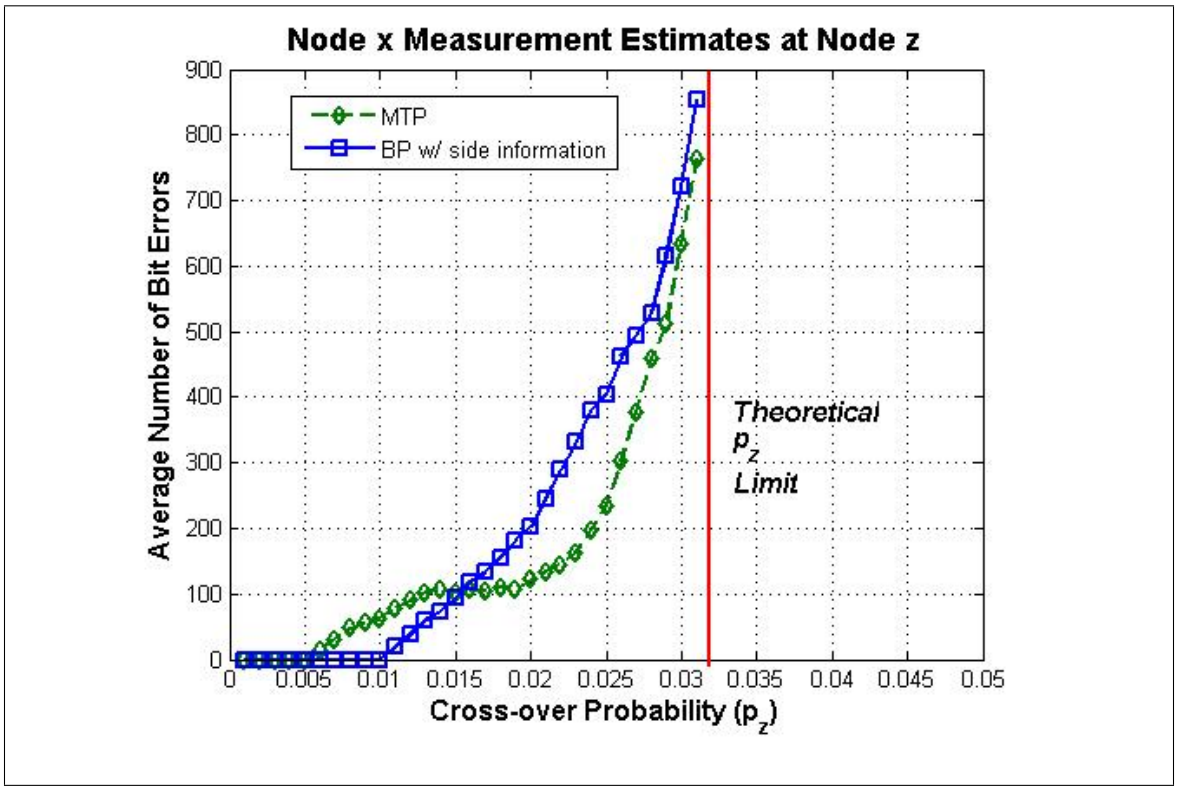

Figure 4.14: Node z Estimation Errors of Node x Measurements

The number of bits required to be sent to nodes y and z using the BP algorithm scheme is 106 and 212 respectively. One strong advantage of the MTP algorithm scheme is that it could still perform well even with a lesser number of side information bits.

5

Summary and Conclusions

The continuous expansion of wireless services and capabilities around the world has placed severe strains on the limited bandwidth resources that an ever-increasing number of users have to share. Attaining communication (Shannon) capacity in this multi-user setting is a critical challenge to both the academic research community and the wireless communications industry. The exploitation of modern coding advances such as dirty paper coding has been slowed considerably by the apparent prohibitive complexities inherent to their implementation. Concepts drawn from statistical physics have been recently applied with success asymptotic to the block length [2, 3]. These techniques were originally devised to solve difficult constraint-satisfaction problems common in computer science. Nonetheless, their computational complexity is also non-trivial. This research was mainly motivated by the development of new source coding procedures based on traditionally simpler message-passing techniques. The

results are summarized next along with a discussion about potential future research directions.

5.1 Research Objectives and Contributions

Recent techniques such as SP gave rise to the wide spread belief that the codeword quantization problem could not be solved just by BP-based methods. One of the main accomplishments of this research effort has been to refute this claim. Similar assertions have been made in [32, 34]. Nevertheless, the algorithms proposed in chapter 3 are distinct from those in the sense tha they are computationally less onerous and the results appear to hold even for short block lengths. This feature is very atractive in many applications were high-speed communications are required. To this end, this research focused on two primary objectives:

1. Develop new computationally efficient algorithms to perform codeword quantization
2. Develop and/or expand theoretical background of these new procedures

With regards to the first objective, two new iterative algorithms have been proposed in chapter 3. The first algorithm is called TP, which is essentially BP with a slight modification to the check to variable node update equation as follows:

$$x_j \to f_j = \alpha x_j + (1-\alpha) q_j \to x_j, \quad \text{for} \quad 0 < \alpha < 1$$

Even though TP was developed empirically, additional insight was gained by leveraging key ideas from information geometry. This insight is captured in the proposition below The proof details are found in chapter 3.

Proposition 1.
The fixed points of the TP algorithm are the information-theoretic projection of marginal distributions $\mathbf{t}_a$ onto the scaled $({}^{1}/_{1-\alpha})$ set of product distributions $\prod_{i\in a} t_i$.

The second iterative algorithm proposed was labeled MTP. It was inspired by TP however, its development followed a very distinct path. The MTP update equations are shown below:

$$M_{ts}(x_s) \propto \sum_{x_t} \exp(\theta_{st}(x_s, x_t) - \theta_t(x_t)) M_{\varepsilon t}(x_t) \prod_{u\in N(t)\setminus s,\varepsilon} M_{ut}(x_t)$$

The message coming from the hard constraint ε denoted by $M_{\varepsilon t}(x_t)$ is defined by:

$$M_{\varepsilon t}(x_t) \propto \sum_{x_\varepsilon} \exp((1-\alpha)\theta_{t\varepsilon}(x_t, x_\varepsilon) - \alpha\theta_\varepsilon(x_\varepsilon))(M_\varepsilon(x_\varepsilon))^\alpha \left(\prod_{u\in N(\varepsilon)} M_{u\varepsilon}(x_\varepsilon)\right)^{1-\alpha}$$

where the message $M_\varepsilon(x_\varepsilon)$ is either a fixed value or a known a priori probability.

The equation above unveils the essence of the MTP algorithm since it showcases the peculiar relationship between the hard constraint nodes and their surrounding nodes. More specifically, it shows that the message coming from the hard constraint node is a convex combination (in the log-probability space) of the hard constraint value and the messages arriving at the hard constraint damped by the

truthiness parameter α.

One possible physical interpretation of the truthiness parameter α is that it represents the level of confidence (or reliability) in the side information given. Furthermore, the higher the value of α the more emphasis (reliability) is placed on the hard constraint (side information). On the other hand, the lower the value of α the less reliable the side information is believed to be.

The following two propositions encapsulate the main findings with regards to the MTP algorithm derivation. Additional details can also be found in chapter 3.

Proposition 2.
The interior stationary points of an α-modified constrained Bethe free energy are the MTP fixed points.

Proposition 3.
The MTP candidate marginals μ lay in the relative interior of an α-modified local marginal polytope $LOCAL^{(\alpha)}(G)$ obtained from a modified Bethe approximation to the entropy $H(\mu)$.

Important rate-distortion results obtained with TP and MTP are summarized in section 5.2. Also, interesting results have been found across a number of diverse applications such as steganography, secrecy coding, and wireless sensor networks. These are also discussed in section 5.2.

5.2 Summary of Results

5.2.1 Discussion of Rate-Distortion Results

The rate-distortion experiments were conducted according to the following set up:

1. Fixed code block length of 300 bits.
2. 10 randomly-generated LDGMs per code rate using the method outlined in [67].
3. 1000 runs (100 repetitions over 10 LDGMs) per code rate.
4. Hamming distortion $D = E[d(x, Gz)]/N$ computed as ensemble average ($N = 1000$) for each code rate.
5. 300 TP and MTP iterations allowed for each repetition.
6. An optimized α determined for each code rate.

The MTP rate-distortion function (see Figure 3.3) is in the order of 1 dB above the Shannon limit. This rate-distortion performance is pretty remarkable considering that, in general, coding performance tends to get worse with shorter block length sequences as demonstrated in channel coding results [19, 56, 67, 68, 69].

This feature could be very advantageous in high data rate applications where the required throughput makes it nearly impossible to use long block codes. As

expected, the performance does improve (i.e. get closer to the lower bound) as the sequence length increases in accordance with [53]. Another impressive fact is that regular LDGMs were used to generate these results. Some of the techniques presented in chapter 2 tend to perform poorly or even fail to converge altogether when used with regular LDGM codes.

The performance of TP (not shown in Figure 3.3) appears to be slightly better than MTP as seen on [62]. Nonetheless, the MTP behavior seems to be consistently better than the TAP algorithm results reported in [32, 62]. No direct comparisons were made against the SP algorithm over LDGM codes since their source coding results appeared to be very similar to those obtained with the TAP algorithm as documented in [2, 32, 62].

The rate-distortion performance between TP and MTP using irregular LDGMs (see Figure 3.4) is comparable under the same set of conditions, again with TP having a slight edge. It is interesting to note that in this case the TAP algorithm is inadequate (i.e. non-convergent) for irregular LDGMs with degree greater than two. This could possibly be due to its close ties to the Ising spin model framework which only accounts for pairwise node relationships [32].

5.2.2 Dirty Paper Coding Example

The implementation of DPC typically involves the use of nested lattice codes [84, 85]. Nonetheless, the complexity of lattice codes grows exponentially with the constraint

length thereby limiting its practical application. Therefore, the approach for this example is to use nested LDGM/LDPC codes along with the new iterative source quantization procedures in an attempt to attain the DPC capacity for a simple two-user BC. The set of rate pairs obtained with MTP was:

$$R_1 = [0, 0.0619, 0.0725, 0.0828, 0.0894, 0.0951]$$
$$R_2 = [0.8768, 0.7658, 0.7278, 0.7060, 0.3216, 0]$$

with the corresponding maximum cross-over probabilities $p_1 = 0.3132$ and $p_2 = 0.0492$.

The achieved sum-rate (see Figure 4.3) is compared to the approximate corner points of the achievable DPC sum-rate capacity. There are a number of reasons that explain why the achieved capacity region does not saturate the optimal bound. Some of them are discussed below:

- A constant and non-optimized α parameter in the MTP algorithm was used with only 50 iterations.
- Much larger block lengths ($N \gg 1000$) would be needed in order to reach capacity.
- The code constructions used in this example do not yield the exact intended code rates.
- Code rate unpredictibility worsens when the using nested codes.

- The dual of either LDPC or LDGM codes typically have high girth (i.e. not low density).

5.2.3 Steganography Example

This example involves the succinct modification of a still image in order to insert a message. The input image (see Figure 4.5) is the infamous cameraman photograph. This image is represented by a 256×256 array of pixels with an 8-bit gray-scale per pixel stored in the Tagged Image File (TIF) format.

The MTP algorithm attempts to quantize the carrier sequence into a minimum weight vector which is then used to produce a stego sequence that meets a certain constraint but is also part of the coset of the source code. This binary vector is the cipher sequence which contains the message to be inserted in the image. The message is simply a bit sequence randomly generated from a Bernoulli source of equally probable bits. Figure 4.6 shows the cameraman image containing the stego object.

The cipher sequence is embedded into the second LSB plane by randomly chosing the pixel locations (indexes) along the plane. This is sometimes called the inverse parity check function. Again, these random pixel locations are shared between the sender and the recipient. After the embedding process is complete, the selected bit plane is placed back into the image and the modified image is sent over the channel.

Note that the differences between the original and modified images (see Figures 4.5 and 4.6) are imperceptible to the naked eye which underlines the early success of the embedding algorithm. The robustness of the embedding procedure is put to the test by subjecting the modified image to an attack mounted by an active warden. It is assumed that the warden is able to detect the possible presence of hidden data in the second LSB plane but not the specific locations of the altered bits. Hence, the attack is modeled as passing the entire second LSB plane through a BSC with bit flip probability of 0.1. Note that the post-attack image (see Figure 4.7) now shows a few scattered white spots across the photograph.

The message recovery process begins by collecting the modified bits from the pixel locations chosen by the encoder from the second LSB plane. The message is easily obtained using the regular BP algorithm to decode the low-density code generated earlier.

It is of interest to determine how close the embedding procedure gets to achieving the theoretical maximum embedding rate ($q/n \approx 0.2$ for this example). The average distortion was computed by replacing it with the empirical mean of the distortion between the host and cipher sequences over 100 iterations. The calculated mean was approximately 0.1511. To determine the rate, the message length was lowered bit by bit, from a maximum of 107 bits until no discernible differences between the original and modified images were observed. The achieved embedding rate was $95/534 \cong 0.1779$. The performance is very good considering the relatively short length of the cover signal (see Figure 4.8) and the fact that it is more difficult to achieve capacity when low distortion is desired. Also, the polynomial degree distri-

butions used to generate low-density codes do not yield codes with the exact desired rate.

5.2.4 Information Secrecy Example

The secrecy coding scheme leverages many aspects of the two-user DPC strategy described previously. The main setup steps in the experiment are laid out below:

- Generate LDPC matrix H_{ab} of approximate rate of 0.9.
- Build H_e of approximate rate of 0.8 by nesting the LDPC matrices H_{ab} and H_k of approximate rate 0.9 each (H_k is a column-permutated version of H_{ab}).
- Generate dual of H_e (i.e. assumes G_e produces a good LDGM code).
- The sequence X received from Bob and the message M from Alice are randomly generated from a Bernoulli binary source with probability of 0.5.
- The message M is treated by the encoder as side information and incorporated into the scheme as a constraint.

Once the global minimum is reached ($w \approx w_{\min}$) via the MTP algorithm then the codeword V is generated according to $V = \varepsilon + Gw_{\min}$. The message M from Alice is also used to form a syndrome vector. This syndrome vector is then used to find a particular solution ε that satisfies the constraint via Gaussian elimination in the $GF(2)$ domain. The cipher V is transmitted to Bob via a BSC with cross-over

probability p. Bob recovers the message first by regular BP decoding using H_{ab} and then calculating $M = H_k V$.

The achieved secrecy rate capacity (see Figure 4.10) is obtained by setting the cross-over probability p of the Alice-Bob channel to the minimum value according to the actual rate achieved by the parity check matrix H_{ab} and then gradually increase the cross-over probability q until convergence with MTP is lost. The procedure is repeated by increasing p from its minimum and increasing q again until convergence is lost. The maximum probability q where convergence is lost varies with each minimum probability p.

The set of collected rate pairs $(R2, C_s(R2))$ constitutes the secrecy rate capacity, which delineates the achieved secrecy region. This secrecy capacity is a function of the attained rate in the Alice-Eve channel $(R2)$ as exposed earlier. The MTP algorithm once again shows good performance given the limitations imposed by the short block lengths and the nested codes built for this example.

The set of achieved rate pairs in this example are:

$$C_s(R2) = [0, 0.2711, 0.4525, 0.6621, 0.7194]$$
$$R2 = [0.8464, 0.6581, 0.3978, 0.1260, 0]$$

with the corresponding minimum cross-over probability $p = 0.2215$.

5.2.5 Three-Node Wireless Sensor Network Example

Consider a relatively simple sensor network composed of just three nodes denoted x, y, and z. The main objective is for nodes y and z to attempt to estimate the measurements collected by node x. The set of measurements X_N collected by node x is modeled as a Bernoulli sequence of 1000 independent samples with equal probability of bit occurrence. The correlation among the measurements is modeled by passing the sequence X_N through two separate BSC with cross-over probabilities p_y and p_z. A low-density linear block code of approximate rate 0.9 is created by generating a parity check matrix H_{12} via the polynomial degree distribution method [67]. A separate low-density parity check matrix of the same rate denoted by H_Δ is created by randomly permuting the columns of H_{12}. This two matrices are nested to form a new matrix dubbed H_{13}.

An approximate reconstruction of the measurements X_N implies that a lesser number of bits need to be sent to nodes y and z. Since the minimum number of bits required to perfectly reconstruct X_N is given by the entropy $H(X_N)$, a lesser rate denoted by R_b would thus be acceptable [59, 124]:

$$R_b = \frac{H(X_N) - b}{N}$$

where $b > 0$.

As the cross-over probabilities of the virtual BSC increase, the average of 10 X_N estimates is obtained at both nodes y and z, the results are compared to the

original sequences, and the errors are tabulated. show The results obtained with MTP (see Figures 4.13 and 4.14) are compared to the results obtained with using the regular BP algorithm with side information at the decoder. Even though the BP algorithm (with side information) maintains an error-free virtual channel for longer, once the errors start to show up they continue to grow very rapidly. On the other hand, the MTP algorithm does yield errors at lower cross-over probabilities. Nonetheless, the errors appear to settle at around 100 bit errors until the cross-over probabilities get closer to their theoretical limits.

The maximum values for the cross-over probabilities are $p_y \approx 0.013$ and $p_z \approx 0.031$, which are consistent with the nested code rates of 0.9 and 0.8. The number of bits required to be sent to nodes y and z using the BP algorithm scheme is 106 and 212 respectively. One clear advantage of the MTP algorithm scheme is that it can still perform well even with a lesser number of side information bits available.

5.3 Conclusions and Future Directions

One of the most remarkable aspects of this research has been the formal establishment of a link between the new algorithms and the standard BP algorithm. This relatiionship helps to situate the new source coding procedures on firm theoretical ground given the broad acceptance and usage of BP and BP-derived techniques. As previously mentioned, it provides further evidence which disputes the widely held assumption about the inadequacy of BP-based procedures for codeword quantization.

Another crucial and rather pleasing aspect of this research was the wide applicability of these methods to many seemingly unrelated areas (e.g. steganography, WSN, etc.). The relative ease of implementation as well as the rapid convergence times observed in the examples discussed in the previous section (5.2) make the algorithms a feasible option for real-time applications.

On the other hand, there are a number of open questions that could serve as starting points for future research endevours. For instance, the truthiness parameter α required manual tuning in virtually all of the examples in order to yield good results. This makes it difficult for fast applications where there is very little time to fine tune this parameter. Hence, any effort that develops either an analytical (closed form) solution or better rules of thumb to find the optimal α would certainly be welcomed. Another area that needs further investigation is to determine if there is a more rigorous connection between TP/MTP and the generalized SP algorithms, as implied in [29].

Another factor that influenced the results presented here was the imperfect nature of the sparse code constructions. The development of methods that could yield more sparse codes and/or codes with more accurate code rates would have a significant impact. Moreover, the need to nest codes and the relative high girth of some of the dual codes further exacerbates the problem. Potential clues are suggested in [1]. A peculiar aspect of the results is that better rate-distortion performance appears to have been achieved using regular LDGM codes rather than irregular LDGM codes. This certainly needs to be confirmed under a wider variety of conditions before any definitive assertions can be made mainly because the opposite behavior is usually

expected.

Since a number of recently proposed algorithms for codeword quantization involves some sort of decimation or pruning step, it would be very constructive to look into potential quantization gains for both TP and MTP when coupled with decimation even at the expense of additional complexity. It is also very important to note that the results presented in this work assumed discrete binary sources. Possible extensions that consider Gaussian sources would indeed be very valuable.

In closing, the findings outlined in this dissertation certainly provide enough fertile ground for further research and improvements that could result in even more powerful algorithms that could bring the multi-user communication technologies of tomorrow closer to reality.

Bibliography

[1] M. Wainwright, "Sparse Graph Codes for Side Information and Binning," IEEE Signal Processing Magazine, vol. 24, September 2007.

[2] M. Wainwright and E. Maneva, "Lossy Source Coding via Message-Passing and Decimation over Generalized Codewords of LDGM Codes," in IEEE International Symposium on Information Theory, (Adelaide, Australia), September 2005.

[3] E. Martinian and J. Yedidia, "Iterative Quantization using Codes on Graphs," in Proceedings of the Allerton Conference on Communication, Control and Computing, (Monticello, IL), pp. 110–122, October 2003.

[4] S. Li, Markov Random Field Modeling in Computer Vision. Springer–Verlag, 1st ed., 1995.

[5] F. Kschischang, B. Frey, and H. Loeliger, "Factor Graphs and the Sum-Product Algorithm," IEEE Transactions on Information Theory, vol. 47, February 2001.

[6] B. Frey, Graphical Models for Machine Learning and Digital Communication. MIT Press, 1998.

[7] M. Wainwright and M. Jordan, "Graphical Models, Exponential Families, and Variational Inference," tech. rep., UC–Berkeley Dept. of Statistics, September 2003.

[8] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding Belief Propagation and its Generalizations," in Exploring Artificial Intelligence in the New Millenium, ch. 8, pp. 239–269, San Francisco, CA: Morgan Kaufmann, September 2003.

[9] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," Journal of the Royal Statistical Society, 1974. Series B.

[10] R. Tanner, "A Recursive Approach to Low Complexity Codes," IEEE Transactions on Information Theory, vol. 27, 1981.

[11] R. K. N. Wiberg, H.A. Loeliger, "Codes and Iterative Decoding on General Graphs," European Transactions on Telecommunications, vol. 6, 1995.

[12] S. Gilks and D. Spiegelhalter, Markov Chain Monte Carlo in Practice. New York, NY: Chapman and Hall, 1996.

[13] D. MacKay, Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2005.

[14] M. Wainwright and M. Jordan, "Variational Inference in Graphical Models: The View from the Marginal Polytope," in Proceedings of the Allerton Conference on Communication, Control and Computing, (Monticello, IL), October 2003.

[15] R. Gallager, "Low-Density Parity Check Codes," IRE Transactions on Information Theory, 1962.

[16] H. Bethe, "Statistical Theory of Super-Lattices," Proceedings of the Royal Society of London A, 1935.

[17] R. Kikuchi, "A Theory of Cooperative Phenomena," Physical Review, vol. 81, no. 6, 1951.

[18] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Kaufmman, 1988.

[19] T. Richardson and T. Urbanke, "The Capacity of Low-Density Parity Check Codes under Message-Passing Decoding," IEEE Transactions on Information Theory, vol. 47, pp. 599–618, February 2001.

[20] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms," IEEE Transactions on Information Theory, vol. 51, July 2005.

[21] M. Wainwright, T. Jaakkola, and A. Willsky, "A New Class of Upper Bounds on the Log-Partition Function," IEEE Transactions on Information Theory, vol. 51, July 2005.

[22] W. Wiengerinck and T. Heskes, "Fractional Belief Propagation," in Proceedings of the Neural Information Processing Systems, vol. 12, (Vancouver, BC, Canada), 2002.

[23] T. Jaakkola and D. Sontag, "New Outer Bounds on the Marginal Polytope," in Proceedings of the Neural Information Processing Systems, (Vancouver, BC, Canada), 2007.

[24] P. Regalia and J. Walsh, "Optimality and Duality of the Turbo Decoder," Proceedings of the IEEE, vol. 95, June 2007.

[25] S. Ikeda, T. Tanaka, and S. Amari, "Information Geometry of Turbo and Low-Density Parity-Check Codes," IEEE Transactions on Information Theory, vol. 50, June 2004.

[26] S. Amari and H. Nagaoka, Methods of Information Geometry. AMS and Oxford University Press, 2000.

[27] M. Mezard and R. Zecchina, "Random K-Satisfiability Problem: From an Analytic Solution to an Efficient Algorithm," Physical Review E, vol. 66, no. 5, 2002.

[28] A. Braunstein, M. Mezard, and R. Zecchina, "Survey Propagation: An Algorithm for Satisfiability," Random Structures and Algorithms, vol. 27, March 2005.

[29] E. Maneva, E. Mossel, and M. Wainwright, "A New Look at Survey Propagation and its Generalizations," Journal of the ACM, vol. 54, July 2007.

[30] R. Tu, Y. Mao, and J. Zhao, "On Generalized Survey Propagation: Normal Realization and Sum-Product Interpretation," in IEEE International Symposium on Information Theory, (Seattle, WA), July 2006.

[31] R. Tu, Y. Mao, and J. Zhao, "On the Interpretation of Survey Propagation," in IEEE 10th Canadian Workshop on Information Theory, (Canada), June 2007.

[32] T. Murayama, "Thoules-Anderson-Palmer Approach for Lossy Compression," Physical Review E, vol. 69, no. 035105(R), 2004.

[33] N. Sourlas, "Spin-Glass Models as Error-Correcting Codes," Nature, vol. 339, 1989.

[34] T. Filler and J. Fridrich, "Binary Quantization using Belief Propagation with Decimation over Factor Graphs of LDGM Codes," in Proceedings of the Allerton Conference on Communication, Control and Computing, (Monticello, IL), October 2003.

[35] L. Xiong, F. Wang, and C. Zhang, "Multilevel Belief Propagation for Fast Inference on Markov Random Fields," in IEEE International Conference on Data Mining, 2007.

[36] M. Yasdani, S. Hemati, and A. Banihashemi, "Improving Belief Propagation on Graphs with Cycles," IEEE Communication Letters, vol. 8, January 2004.

[37] M. Briers, A. Doucet, and S. Singh, "Sequential Auxiliary Particle Belief Propagation," in IEEE International Conference on Information Fusion, 2005.

[38] G. Elidan, I. McGraw, and D. Koller, "Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing," in Proceedings of the Conference on Uncertainty in Artificial Intelligence, (Cambridge, MA), July 2006.

[39] A. Vila Casado, M. Griot, and R. Wesel, "Informed Dynamic Scheduling for Belief Propagation Decoding of LDPC Codes," in Proceedings of the IEEE International Conference on Communications, (Glasgow, Scotland), June 2007.

[40] J. Kim, M. Nam, and H. Song, "Variable-to-Check Residual Belief Propagation for LDPC Codes," IEEE Electronics Letters, vol. 45, January 2009.

[41] J. Zhang and M. Fossorier, "Shuffled Belief Propagation Decoding," IEEE Transactions on Communications, vol. 53, 2005.

[42] M. Rovini, F. Rossi, P. Ciao, N. Linsalata, and L. Fanucci, "Layered Decoding of Non-Layered LDPC Codes," in Proceedings of the EUROMICRO Conference on Digital System Design, (Cavtat, Croatia), August 2006.

[43] J. Zhang, Y. Wang, M. Fossorier, and J. Yedidia, "Replica Suffled Iterative Decoding," in Proceedings of the IEEE International Symposium on Information Theory, (Adelaide, Australia), September 2005.

[44] P. Radosavljevic, A. de Baynast, and J. Cavallaro, "Optimized Message Passing Schedules for LDPC Decoding," in Proceedings of the Asilomar Conference on Signals, Systems and Computers, (Pacific Grove, CA), November 2005.

[45] Z. He, S. Roy, and P. Fortier, "Lowering Error Floor of LDPC Codes using a Joint Row-Column Decoding Algorithm," in Proceedings of the IEEE International Conference on Communications, (Glasgow, Scotland), June 2007.

[46] O. Golov and O. Amrani, "Edge-Based Scheduled BP in LDPC Codes," in Proceedings of the IEEE International Symposium on Information Theory, (Nice, France), June 2007.

[47] S. Gounai, T. Ohtsuki, and T. Kaneko, "Modified Belief Propagation Decoding Algorithm for Low-Density Parity Check Code baed on Oscillation," in Proceedings of the IEEE Vehicular Technology Conference, 2006.

[48] T. Minka and Y. Qi, "Tree-structured Approximations by Expectation Propagation," Advances in Neural Information Processing Systems, vol. 16, 2004.

[49] S. Ciliberti, M. Mezard, and R. Zecchina, "Lossy Data Compression with Random Gates," Physical Review Letters, vol. 95, no. 038701, 2005.

[50] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree-based Re-parameterization for Analysis of Sum-Product and Related Algorithms," IEEE Transactions on Information Theory, vol. 49, no. 5, 2003.

[51] T. Roosta, M. Wainwright, and S. Sastry, "Convergence Analysis of Re-weighted Sum-Product Algorithms," in Proceedings of the IEEE ICASSP, 2007.

[52] C. Moallemi and B. Van Roy, "Consensus Propagation," IEEE Transactions on Information Theory, vol. 52, no. 11, 2006.

[53] C. Shannon, "A Mathematical Theory of Communication," tech. rep., Bell Labs Bell System Technical Journal, July-October 1948.

[54] T. Cover, Elements of Information Theory. Wiley, 1991.

[55] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," in Proceedings of the IEEE International Conference on Communications, (Geneva, Switzerland), May 1993.

[56] D. MacKay, "Good Error-Correcting Codes based on Very Sparse Matrices," IEEE Transactions on Information Theory, vol. 45, no. 2, 1999.

[57] S. Cook, "The Complexity of Theorem Proving Procedures," in Proceedings of the Annual ACM Symposium on Theory of Computing, 1971.

[58] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the Inherent Intractability of Certain Coding Problems," IEEE Transactions on Information Theory, vol. 24, no. 3, 1978.

[59] A. Wyner and J. Ziv, "The rate-distortion function for source encoding with side information at the decoder," IEEE Transactions on Information Theory, vol. 22, no. 1, 1976.

[60] S. Pradhan, J. Chou, and K. Ramchandran, "Duality between Source Coding and Channel Coding and its Extension to the Side Information Case," IEEE Transactions on Information Theory, vol. 49, no. 5, 2003.

[61] R. Barron, B. Chen, and G. Wornell, "The Duality between Information Embedding and Source Coding with Side Information and Some Applications," IEEE Transactions on Information Theory, vol. 49, no. 5, 2003.

[62] P. Regalia, "A Modified Belief Propagation Algorithm for Codeword Quantization," IEEE Transactions on Communications, vol. 57, no. 12, 2009.

[63] A. Dimakis, K. Ramchandran, and W. Wainwright, "Lower Bounds on the Rate-Distortion Function of LDGM Codes," in Proceedings of the IEEE Information Theory Workshop, (Lake Tahoe, CA), September 2007.

[64] P. Regalia, "Gradient Decoding Revisited," in Proceedings of the Asilomar Conference on Circuits, Systems, and Computers, (Pacific Grove, CA), November 2007.

[65] P. Regalia, "belief propagation fixed points," 2009. correspondence and notes.

[66] B. Lathi, Modern Digital and Analog Communication Systems. Oxford University Press, 1998.

[67] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity Check Codes," IEEE Transactions on Information Theory, vol. 47, February 2001.

[68] S.-Y. Chung, G. Forney, T. Richardson, and R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit," IEEE Communication Letters, vol. 5, 2001.

[69] D. MacKay, S. Wilson, and M. Davey, "Comparison of Constructions of Irregular Gallager Codes," IEEE Transactions on Communication, vol. 47, 2001.

[70] G. Caire and S. Shamai, "On Achievable Rates in a Multi-Antenna Broadcast Downlink," in Proceedings of the Allerton Conference on Communications, Control, and Computing, (Monticello, IL), October 2000.

[71] R. Anderson and F. Peticolas, "On the Limits of Steganography," IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, 1998.

[72] M. Costa, "Writing on Dirty Paper," IEEE Transactions on Information Theory, vol. 29, 1983.

[73] G. Foschini, "Layered Space-Time Architecture for Wireless Communication in Fading Environments when using Multi-Element Antennas," tech. rep., Bell Labs Bell System Technical Journal, 1996.

[74] E. Telatar, "Capacity of Multi-Antenna Gaussian Channels," European Transactions in Telecommunications, vol. 10, November 1999.

[75] J. Winters, "On the Capacity of Radio Communication Systems with Diversity in a Rayleigh Fading Environment," IEEE Journal on Selected Areas in Communications, vol. 5, 1987.

[76] A. Goldsmith, S. Jafar, N. Jindhal, and S. Vishwanath, "Capacity Limits of MIMO Gaussian Channels," IEEE Journal on Selected Areas in Communications, vol. 21, June 2003.

[77] K. Marton, "A Coding Theorem for the Discrete Memoryless Broadcast Channel," IEEE Transactions on Information Theory, vol. 25, May 1979.

[78] H. Sato, "An Outer Bound on the Capacity Region of the Broadcast Channel," IEEE Transactions on Information Theory, vol. 24, 1978.

[79] W. Yu and J. Cioffi, "Trellis Precoding for the Broadcast Channel," in Proceedings of the Global Communications Conference, 2001.

[80] S. Vishwanath, N. Jindhal, and A. Goldsmith, "On the Capacity of Multiple Input Multiple Output Broadcast Channels," in Proceedings of the International Conference on Communications, 2002.

[81] S. Vishwanath and D. Tse, "Sum Capacity of the Multiple Antenna Gaussian Broadcast Channel," in Proceedings of the International Symposium on Information Theory, 2002.

[82] W. Yu and J. Cioffi, "Sum Capacity of a Gaussian Vector Broadcast Channel," in Proceedings of the Global Communications Conference, 2001.

[83] S. Vishwanath, N. Jindhal, and A. Goldsmith, "Duality, Achievable Rates, and Sum-Rate Capacity of Gaussian MIMO Broadcast Channels," IEEE Transactions on Information Theory, vol. 49, October 2003.

[84] U. Erez, S. Shamai, and R. Zamir, "Capacity and Lattice-Strategies for Cancelling Known Interference," in Proceedings of the International Symposium on Information Theory and its Applications, (Honolulu, HI), November 2000.

[85] U. Erez and S. ten Brink, "Approaching the Dirty Paper Limit for Cancelling Known Interference," in Proceedings of the Allerton Conference on Communications, Control, and Computing, (Monticello, IL), October 2003.

[86] M. Marcellin and T. Fischer, "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources," IEEE Transactions in Communications, vol. 38, January 1990.

[87] E. Martinian and M. Wainwright, "Low-Density Graph Codes that are Optimal for Binning and Coding with Side Information," IEEE Transactions on Information Theory, vol. 55, March 2009.

[88] P. Regalia, "dirty paper coding: two-user case," 2010. correspondence and notes.

[89] R. Anderson and F. Peticolas, "Information Hiding–A Survey," Proceedings of the IEEE, vol. 87, July 1999.

[90] N. Johnson, "Information hiding: steganography and digital watermarking." http://www.jjtc.com/Steganography/, May 2011.

[91] G. Simmons, "The Prisoners' Problem and the Subliminal Channel," in Advances in Cryptology: Proceedings of CRYPTO '83, 1984.

[92] A. Kerckhoffs, "Military Cryptology," French Journal of Military Science, 1883.

[93] S. Gelfand and M. Pinsker, "Coding for Channel with Random Parameters," Problems in Control Theory, vol. 9, no. 1, 1980.

[94] C. Heegard and A. El-Gamal, "On the Capacity of Computer Memory with Defects," IEEE Transactions on Information Theory, vol. 29, no. 5, 1983.

[95] P. Moulin and J. O'Sullivan, "Information-Theoretic Analysis of Information Hiding," IEEE Transactions on Information Theory, vol. 49, March 2003.

[96] L. Marvel, C. Boncelet, and C. Retter, "Spread Spectrum Steganography," IEEE Transactions on Image Processing, vol. 8, no. 8, 1999.

[97] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on Wet Paper," IEEE Transactions on Signal Processing, vol. 53, October 2005.

[98] A. Kuznetsov and B. Tsybakov, "Coding in a Memory with Defective Cells," Problems of Information Transmission, vol. 10, 1974.

[99] J. Fridrich, M. Goljan, and D. Soukal, "Wet Paper Codes with Improved Coding Efficiency," IEEE Transactions on Information Forensics and Security, vol. 1, March 2006.

[100] J. Fridrich and D. Soukal, "Matrix Embedding for Large Payloads," IEEE Transactions on Information Forensics and Security, vol. 1, September 2006.

[101] C. Shannon, "Communication Theory of Secrecy Systems," tech. rep., Bell Labs Bell System Technical Journal, 1949.

[102] M. Bloch and J. Barros, Physical Layer Security: From Information Theory to Security Engineering. Cambridge University Press, 1st ed., 2011.

[103] Y. Liang, H. Poor, and S.Shamai, Information Theoretic Security: Foundations and Trends in Communications and Information Theory. Now Publishers, 1st ed., 2009.

[104] A. Wyner, "The Wire-Tap Channel," tech. rep., Bell Labs Bell System Technical Journal, 1975.

[105] I. Csiszar and J. Korner, "Broadcast Channels with Confidential Messages," IEEE Transactions on Information Theory, vol. 24, 1978.

[106] M. van Dijk, "On a special class of Broadcast Channels with Confidential Messages," IEEE Transactions on Information Theory, vol. 43, 1997.

[107] H. Yamamoto, "On Secret Sharing Communication Systems with Two or Three Channels," IEEE Transactions on Information Theory, vol. 32, 1986.

[108] J. Katz and Y. Lindell, Introduction to Modern Cryptography. Chapman and Hall/CRC Press, 1st ed., 2007.

[109] U. Maurer, "Secret Key Agreement by Public Discussion from Common Information," IEEE Transactions on Information Theory, vol. 39, 1993.

[110] J. Muramatsu, "Secret Key Agreement from Correlated Source Outputs using LDPC Matrices," in Proceedings of the International Symposium on Information Theory, (Chicago, IL), June 2004.

[111] M. Anderson, V. Rathi, R. Thobaben, J. Kliewer, and M. Skoglund, "Equivocation of Eve using Two-Edge Type LDPC Codes for the Binary Erasure Wiretap Channel," in Proceedings of the Asilomar Conference on Signals, Systems, and Computing, (Pacific Grove, CA), November 2010.

[112] P. Regalia, "physical layer security," 2011. correspondence and notes.

[113] D. Estrin, R. Govindan, J. Heidmann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, 1999.

[114] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," in Proceedings of the International Conference on Distributed Computing Systems, 2001.

[115] F. Lewis, "Wireless Sensor Networks," in Smart Environments: Technologies, Protocols, and Applications (D. Cook and S. Das, eds.), Wiley–Interscience, 1st ed., 2005.

[116] E. F. Nakamura, A. Loureiro, and A. C. Frery, "Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications," ACM Computing Surveys, vol. 39, September 2007.

[117] K. Romer and F. Mattern, "The Design Space of Wireless Sensor Networks," IEEE Wireless Communications, vol. 11, December 2004.

[118] N. Zhang and W. Zhao, "Privacy Protection against Malicious Adversaries in Distributed Information Sharing Systems," IEEE Transactions on Knowledge and Data Engineering, vol. 20, August 2008.

[119] D. Slepian and J. Wolf, "Noiseless Coding of Correlated Information Sources," IEEE Transactions on Information Theory, vol. 19, 1973.

[120] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed Compression in a Dense Micro-Sensor Network," IEEE Signal Processing Magazine, vol. 19, 2002.

[121] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," IEEE Signal Processing Magazine, vol. 28, 1982.

[122] J. Chou, D. Petrovic, and K. Ramchandran, "A Distributed and Adaptive Signal Processing Approach to Reducing Energy Consumption in Sensor Networks," in Proceedings of the IEEE International Conference on Computer Communications, 2003.

[123] A. Scaglione and S. Servetto, "On the Interdependence of Routing and Data Compression in Multi-hop Sensor Networks," in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, 2002.

[124] P. Regalia, "distributed information sharing," 2010. correspondence and notes.