Master Thesis

Tarkus Belief Propagation

On Message Passing Algorithms and Computational Commutative Algebra

conducted at the
Signal Processing and Speech Communications Laboratory
Graz University of Technology, Austria

by
Carlos Eduardo Cancino Chacón

Supervisor:
Assoc.Prof. Dipl.-Ing. Dr. Franz Pernkopf

Assessors/Examiners:
Assoc.Prof. Dipl.-Ing. Dr. Franz Pernkopf
Dr. Pejman Mowlaee, B.Sc., M.Sc., Ph.D.
Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Christian Magele

Graz, July 17, 2014

This work was funded by the Mexican National Council for Science and Technology (CONACyT) under scholarship 217746.

ABSTRACT

Probabilistic graphical models are used in several areas, including signal processing, artificial intelligence, machine learning and physics. Belief propagation is among the most popular message passing algorithms for dealing with probabilistic inference. This algorithm uses the structure of the graph representing the conditional dependencies of the random variables in the probabilistic model to efficiently calculate marginal probability distributions. On the other hand, computational commutative algebra studies the algorithms for characterizing solutions of systems of multivariate polynomial equations. One of the main concepts in this theory is that of Gröbner bases. These bases allow the description of the algebraic and geometric structures generated by a system of polynomial equations, and can be used to compute the solutions for such equations. In this thesis, it is shown that belief propagation can be alternatively understood as computing marginal probabilities by solving a system of polynomial equations. By doing so, the relationship between belief propagation and computational commutative algebra can be explored. The notion of convergence in belief propagation is analyzed in algebraic terms, which leads to new conditions for convergence that use the properties of Gröbner bases. These concepts are used to derive new proofs for the well-known convergence results of the belief propagation algorithm for graphical models with chain, tree and single loop structures. Furthermore, using the framework of computational commutative algebra, an alternative formulation of belief propagation is proposed. We denominate this new approach Tarkus belief propagation. This method is experimentally compared with standard belief propagation and exact inference using a 2×2 spin glass. The experimental results show that Tarkus belief propagation is more computationally expensive and less reliable than standard belief propagation. Nevertheless, this new approach suggest an interesting insight into the basic principles of probabilistic inference, by showing some possible applications of methods from computational commutative algebra into the probabilistic graphical models.

KURZFASSUNG

Probabilistische graphische Modelle werden in unterschiedlichen Bereichen, wie Signalverarbeitung, Künstliche Intelligenz, Machine Learning oder Physik verwendet. Dabei ist Belief Propagation eines der am häufigsten verwendeten Message Passing-Algorithmen für probabilistische Inferenz. Dieser Algorithmus verwendet die Struktur des Graphen, welcher die konditionellen Abhängigkeiten der Zufallsvariablen repräsentiert, um die marginalen Wahrscheinlichkeitsverteilungen effizient zu modellieren. Kommutative Algebra behandelt Algorithmen zum Lösen von Systemen multivariater polynomischer Gleichungen. Ein wichtiges Konzept dabei sind die Gröbnerbasen. Diese Basen beschreiben die algebraischen und geometrischen Strukturen, welche von einem System polynomischer Gleichungen generiert werden und kann auch zur Lösung solcher Gleichungen verwendet werden. In dieser Arbeit wird gezeigt, dass Belief Propagation auch als das Berechnen von marginalen Wahrscheinlichkeiten durch Lösen eines polynomiellen Gleichungssystems verstanden werden kann. Dabei wird auch der Zusammenhang zwischen Belief Propagation und kommutativer Algebra untersucht. Die Idee der Konvergenz in Belief Propagation wird algebraisch analysiert, was durch Gröbner Basen zu neuen Bedingungen für eine solche Konvergenz führt. Diese Konzepte werden dafür verwendet, neue Beweise für bekannte Konvergenz-Resultate des Belief Propagation Algorithmus für Graphenmodelle mit Ketten-, Baum- und Single-Loop Strukturen zu finden. Darüber hinaus wird durch die Verwendung von computerisierter kommutativer Algebra eine alternative Formulierung von Belief Propagation vorgeschlagen. Wir nennen diesen neuen Ansatz Tarkus Belief Propagation. Diese Methode wird experimentell an einem 2×2 Spin Glass sowohl mit der herkömmlichen Belief Propagation, wie auch mit exakter Inferenz verglichen. Die Resultate zeigen, dass Tarkus Belief Propagation einerseits mit einem höheren Rechenaufwand einher geht, andererseits ermöglicht diese Methode interessante Einblicke in die Prinzipien probabilistischer Inferenz.

RESUMEN

Los modelos probabilísticos gráficos son utilizados en diversas áreas, incluyendo procesamiento de señales, inteligencia artificial, aprendizaje automático y física. Propagación de Creencias es uno de los algoritmos de paso de mensajes mas usados en el área de inferencia probabilística. Este algoritmo usa grafos, cuya estructura representa dependencias condicionales de variables aleatorias en los modelos probabilísiticos, para calcular eficientemente distribuciones marginales de probabilidad. Por otra parte, el álgebra conmutativa computacional estudia los algoritmos para caracterizar las soluciones de sistemas de ecuaciones polinomiales en múltiples variables. Uno de los conceptos más importantes en esta teoría son las bases de Gröbner. Estas bases permiten describir los objetos algebraicos y geométricos descritos por un sistema de ecuaciones polinomiales. En esta tesis se muestra que el algoritmo de Propagación de Creencias puede ser entendido como un método para calcular distribuciones marginales de probabilidad, resolviendo un sistema de ecuaciones polinomiales. Esto permite explorar la relación entre Propagación de Creencias y el álgebra conmutativa computacional. La noción de convergencia en este algoritmo es analizada en términos algebraicos, lo cual lleva a nuevas condiciones de convergencia, las cuales usan las propiedades de las bases de Gröbner. Estas condiciones son usadas para derivar demostraciones alternativas de los casos conocidos de convergencia del algoritmo de Propagación de Creencias en grafos acíclicos y grafos monociclo. Usando el marco teórico de álgebra conmutativa computacional, se propone una formulación alternativa al algoritmo de Propagación de Creencias, la cual es denominada Propagación de Creencias Tarkus. Este método es comparado experimentalmente con el algoritmo tradicional de Propagación de Creencias e inferencia exacta usando un vidrio de spin de 2×2. Los resultados muestran que el método Tarkus es computacionalmente más costoso e impreciso que el algoritmo original. Sin embargo, esta nueva formulación ofrece un punto de vista interesante sobre los principios básicos de inferencia probabilística.

Acknowledgments

I would like to thank Prof. Franz Pernkopf for introducing me to the area of machine learning and probabilistic graphical models, and giving me the opportunity to pursue this thesis under his supervision. I also thank him for his constant support and patience through many discussions we had over the course of this thesis. I also would like to thank Sebastian Tschiatschek and Michael Wohlmayr for their useful insight and interesting commentaries in the preparation of this work.

Living in Graz could have been not such a great experience without my dear friends Linda Lüchtrath and Mario Watanabe. I will always thank them for making me feel at home in this land so far away from Mexico. I thank Marisol Carrillo for her friendship, and for helping me (re)discover latin-american music.

I can't express enough gratitude to Rafael Cruz and José Antonio Maqueda, for their unconditional friendship. Syninoforcimno! Also, I would like to thank Myriam Albor, because her time and space always win by a nose to my space-time.

I thank my parents for their unconditional love, encouragement and for always believing in me. I'm grateful for having the most awesome siblings in the world! I also thank Casilda Rivera and Walter Sottolarz, for all their help and support. They have been like a family to me all these years in Austria.

I thank the Mexican National Council for Science and Technology (CONACyT) for funding this work under scholarship 217746.

This thesis is respectfully dedicated to the memory of Mr. Elias Howe, who, in 1846, invented the sewing machine.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

________________________	______________________________
date	(signature)

Contents

List of Algorithms

1 Introduction

Inference in PGMs is used in a wide range of applications, including signal processing [1], artificial intelligence [2, 3], and statistical physics [4]. Message Passing Algorithms (MPAs) over PGMs, are among the most popular methods when dealing with inference [5], due to their simplicity and computational efficiency [6]. Introduced by Judea Pearl in 1982 [7], the Belief Propagation (BP) algorithm is an MPA frequently applied in the field of artificial intelligence [2,3], error correcting codes [8], speech recognition [9] and computer vision [10], among others. This algorithm performs probabilistic inference iteratively, by exploiting the structure of the graph in a PGM, to efficiently compute marginal probability distributions [11]. It was shown by Pearl that BP converges to a solution, and this solution is equal to the true marginal probability distribution for problems that can be represented by graphical models without cycles [12].

The Loopy Belief Propagation (LBP) algorithm is an extension of BP to graphs with cycles (also known as loops, and hence the name) [5, 6]. It has been empirically shown that LBP provides good approximate results, although not necessarily the true marginal probabilities [5, 13]. However, in general, LBP is not guaranteed to converge to a solution [2, 3]. Yedida et al. showed a connection between the convergence of the BP and LBP algorithms and the fixed points of the Bethe free energy [14], a concept first originated in thermodynamics that represents the available energy of a physical system for performing mechanical work [15]. The investigation of the fixed points of the Bethe free energy has led to the derivation of convergence criteria for the LBP algorithm such as the ones proposed by Ihler et al. [16] and Mooij et al. [17]. Weiss showed that the LBP algorithm converges for graphs with a single loop [3].

On the other hand, commutative algebra studies systems of polynomial equations, trying to answer the questions whether such systems have finitely or infinitely solutions, and how to describe them [18]. Although some of the theoretical foundations of commutative algebra date from the end of the 19th century, it is only in recent years that it has regained its prominence, due to the increase of computational power and the development of new algorithms [19]. One of the most important concepts in commutative algebra is that of Gröbner Bases (GBs), introduced by Bruno Buchberger in 1965 [19]. Similar to bases of a vector space, a GB is a finite set of polynomials that allows us to represent all members of a (possibly infinite) set of polynomials called *polynomial ring* by polynomial combinations of the elements of such a GB. These ideas found their way into many applications ranging from pure mathematics [20] to signal processing [21] and robotics [18].

In this thesis we explore the relationship between BP and computational commutative algebra by showing that the equations describing the message passing in the BP algorithm can be understood as a system of polynomial equations. This result allows us to use the concept of GBs to express convergence criteria for the BP algorithm and to formulate an alternative method for computing the marginal probabilities.

The methods proposed in this work are only suited for toy examples. However they give an interesting insight into the basic principles of probabilistic inference, since they show that the ability of PGMs to represent joint probability distributions as products of conditional probability distributions can be exploited using purely algebraic methods to answer probabilistic inference queries.

1.1 Summary of Contributions

1. **Convergence of the BP algorithm**. Using the framework of computational commutative algebra, new conditions for convergence of the BP algorithm can be derived. These conditions make use of the properties of GBs to characterize the solutions of systems of polynomial equations.
2. **Convergence for Graphs without loops and for Graphs with a single loop**. Using the aforementioned convergence conditions, we show that the (L)BP algorithm converges for graphs without loops and for graphs with a single loop. While these are well-known results [3, 12], the proofs provided in this thesis present a new and interesting approach, by using the framework of computational commutative algebra.
3. **Tarkus Belief Propagation**. An alternative formulation of the BP algorithm using GBs is proposed. This algorithm exploits the fact that BP can be understood as a system of polynomial equations, and uses the properties of GBs to find an equivalent system of equations, that might be easier to solve. Due to the eclectic nature of this new algorithm, we call it Tarkus Belief Propagation (TBP), as an homage to the 1971 eponymous album by the british progressive rock band Emerson, Lake & Palmer.

1.2 Organization

The organization of this thesis is summarized as follows:

Chapter 2: In this chapter, the concepts of probabilistic graphical models, as well as probabilistic inference using the Belief Propagation algorithm are briefly reviewed. We emphasize how equations of message passing can be seen as a system of multivariate polynomial equations.

Chapter 3: In this chapter, an overview of the concepts of algebraic geometry and computational commutative algebra, which were applied in this work, is presented. The concept of *affine varieties*, i.e. a geometric object that represents the set of roots of a system of polynomial equations, is briefly reviewed. It is shown how we can characterize such varieties using Hilbert's Nullstellensatz and GBs.

Chapter 4: In this chapter, an algebraic formulation of the BP algorithm is presented. We use the message passing equations (MPEs) described by the BP algorithm to define a system of polynomials, which has an associated affine variety. By computing the GB of such a variety, conditions for convergence can be found. These methods lead to the introduction of the TBP algorithm. This algorithm computes marginal probability distributions by finding solutions of the MPEs.

Chapter 5: In this chapter, the methods proposed in Chapter 4 are empirically compared to LBP and exact inference using a 2×2 spin glass.

Chapter 6: In the last chapter, the conclusions and future work of this thesis are provided. Some conjectures about the efficiency and stability of methods of computational commutative algebra, and its possible applications in PGMs are presented.

2

Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs) have become the method of choice for dealing with uncertainty and distributions in several research areas including computer vision [10], speech processing [9], signal processing [1, 22], machine learning [13] and in the area of artificial intelligence [23]. By merging graphical models and probabilistic inference, such a framework allows to transfer concepts and ideas among different application areas [5]. One reason for its popularity is that qualitative patterns of commonsense reasoning "*are naturally embedded within the syntax of probability calculus*" [12, pp. 19].

PGMs describe the way a joint probability distribution over a set of N random variables (RVs) can be factored into a product of conditional probability distributions, defined over smaller subsets of RVs [24]. Examples of the well-known statistical models that can be represented as PGMs are hidden Markov models, Kalman filters and Boltzmann machines [5,11]. The structure of the graphical model represents the conditional independence between RVs and alleviates the computational burden for model learning and inference [5, 6].

Among the most popular representations of PGMs are Markov Networks (MN) or undirected graphical models, Bayesian Networks (BNs), or directed graphical models, and Factor Graphs (FGs) [5]. Each representation captures different aspects of probabilistic models, and therefore, has its specific advantages and disadvantages [11]. For the sake of simplicity, in this thesis we focus only on MNs. Nevertheless the discussion and methods presented in this chapter can also be extended to BNs and FGs.

If not stated otherwise, the definitions and notation for PGMs used in this thesis are taken from the tutorial by Pernkopf, Perharz and Tschiatschek [5]. For a more extensive treatment of this subject, we refer the reader to the standard text by Koller and Friedman [11] and the above mentioned tutorial. The rest of this chapter is organized as follows: In Sections 2.1 and 2.2, respectively, a short review of probability theory and graph theory is provided. In Section 2.3 the representation of PGMs using MN is presented. In Section 2.4, the concept of probabilistic inference is reviewed. We conclude this chapter in Section 2.5, where the BP algorithm for the case of MNs is described in detail.

2.1 Probability theory overview

The probability distribution of an RV[1] can be characterized using its cumulative distribution function, which is related to the probability density function or the probability mass function, respectively. These functions are defined as follows:

Definition 1. ***(Cumulative Distribution Function, Probability Density Function, Probability Mass Function).*** *The* cumulative distribution function *(CDF) of an RV X, denoted as $F_X(x)$, is defined as the probability of X taking a value less than or equal to x , i.e.*

$$F_X(x) = P(X \leq x). \tag{2.1}$$

For a set of N RVs $\mathbf{X} = \{X_1, \dots, X_N\}$, the joint CDF *is defined as*

$$F_{\mathbf{X}}(\mathbf{x}) = P(X_1 \leq x_1 \cap \dots \cap X_N \leq x_N). \tag{2.2}$$

If $\mathbf{X}$ is a set of continuous *RVs, the* probability density function (pdf) *is defined as*

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{\partial^n F_{\mathbf{X}}(\mathbf{x})}{\partial x_1 \dots \partial x_n}, \tag{2.3}$$

where $\mathbf{x} = \{x_1, \dots, x_N\}$ is an ordered set of N values from $\mathbb{R}$. In the case of $\mathbf{X}$ being a set of discrete *RVs, the* probability mass function (pmf) *is given as*

$$p_{\mathbf{X}}(\mathbf{x}) = P(X_1 = x_1 \cap \dots \cap X_N = x_N), \ \mathbf{x} = \{x_1, \dots, x_N\} \in \mathbf{val}(\mathbf{X}), \tag{2.4}$$

where $\mathbf{val}(\mathbf{X})$ denotes the set of values which can be assumed by a set of random variables $\mathbf{X}$.

In this thesis, it is assumed that $p_{\mathbf{X}}(\mathbf{x})$ represents the underlying probability distribution P and with slight abuse of notation, $p_{\mathbf{X}}(\mathbf{x})$ is itself referred as probability distribution. Whenever it is clear, the shorthand notation $p(\mathbf{x}) = p_{\mathbf{X}}(\mathbf{x})$ is used. We focus only on the case of discrete RVs, and we will restrict $p(\mathbf{x})$ to be discrete. The number of possible states of variable X_i is denoted as $\mathbf{sp}(X_i) = |\mathbf{val}(X_i)|$. For the case of a set of discrete RVs $\mathbf{X}$ the number of possible states is given as

$$\mathbf{sp}(\mathbf{X}) = \prod_i \mathbf{sp}(X_i). \tag{2.5}$$

Definition 2. ***(Marginal Distribution, Conditional Distribution).*** *Let $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ be sets of RVs, where $\mathbf{Y} \subseteq \mathbf{X}$ and $\mathbf{Z} = \mathbf{X} \setminus \mathbf{Y}$, i.e. $\mathbf{X} = \mathbf{Y} \cup \mathbf{Z}$. The joint distribution over $\mathbf{X}$ is then $p(\mathbf{X}) = p(\mathbf{Y}, \mathbf{Z})$. The* marginal distribution *$p(\mathbf{Y})$ over $\mathbf{Y}$ is given as*

$$p(\mathbf{Y}) = \sum_{\mathbf{z} \in \mathbf{val}(\mathbf{Z})} P(\mathbf{Y}, \mathbf{Z} = \mathbf{z}) = \sum_{\mathbf{z} \in \mathbf{val}(\mathbf{Z})} p(\mathbf{Y}, \mathbf{z}). \tag{2.6}$$

The conditional distribution *$p(\mathbf{Y}|\mathbf{Z})$ over $\mathbf{Y}$ conditioned on $\mathbf{Z}$ is*

$$p(\mathbf{Y}|\mathbf{Z}) = \frac{p(\mathbf{Y}, \mathbf{Z})}{p(\mathbf{Z})}. \tag{2.7}$$

Using these definitions, we can use the *Bayes' rule* to manipulate conditional probability distri-

[1] The formal definitions of probability distributions and RVs can be found in Appendix B.1.

butions. This rule states that

$$p(\mathbf{Z}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{Z})p(\mathbf{Z})}{p(\mathbf{Y})}. \tag{2.8}$$

Definition 3. ***(Conditional Statistical Independence)****. Assuming* $\mathbf{X}, \mathbf{Y}$ *and* $\mathbf{Z}$ *are mutually disjoint sets of RVs.* $\mathbf{X}$ *and* $\mathbf{Z}$ *are* conditionally statistical independent *given* $\mathbf{Z}$*, denoted as* $\mathbf{X} \perp \mathbf{Y} | \mathbf{Z}$ *iff* $p(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) = p(\mathbf{X}|\mathbf{Z})p(\mathbf{Y}|\mathbf{Z})$*.*

In case that $\mathbf{Z} = \emptyset$*,* $\mathbf{X}$ *and* $\mathbf{Y}$ *are called* statistically independent*.*

The *chain rule of probability* uses conditional distributions to factorize an arbitrary distribution as

$$p(\mathbf{X}) = p(X_1) \prod_{i=2}^{n} p(X_i|X_{i-1}, \ldots, X_1). \tag{2.9}$$

This rule holds for any permutation of the indexes of the RVs $\mathbf{X}$.

2.2 Graph theory overview

Graph theory refers to the study of mathematical structures, which are used to model pairwise relations between objects [25]. The definitions of graphs, the fundamental structures of this theory, are presented as follows:

Definition 4. ***(Graph)****. A* graph $\mathcal{G} = (\mathbf{X}, \mathbf{E})$ *is a tuple consisting in a set of* vertices $\mathbf{X}$ *(also called* nodes*), and a set of* edges $\mathbf{E}$*.*

A graph is said to be *directed* if all edges $e \in \mathbf{E}$ are directed. Conversely, if all edges are undirected, the graph is said to be *undirected.* If the set of edges of a graph contains both directed and undirected edges, the graph is called *mixed.* For this thesis, only undirected graphical models are considered, therefore, whenever we are speaking of a graph, we are in fact considering an undirected graph. We introduce the concepts of neighborhood, cliques and paths, which define the relationships between vertices in a graph.

Definition 5. ***(Neighbor, Degree of a Vertex)****. Let* $\mathcal{G}$ *be a graph and* $X_i, X_j \in \mathbf{X}, i \neq j$*. If* $(X_i - X_j) \in \mathbf{E}$*, then* X_i *is a* neighbor *of* X_j*. The set of all neighbors of* X_i *is*

$$\mathbf{Nb}_{\mathcal{G}}(X_i) = \{X_j \mid (X_i - X_j) \in \mathbf{E}, X_j \in \mathbf{X}\}. \tag{2.10}$$

The degree of a vertex X_i*, denoted by* $\deg(X_i)$ *is the number of edges incident to the vertex [25]. Edges with themselves are counted twice.*

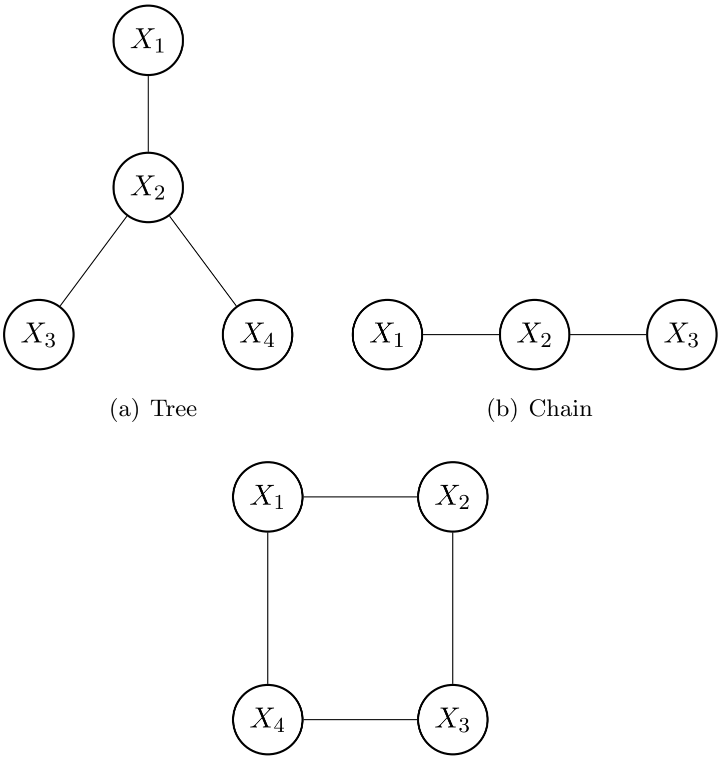
Definition 6. ***(Clique, Maximal Clique)*** *Let* $\mathcal{G}$ *be a graph and* $\mathbf{C} \subseteq \mathbf{X}$ *a subset of the nodes of the graph.* $\mathbf{C}$ *is a* clique*, if there exists an edge between all pairs of nodes in* $\mathbf{C}$*, i.e.*

$$\forall C_i, C_j \in \mathbf{C}, i \neq j, : (C_i - C_j) \in \mathbf{E}. \tag{2.11}$$

A clique is called maximal*, if adding any node* $X \in \mathbf{X} \setminus \mathbf{C}$ *makes it no longer a clique.*

Definition 7. ***(Path)*** *Let $\mathcal{G}$ be a graph. A sequence of nodes $Q = (X_1, \ldots, X_n)$ is a* path *from $X_1, \ldots, X_n$ if*

$$(X_i - X_{i+1}) \in \mathbf{E}, \ \textit{for } 1 \leq i \leq n-1. \tag{2.12}$$



(a) Tree

(b) Chain

(c) Single Loop

Figure 2.1: Graphical representation of different graph structures.

In this thesis, we consider three basic graph structures:

1. *Trees* are graphs in which any two different nodes are connected by exactly one path. As an example, consider the tree shown in Figure 2.1 (a), given by

$$\mathcal{G}_{BTree} = (\{X_1, X_2, X_3, X_4\}, \{(X_1 - X_2), (X_2 - X_3), (X_2 - X_4)\})\,. \tag{2.13}$$

2. *Chains* are a special case of trees for which the maximal degree is 2 for all vertices $X \in \mathbf{X}$. As an example, consider the chain with three nodes shown in Figure 2.1 (b), given by

$$\mathcal{G}_{Chain3} = (\{X_1, X_2, X_3\}, \{(X_1 - X_2), (X_2 - X_3)\})\,. \tag{2.14}$$

3. *Loops* are graphs in which for at least one vertex $X_i \in \mathbf{X}$ there exists a path from X_i to X_i. As an example, consider the single loop with four nodes shown in Figure 2.1 (c), given by

$$\mathcal{G}_{2\times 2SG} = (\{X_1, X_2, X_3, X_4\}, \{(X_1 - X_2), (X_2 - X_3), (X_3 - X_4), (X_4 - X_1)\})\,. \tag{2.15}$$

It can be seen that every *finite acyclic graph*, i.e. every graph without loops and a finite number of nodes, can be formed by concatenating trees. A proof of this result is shown in [25, pp.

311].

2.3 Markov Networks

In this section, we introduce an undirected graphical model, known as Markov Networks (MN), or Markov Random Fields. Certain factorization and conditional independence properties of a joint probability distribution can be expressed in such a way. They can be defined as follows:

Definition 8. ***(Markov Network)*** *A* Markov Network *is a tuple $\mathcal{M} = (\mathcal{G}, \boldsymbol{\Psi})$, where $\mathcal{G}$ is an undirected graph with nodes $\mathbf{X} = \{X_1, \dots, X_N\}$ representing RVs, and $\mathbf{C}_1, \dots, \mathbf{C}_L$ are maximal cliques in $\mathcal{G}$. The set $\boldsymbol{\Psi} = \{\Psi_{\mathbf{C}_1}, \dots, \Psi_{\mathbf{C}_L}\}$ is called set of* potentials*, and $\Psi_{\mathbf{C}_i} : \mathbf{val}(\mathbf{C}_i) \mapsto \mathbb{R}_{\geq 0}$ are nonnegative functions.*

The joint probability distribution of $\mathbf{X}$ defined by the MN is given by

$$p_{\mathcal{M}}(X_1, \dots, X_N) = \frac{1}{Z} \prod_{l=1}^{L} \Psi_{\mathbf{C}_l}(\mathbf{C}_l), \tag{2.16}$$

where Z is a normalization constant (also referred to as partition function) calculated as

$$Z = \sum_{\mathbf{x} \in \mathbf{val}(\mathbf{X})} \prod_{l=1}^{L} \Psi_{\mathbf{C}_l}(\mathbf{x}(\mathbf{C}_l)). \tag{2.17}$$

Using the above definition, we can compute the joint probability distribution for the MNs generated by the example graphs from Section 2.2. These joint probabilities are later used in Chapter 5. For the tree $\mathcal{G}_{BTree}$, the chain $\mathcal{G}_{Chain3}$, and the single loop $\mathcal{G}_{2\times 2SG}$, respectively, are given by

$$p_{Btree}(X_1, X_2, X_3, X_4) = \frac{1}{Z} \Psi_{X_1,X_2}(X_1, X_2) \Psi_{X_2,X_3}(X_2, X_3) \Psi_{X_2,X_4}(X_2, X_4), \tag{2.18}$$

$$p_{Chain3}(X_1, X_2, X_3) = \frac{1}{Z} \Psi_{X_1,X_2}(X_1, X_2) \Psi_{X_2,X_3}(X_2, X_3), \text{and} \tag{2.19}$$

$$p_{2\times 2SG}(X_1, X_2, X_3, X_4) = \frac{1}{Z} \Psi_{X_1,X_2}(X_1, X_2) \Psi_{X_2,X_3}(X_2, X_3) \Psi_{X_3,X_4}(X_3, X_4) \Psi_{X_4,X_1}(X_4, X_1). \tag{2.20}$$

2.4 Probabilistic Inference

In this section, concept of probabilistic inference is briefly reviewed. Lets assume that $\mathbf{X}$, the set of RVs in a MN $\mathcal{M}$, is now partitioned into the mutually disjoint sets $\mathbf{O}$ and $\mathbf{Q}$, i.e. $\mathbf{X} = \mathbf{O} \cup \mathbf{Q}$ and $\mathbf{O} \cap \mathbf{Q} = \emptyset$. The variables $\mathbf{O}$ are called *observed nodes*, referring to the observed evidence variables, and $\mathbf{Q}$ denotes the set of query variables. Given some observations (or evidence), the task of probabilistic inference using PGMs consists in assessing the marginal or the most likely configuration of variables [5] . There are two kinds of inference queries:

1. **Marginalization**: This query tries to infer the marginal distribution of the query variables $\mathbf{Q}$ conditioned on the observation $\mathbf{O}$. Using Eq. (2.7), the conditional probability $p(\mathbf{Q}|\mathbf{O} =$

o) is given by

$$p(\mathbf{Q}|\mathbf{O}=\mathbf{o}) = \frac{p(\mathbf{Q},\mathbf{O}=\mathbf{o})}{p(\mathbf{O}=\mathbf{o})}. \tag{2.21}$$

Using Eq. (2.6), the term $p(\mathbf{O}=\mathbf{o})$ can be computed as

$$p(\mathbf{O}=\mathbf{o}) = \sum_{\mathbf{q}\in\mathbf{val}(\mathbf{Q})} p(\mathbf{Q}=\mathbf{q},\mathbf{O}=\mathbf{o}). \tag{2.22}$$

2. **Maximum a-posteriory (MAP)**: This query tries to infer the most likely instantiation of the query variables $\mathbf{Q}$ given the observations $\mathbf{O}$, i.e.

$$\mathbf{q}^* = \arg\max_{\mathbf{q}\in\mathbf{Q}} p(\mathbf{Q}=\mathbf{q}|\mathbf{O}=\mathbf{o}). \tag{2.23}$$

Both of these queries can be answered directly evaluating the sums in their corresponding equations. Nevertheless this approach becomes intractable for models with many variables [3, 26]. As an example, let us consider the marginal $p(\mathbf{O}=\mathbf{o})$ from Eq. (2.22). Using Eq. (2.5) it can be seen that the computation of the marginal involves $\prod_{i=1}^{|\mathbf{Q}|}\mathbf{sp}(Q_i)$ summations. If we assume that $\mathbf{sp}(Q_i)=k$ for all Q_i's, the number of summations required to evaluate the marginal $p(\mathbf{O}=\mathbf{o})$ simplifies to $k^{|\mathbf{Q}|}$. This implies, that the complexity of the computation of the marginal $p(\mathbf{O}=\mathbf{o})$ is $O(k^{|\mathbf{Q}|})$, i.e. it grows exponentially with the number of variables.

MPAs are efficient methods developed for probabilistic inference, by exploiting the factorization of the joint probability induced by the graph structure [26]. A solution for the marginalization query, can be found using the BP algorithm, also known as the *sum-product algorithm* [3, 16]. For solving the MAP query, the corresponding MPAs are the *max-product algorithm*, or its alternative formulation in log-domain, the *max-sum algorithm* [5, 11]. The focus of this thesis is on marginalization queries, i.e. on the sum-product algorithm.

2.5 Belief Propagation

BP is a procedure which calculates the marginal distribution for each unobserved node, conditioned on the observed nodes. BP is an iterative process which can be seen as neighboring variables *passing messages* to each other, like: "I, variable X_i, *think* on how likely is it that you, variable X_j, are in state x_j". This series of conversations are likely to converge to a consensus after enough iterations, which determines the marginal probabilities of all variables. These estimated marginals are called *beliefs* [7].

Formally, let $\mathcal{M} = (\mathcal{G},\mathbf{\Psi})$ be a MN, with $\mathbf{X} = \{X_1,\ldots,X_N\}$ variable nodes. A *message* $\mu_{X_i\to X_j}(x_j)$ from (variable) node X_i to (variable) node X_j represents how much X_i *believes* that X_j will be in the state $x_j \in \mathbf{val}(X_j)$. The *belief* $b_{X_j}(x_j)$ of a variable node X_j to be in state x_j is proportional to the product of all messages from the neighboring factor nodes, i.e. $\mu_{X_i\to X_j}(x_j)$ for all $X_i \in \mathbf{Nb}(X_j)$, i.e.

$$b_{X_j}(x_j) = \frac{1}{Z}\prod_{X_i\in\mathbf{Nb}(X_j)} \mu_{X_i\to X_j}(x_j), \tag{2.24}$$

where Z is a normalization constant, such that $\sum_i b_{X_j}(x_j) = 1$. This message passing update is graphically represented in Figure 2.2. We now prove that the BP algorithm converges to

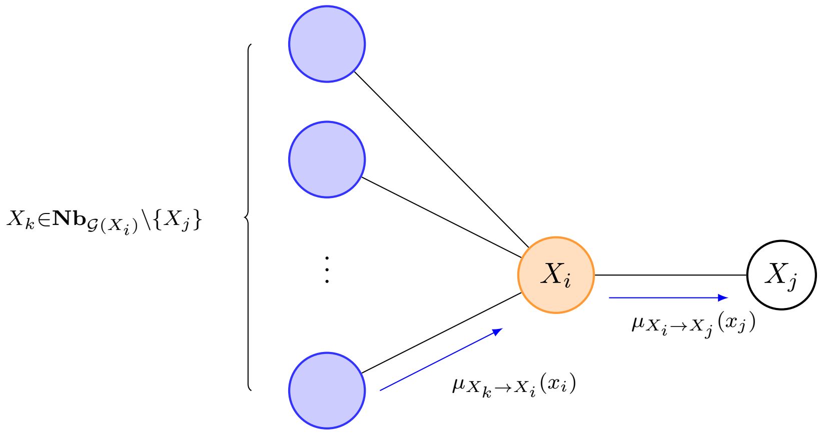


Figure 2.2: Representation for the message passing from X_i to X_j

the true marginal probabilities for the case of graphs without loops. While this result was first introduced in [7], in this thesis we present a proof that shows the polynomial nature of the BP algorithm.

Theorem 1. ***(Belief Propagation)*** *Let $\mathcal{M} = (\mathcal{G}, \mathbf{\Psi})$ be an acyclic pairwise MN, i.e. the maximal cliques consist of only two variables, with $\mathbf{X} = \{X_1, \ldots, X_N\}$ variable nodes. Then the belief of X_j calculated as in Eq.* (2.24) *is equal to the marginal probability of X_j, i.e.*

$$p_{\mathcal{M}}(X_j) = b_{X_j}(x_j), \tag{2.25}$$

if the messages $\mu_{X_i \to X_j}$ are computed as

$$\mu_{X_i \to X_j}(x_j) = \sum_{x_i \in \mathbf{val}(X_i)} \Psi_{X_i, X_j}(x_i, x_j) \prod_{X_k \in \mathbf{Nb}(X_i) \setminus \{X_j\}} \mu_{X_k \to X_i}(x_i). \tag{2.26}$$

Proof. While the core of this Theorem is the BP algorithm proposed [7], in this proof, we make emphasis on the polynomial nature of the MPE described by Eq. (2.26). Using the conditional independence relations described by the MN allow us to algebraically manipulate the beliefs. Without loss of generality, we compute $b_{X_1}(x_1)$. The variables in $\mathbf{X}$ can be relabeled, such that $\mathcal{G}$ has the tree structure shown in Figure 2.3, where X_1 is the root. We use $\mathbf{I}^1_1 = \mathbf{Nb}(X_1) = \{X_{I^1_1}, \ldots, X_{I^1_m}\}$ to denote the neighbor variable nodes of X_1, i.e. the nodes located in level $\mathbf{I}^1$. The set $\mathbf{I}^{k+1}_{I^k_i} = \mathbf{Nb}(X_{I^k_i}) \setminus \{X_{I^{k-1}_j}\}$ denotes the neighbors of variable node $X_{I^k_i}$, i.e. the i-th variable node in level $\mathbf{I}^k$, that are entirely in level $\mathbf{I}^{k+1}$. We use $\mathbf{x}_{I^k_i}$ as a shorthand notation for $x_{I^k_i} \in \mathbf{val}(X_{I^k_i})$. Substituting the messages from Eq. (2.26) in Eq. (2.24), the beliefs can be

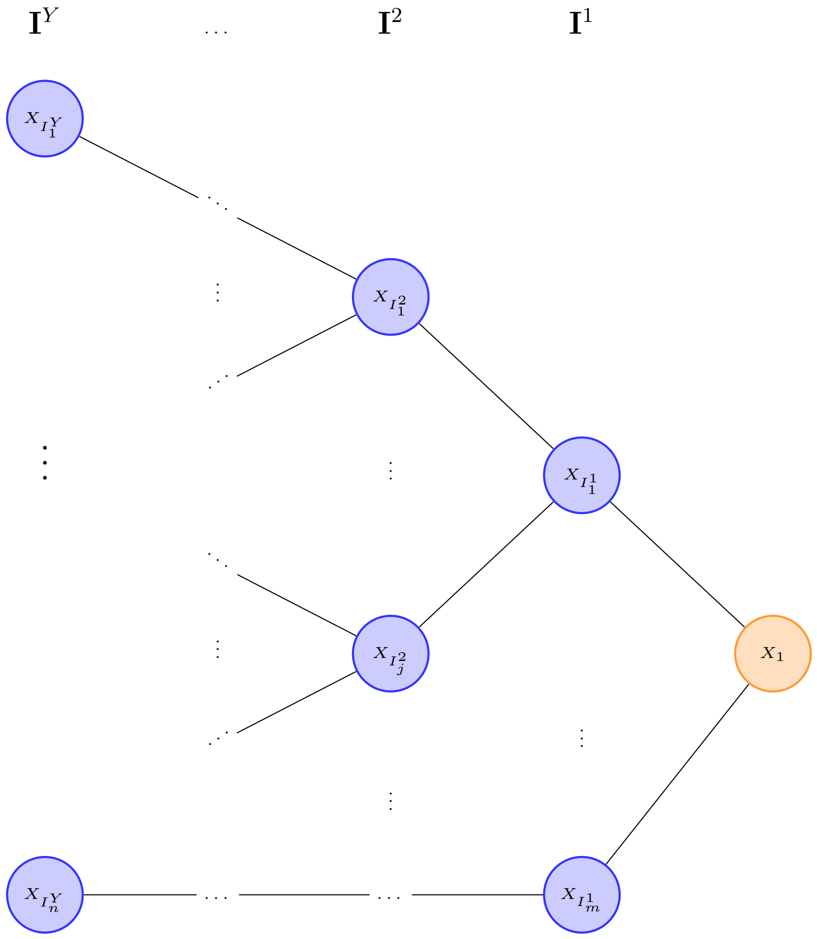


Figure 2.3: Topology of a general tree used for the proof of Theorem 1

computed as

$$\begin{aligned} b_{X_1}(x_1) &= \frac{1}{Z} \prod_{X_i \in \mathbf{I}_1^1} \mu_{X_i \to X_1}(x_1) \\ &= \frac{1}{Z} \left(\sum_{\mathbf{x}_{I_1^1}} \Psi_{X_{I_1},X_1}(x_{I_1^1}, x_1) \prod_{X_k \in \mathbf{I}^2_{I_1^1}} \mu_{X_k \to X_{I_1^1}}(x_{I_1^1}) \right) \times \dots \\ &\quad \times \left(\sum_{\mathbf{x}_{I_m^1}} \Psi_{X_{I_m^1},X_1}(x_{I_m^1}, x_1) \prod_{X_k \in \mathbf{I}^2_{I_m^1}} \mu_{X_k \to X_{I_m^1}}(x_{I_m^1}) \right) . \end{aligned} \tag{2.27}$$

Expanding every product of the form $\prod_k \mu_{k \to i}(x_i)$ results in

$$\begin{aligned} \prod_{X_k \in \mathbf{I}^2_{I_i^1}} \mu_{X_k \to X_{I_i^1}}(x_{I_i^1}) = \left(\sum_{\mathbf{x}_{I_j^2}} \Psi_{X_{I_j^2},X_{I_i^1}}(x_{I_j^2}, x_{I_i^1}) \left(\sum_{\mathbf{x}_{I_l^3}} \Psi_{X_{I_l^3},X_{I_j^2}}(x_{I_l^3}, x_{I_j^2}) \Bigg(\right.\right. \\ \left.\left. \dots \left(\sum_{\mathbf{x}_{I_z^Y}} \Psi_{X_{I_z^Y},X_{I_y^{Y-1}}}(x_{I_z^Y}, x_{I_y^{Y-1}}) \right) \dots \Bigg) \right) \dots \right) . \end{aligned} \tag{2.28}$$

Using that the sum is distributive over products (see Section 3.1), and that every potential is pairwise defined, the above equation can be written as

$$\prod_{X_k \in \mathbf{I}^2_{I_i^1}} \mu_{X_k \to X_{I_i^1}}(x_{I_i^1}) = \sum_{\mathbf{x}_{I_j^2}} \cdots \sum_{\mathbf{x}_{I_z^Y}} \Psi_{X_{I_j^2}, X_{I_i^1}}(x_{I_j^2}, x_{I_i^1}) \dots \Psi_{X_{I_z^Y}, X_{I_y^{Y-1}}}(x_{I_z^Y}, x_{I_y^{Y-1}}). \tag{2.29}$$

Here we recognize that the above equation is the sum over all states $x_{I_i^k} \in \mathbf{val}(X_{I_i^k})$ of variable nodes $X_{I_i^k}$ in levels $\mathbf{I}^2, \dots, \mathbf{I}^Y$ along the path that started in $X_{I_i^1}$. Because of the tree structure of the graph, there is no variable $X_{I_q^k}$ that belongs the neighborhoods of both $X_{I_i^{k-1}}$ and $X_{I_j^{k-1}}$ for all $i \neq j$. Hence, every pairwise potential appears only once, and there is only a summation in every variable, (i.e. the maximal degree of every pairwise potential is 1). It can be seen, that this statement is not true in loopy graphs. Substituting Eq. (2.29) in Eq. (2.27) results in

$$\begin{aligned} b_{X_1}(x_1) = \frac{1}{Z} & \underbrace{\left(\sum_{\mathbf{x}_{I_1^1}} \Psi_{X_{I_1^1}, X_1}(x_{I_1^1}, x_1) \sum_{\mathbf{x}_{I_j^2}} \cdots \sum_{\mathbf{x}_{I_z^Y}} \Psi_{X_{I_j^2}, X_{I_i^1}}(x_{I_j^2}, x_{I_i^1}) \dots \Psi_{X_{I_z^Y}, X_{I_y^{Y-1}}}(x_{I_z^Y}, x_{I_y^{Y-1}}) \right)}_{\mu_{X_{I_1^1} \to X_1(x_1)}} \times \dots \\ & \times \underbrace{\left(\sum_{\mathbf{x}_{I_m^1}} \Psi_{X_{I_m^1}, X_1}(x_{I_m^1}, x_1) \sum_{\mathbf{x}_{I_{j'}^2}} \cdots \sum_{\mathbf{x}_{I_{z'}^Y}} \Psi_{X_{I_{j'}^2}, X_{I_{i'}^1}}(x_{I_{j'}^2}, x_{I_{i'}^1}) \dots \Psi_{X_{I_{z'}^Y}, X_{I_{y'}^{Y-1}}}(x_{I_{z'}^Y}, x_{I_{y'}^{Y-1}}) \right)}_{\mu_{X_{I_m^1} \to X_1}(x_1)} \\ & = \frac{1}{Z} \sum_{\mathbf{x}_{I_1^1}} \cdots \sum_{\mathbf{x}_{I_n^Y}} \Psi_{X_{I_1^1}, X_1}(x_{I_1^1}, x_1) \dots \Psi_{X_{I_n^Y}, X_{I_z^{Y-1}}}(x_{I_n^Y}, x_{I_z^{Y-1}}). \end{aligned} \tag{2.30}$$

Comparing the right-hand-side of the above equation with Eq. (2.16), the belief $b_{X_1}(x_1)$ simplifies to

$$b_{X_1}(x_1) = \sum_{x_{I_1}} \cdots \sum_{x_{Y_n}} p_{\mathcal{M}}(X_1, \dots, X_{Y_n}), \tag{2.31}$$

which according with Eq. (2.6), is the marginal probability $p_{\mathcal{M}}(X_1)$. □

It should be noted, that the messages per se do not necessarily represent probabilities and need to be normalized. Nevertheless, to avoid numerical instabilities [2,3,16], a common formulation of the MPE defined by Eq. (2.26) includes a normalization as follows:

$$\mu_{X_i \to X_j}(x_j) = Z_{i \to j} \sum_{x_i \in \mathbf{val}(X_i)} \Psi_{X_i, X_j}(x_i, x_j) \prod_{X_k \in \mathbf{Nb}(X_i) \setminus \{X_j\}} \mu_{X_k \to X_i}(x_i), \tag{2.32}$$

where $Z_{i \to j}$ is a normalization constant[2] such that

$$\sum_{x_j \in \mathbf{val}(X_j)} \mu_{X_i \to X_j}(x_j) = 1. \tag{2.33}$$

The above equations can be understood as a system of polynomial equations, where the messages $\mu_{X_i \to X_j}(x_j)$ and the normalization constants $Z_{i \to j}$ are the variables. Theorem 1 guarantees that as long as the graph has a tree-like structure, the beliefs calculated obtained from message

[2] Since the potentials are strictly positive functions, it is easy to see that $Z_{i \to j} > 0$. Contrary to references [3, 5, 11, 14, 16], in this thesis we decided to define this constant as a multiplicative factor, in order to use the theoretical framework of commutative algebra from Chapter 3. A similar formulation can be found in [17].

passing converge to the true marginals. The reason behind this exactness is attributed to the fact that in a tree-like graph, messages received by a node from its neighbors are independent. This does not always hold in presence of loops, which make the neighbors of a node correlated and therefore, the messages are no longer independent [26]. Nevertheless, convergence of BP in graphs with cycles has been experimentally confirmed for many applications [27]. This *Loopy Belief Propagation* for general graphs has been successfully applied in many areas [5,14]. In this case, the (L)BP, shown in Algorithm 2.1, is said to converge to a solution [5,11,16], although not necessarily to the true marginals, if there is a finite $k > 0$ such that

$$\mu^{(k)}_{X_i \to X_j}(x_j) = \mu_{X_i \to X_j}(x_j). \tag{2.34}$$

Algorithm 2.1: *LBP*$(\cdot)$ (Loopy) Belief Propagation

```
(Adapted from [26])
input  : a MN $\mathcal{M} = (\mathcal{G}, \{\Psi_{\mathbf{C}_1}, \dots, \Psi_{\mathbf{C}_L}\})$;
         maximum number of iterations $k_{max}$;
         requested precision $\epsilon$
output: the set of beliefs $\{b_{X_j}(x_j) \mid X_j \in \mathbf{X}\}$

Initialize messages $m^{(0)}_{X_i \to X_j}(x_j) = 1$ for all pairs of variable nodes $X_i, X_j$ for which
$(X_i - X_j) \in \mathbf{E}$.
for $k = 1 \colon k_{max}$ do
    $\forall (X_i - X_j) \in \mathbf{E}$:
      1. Update:
           $m^{(k)}_{X_i \to X_j}(x_j) := \sum_{x_i \in \mathbf{val}(X_i)} \Psi_{X_i, X_j}(x_i, x_j) \prod_{X_l \in \mathbf{Nb}(X_i) \setminus \{X_j\}} m^{(k-1)}_{X_l \to X_i}(x_i).$   (2.35)
      2. Compute normalization constant:
           $Z^{(k)}_{i \to j} := \frac{1}{\sum_{x_j \in \mathbf{val}(X_j)} m^{(k)}_{X_i \to X_j}(x_j)}$   (2.36)
      3. Normalize messages:
           $\mu^{(k)}_{X_i \to X_j}(x_j) := Z^{(k)}_{i \to j} m^{(k)}_{X_i \to X_j}(x_j)$   (2.37)
    if $|\mu^{(k+1)}_{X_i \to X_j}(x_j) - \mu^{(k)}_{X_i \to X_j}(x_j)| < \epsilon \ \forall (X_i - X_j) \in \mathbf{E}$ then
        break;
if $k = k_{max}$ then
    return UNCONVERGED;
    else for $X_i \in \mathbf{X}$ do
        $b_{X_i}(x_i) := \prod_{X_j \in \mathbf{Nb}(X_i)} \mu_{X_j \to X_i}(x_i)$
        return $\{b_{X_j}(x_j) \mid X_j \in \mathbf{X}\}$
    ;
```

The work of Yedidia et al. represents perhaps the most significant breakthrough in the study of convergence of the (L)BP algorithm. Their work came with the insight that the (L)BP converge to stationary points of an approximate free energy, known as Bethe free energy in statistical physics [14, 27]. The investigation of the fixed points of the Bethe free energy has led to the derivation of convergence criteria for the (L)BP algorithm. Following this work, several convergence criteria for BP have been proposed, including the work of Ihler et al. [16] and Mooij et al. [17]. Using results from linear algebra, Weiss showed that the LBP algorithm converges for graphs with a single loop [3].

In Chapter 4, a reformulation of this convergence condition in terms of computational commutative algebra is given. In order to do so, we use the fact that the equations describing the messages involved in BP can be seen as a system of multivariate polynomials, and thus, solving the convergence problem is analog to the problem of finding the solutions of a system of polynomial equations.

3 Computational Commutative Algebra

As discussed in Chapter 2, the Belief Propagation algorithm can be interpreted as finding the roots of a system of polynomial equations. This system is explicitly shown in Eq. (2.32) and Eq. (2.33). Therefore it would be interesting to investigate some theoretical background that allows us to manipulate and solve such a system. In this chapter, the framework of computational commutative algebra used for this purpose is provided. If not stated otherwise, the definitions and notation are taken from the book by Cox, Little and O' Shea [18]. We refer the reader to this standard text for a more extensive treatment of this subject.

The rest of this chapter is structured as follows: In Section 3.1, a short overview of the basic concepts of algebraic geometry and commutative algebra is provided. In Section 3.2, the concept of Gröbner Basis is reviewed. In Section 3.3, the Hilbert's Nullstellensatz and its connection to the existence and number of solutions of a system of polynomial equations are discussed. We close this Chapter with the basics of elimination theory as well as some applications to solving systems of polynomial equations.

3.1 Algebraic Geometry

Algebraic Geometry is the study of systems of polynomial equations and its relation with geometrical objects. The solutions of a system of polynomial equations form a geometric object called *variety*, whose corresponding algebraic object is called *ideal*. These concepts allow us to answer such questions as whether a system of polynomial equations has finitely or infinitely many solutions, and how to characterize them [18, 19]. We begin our discussion about solutions of polynomials by defining the basic algebraic structures called rings and fields. A more formal definition of these structures is provided in Appendix B.2.

Definition 9. ***(Ring)*** *A* ring *is a triple* $(\mathbb{A}, +, \cdot)$*, where* $\mathbb{A}$ *is a set, and* $\cdot$ *and* $+$ *are binary operations defined on* $\mathbb{A}$ *for which the following conditions are satisfied:*

1. *(associative)* $(a+b)+c = a+(b+c)$ *and* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ $\forall\ a, b, c \in \mathbb{A}$.
2. *(commutative)* $a+b = b+a$ *and* $a \cdot b = b \cdot a$ $\forall a, b\ \in \mathbb{A}$.

3. *(distributive)* $a \cdot (b + c) = a \cdot b + a \cdot c \; \forall a, b, c \in \mathbb{A}$.
4. *(identities) There are* $0, 1 \in \mathbb{A}$ *such that* $a + 0 = a$ *and* $a \cdot 1 = a \; \forall a \in \mathbb{A}$.
5. *(additive inverse) Given* $a \in \mathbb{A}$, *there is* $b \in \mathbb{A}$ *such that* $a + b = 0$.

A common example of a ring is the set of all integer numbers $\mathbb{Z}$. As a notation remark, in this work, $\mathbb{Z}_{\geq 0}$ denotes the set of all integers equal or larger than zero.

Definition 10. ***(Field)*** *A* field *is a triple* $(\mathbb{K}, +, \cdot)$, *where* $\mathbb{K}$ *is a set, and* $\cdot$ *and* $+$ *are binary operations defined on* $\mathbb{K}$ *for which the following conditions are satisfied*

1. *(associative)* $(a + b) + c = a + (b + c)$ *and* $(a \cdot b) \cdot c = a \cdot (b \cdot c) \; \forall \; a, b, c \in \mathbb{K}$,
2. *(commutative)* $a + b = b + a$ *and* $a \cdot b = b \cdot a \; \forall a, b \; \in \mathbb{K}$,
3. *(distributive)* $a \cdot (b + c) = a \cdot b + a \cdot c \; \forall a, b, c \in \mathbb{K}$,
4. *(identities) There are* $0, 1 \in \mathbb{K}$ *such that* $a + 0 = a$ *and* $a \cdot 1 = a \; \forall a \in \mathbb{K}$,
5. *(additive inverse) Given* $a \in \mathbb{K}$, *there is* $b \in \mathbb{K}$ *such that* $a + b = 0$,
6. *(multiplicative inverse) Given* $a \neq 0 \in \mathbb{K}$, *there is* $c \in \mathbb{K}$ *such that* $a \cdot c = 1$

Common examples of fields are the set of all rational numbers $\mathbb{Q}$, the set of real numbers $\mathbb{R}$ and the set of complex numbers $\mathbb{C}$. With this definitions, we can study the most important algebraic structure used in this thesis, a polynomial ring.

Definition 11. ***(Monomial, Total degree, Polynomial, Polynomial Ring)*** *Given a field* $\mathbb{K}$; $x_1, \ldots, x_n$; $a_1, \ldots, a_n \in \mathbb{K}$ *and* $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}_{\geq 0}$, *a* monomial *in* $x_1, \ldots, x_n$ *is a product in the form*

$$x_1^{\alpha_1} \ldots x_n^{\alpha_n}. \tag{3.1}$$

The total degree *of a monomial is given as*

$$|\alpha| = \sum_{i}^{n} \alpha_i. \tag{3.2}$$

As a shorthand notation, we will write $x_1^{\alpha_1} \ldots x_n^{\alpha_n} = x^\alpha$. *A* polynomial f *in* $x_1, \ldots, x_n$ *with coefficients in* $\mathbb{K}$ *is a finite linear combination of monomials, that can be written as*

$$f = \sum_{\alpha} a_\alpha x^\alpha. \tag{3.3}$$

The set of all polynomials in $x_1, \ldots, x_n$ *with coefficients in* $\mathbb{K}$ *is called a* polynomial ring, *denoted by* $\mathbb{K}[x_1, \ldots, x_n]$, *which satisfies all the conditions of a commutative ring for the sum and product of polynomials.*

A field $\mathbb{K}$ is said to be *algebraically closed* if it contains the roots of every non-constant polynomial $f \in \mathbb{K}[x_1, \ldots, x_n]$. The field of real numbers $\mathbb{R}$ is not algebraically closed, because $f(x) = x^2 + 1$ has no root in $\mathbb{R}$. On the other hand, $\mathbb{C}$ is an example of an algebraically closed field. Using these definitions, it is possible to introduce the basic geometric objects called varieties.

Definition 12. ***(Affine Space, Affine Variety)*** *Let* $\mathbb{K}$ *be a field and* $n \in \mathbb{Z}_{\geq 0}$. *The set*

$$\mathbb{K}^n = \{(a_1, \ldots, a_n) \colon a_1, \ldots, a_n \in \mathbb{K}\} \tag{3.4}$$

is called the affine space *over* $\mathbb{K}$*. Furthermore, let* $f_1, \ldots, f_s$ *be polynomials in* $\mathbb{K}[x_1, \ldots, x_n]$*, then the set*

$$\mathbf{V}(f_1, \ldots, f_s) = \{(a_1, \ldots, a_s) \in \mathbb{K}^n \colon f_i(a_1, \ldots, a_s) = 0 \ \forall \ 1 \leq i \leq s\} \tag{3.5}$$

is called the affine variety *defined by* $f_1, \ldots, f_s$ *over the affine space* $\mathbb{K}^n$*.*

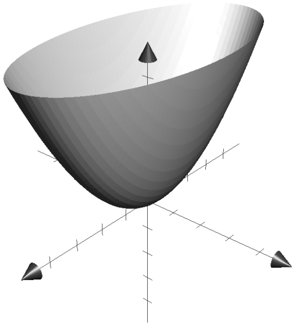


Figure 3.1: Example of a Variety in $\mathbb{R}^3$*: Paraboloid* $\frac{1}{2}x^2 + 2y^2 - z = 0$

As an example, Figure 3.1 shows the variety $\mathbf{V}(\frac{1}{2}x^2 + 2y^2 - z) \subset \mathbb{R}^3$, i.e. the paraboloid given by the set of all points that satisfy $\frac{1}{2}x^2 + 2y^2 - z = 0$.

The analog algebraic objects that allow us to characterize varieties are ideals. These objects can be defined as follows:

Definition 13. ***(Ideal, Finitely generated Ideal, Basis of an Ideal)*** *A subset of a polynomial ring* $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ *is an* ideal *if it satisfies the following conditions:*

1. $0 \in \mathbf{I}$,
2. $a, b \in \mathbf{I} \ \Rightarrow \ a + b \in \mathbf{I}$ *and*
3. *if* $a \in \mathbf{I}$ *and* $b \in \mathbb{K}[x_1, \ldots, x_n]$*, then* $ab \in \mathbf{I}$.

A finitely generated ideal*, is the one that can be generated by a finite set of polynomials* $\{f_1, \ldots, f_s\}$*, called* basis*, defined as*

$$\langle f_1, \ldots, f_s \rangle = \{f \mid f = g_1 f_1 + \cdots + g_s f_s, \ g_i \in \mathbb{K}[x_1, \ldots, x_n]\}. \tag{3.6}$$

The concept of an ideal is similar to that of a vector subspace in linear algebra, while the notion of a basis of an ideal is analog to the basis of a vector space [18]. The relationship between finitely generated ideals and varieties can be stated in the following lemma:

Lemma 1. *Let* $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ *be an ideal and* $\mathbf{V}(\mathbf{I}) \subset \mathbb{K}^n$ *be an affine variety given by*

$$\mathbf{V}(\mathbf{I}) = \{(a_1, \ldots, a_n) \in \mathbb{K}^n \colon f(a_1, \ldots, a_n) = 0 \ \forall f \in \mathbf{I}\}. \tag{3.7}$$

If $\mathbf{I} = \langle f_1, \ldots, f_s \rangle$*, then* $\mathbf{V}(\mathbf{I}) = \mathbf{V}(f_1, \ldots, f_s)$*.*

The proof of this lemma can be found in [18, pp. 79]. This result guarantees that the affine variety of a system of polynomial equations is exactly the same as the variety of the ideal generated by the polynomials of such a system. The following lemma further explores the relationship between varieties and ideals, by connecting the concepts of bases and varieties of an ideal:

Lemma 2. *If $f_1, \ldots, f_s$ and $g_1, \ldots, g_t$ are bases of the same ideal $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$, i.e. $\langle f_1, \ldots, f_s \rangle = \langle g_1, \ldots, g_t \rangle$ it follows that $\mathbf{V}(f_1, \ldots, f_s) = \mathbf{V}(g_1, \ldots, g_t)$.*

The proof of this lemma can be found in [18, pp. 33]. This result is interesting, because it states that different bases of the same ideal are associated with the same variety. Using this lemma, the problem of finding the roots of a system of polynomial equations can be simplified by finding an appropriate basis which generates the same ideal (and thus, the same variety).

3.2 Gröbner Basis

Next, we introduce the concept of ordering, which proves to be very useful at characterizing monomials, polynomials, ideals and varieties [18, 20].

Definition 14. ***(Monomial ordering)*** *A* monomial ordering *is any relation $>$ on a set of monomials $\{x^\alpha \mid \alpha \in \mathbb{Z}^n_{\geq 0}\}$, which satisfies the following conditions:*

1. *$>$ is a* total ordering *on $\mathbb{Z}^n_{\geq 0}$, which means that only one of the three possibilities of $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$ should be true, i.e.*

$$x^\alpha > x^\beta, \qquad x^\alpha = x^\beta, \qquad x^\beta > x^\alpha, \tag{3.8}$$

2. *if $\alpha > \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$, then $\alpha + \gamma > \beta + \gamma$, and*
3. *every nonempty subset of $\mathbb{Z}^n_{\geq 0}$ has a smallest element under $>$.*

Some important monomial orderings are the lexicographic ordering, or lex order, the graded lexicographic ordering, or grlex order, and the graded reverse lexicographic order, or grevlex order [20, 28]. These orderings are defined as follows:

Definition 15. ***(Lex order, Grlex order, Grevlex order).*** *Let $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$. We denote*

1. Lexicographic ordering*: $\alpha >_{lex} \beta$ if in the vector difference $\alpha - \beta = [\alpha_1 - \beta_1, \ldots, \alpha_n - \beta_n]$ the leftmost nonzero entry is positive.*
2. Graded Lexicographic ordering*: $\alpha >_{grlex} \beta$ if $|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i$, or $|\alpha| = |\beta|$ and $\alpha >_{lex} \beta$.*
3. Graded reverse lexicographic ordering*: $\alpha >_{grevlex} \beta$ if $|\alpha| > |\beta|$, or $|\alpha| = |\beta|$ and the rightmost nonzero entry of $\alpha - \beta$ is negative.*

With the concept of ordering, we can use the following terminology to describe the structure of a polynomial:

Definition 16. ***(Multidegree, Leading Coefficient, Leading Monomial, Leading Term).*** *Let $f = \sum_\alpha a_\alpha x^\alpha$ be a nonzero polynomial in $\mathbb{K}[x_1, \ldots, x_n]$, and $>$ be a monomial ordering, then*

1. Multidegree*:*

$$\text{multideg}(f) = \max(\alpha \in \mathbb{Z}^n_{\geq 0} \colon a_\alpha \neq 0), \tag{3.9}$$

where the maximum is taken with respect to the ordering $>$.

2. Leading Coefficient*:*

$$\text{LC}(f) = a_{\text{multideg}(f)} \in \mathbb{K}. \tag{3.10}$$

3. Leading Monomial*:*

$$\text{LM}(f) = x^{\text{multideg}(f)} \in \mathbb{K}[x_1, \dots, x_n]. \tag{3.11}$$

4. Leading Term*:*

$$\text{LT}(f) = \text{LC}(f) \cdot \text{LM}(f) \in \mathbb{K}[x_1, \dots, x_n]. \tag{3.12}$$

Definition 17. ***(Least common multiple, S-polynomial)*** *Let $f, g \in \mathbb{K}[x_1, \dots, x_n]$ be two polynomials, with* multideg$(f) = \alpha$ *and* multideg$(g) = \beta$*, and then let $\gamma = (\gamma_1, \dots, \gamma_n)$ with $\gamma_i = \max(\alpha_i, \beta_i)$. The* least common multiple *of* LM(f) *and* LM(g) *is given by*

$$\text{l.c.m}(\text{LM}(f), \text{LM}(g)) = x^\gamma. \tag{3.13}$$

The S-polynomial *of f and g is defined as*

$$S(f, g) = \frac{\text{l.c.m}(\text{LM}(f), \text{LM}(g))}{\text{LT}(f)} f - \frac{\text{l.c.m}(\text{LM}(f), \text{LM}(g))}{\text{LT}(g)} g. \tag{3.14}$$

Using these concepts, it is possible to generalize the algorithm for dividing a polynomial by another polynomial in one variable to an algorithm for dividing a polynomial by a set of polynomials in several variables. The main idea is that given $\mathbf{F} = (f_1, \dots, f_s)$, an s-tuple of polynomials in $\mathbb{K}[x_1, \dots, x_n]$, and a monomial ordering $>$, a polynomial $f \in \mathbb{K}[x_1, \dots, x_n]$ can be written as

$$f = a_1 f_1 + \cdots + a_s f_s + \overline{f}^{\mathbf{F}}, \tag{3.15}$$

where $a_1, \dots, a_s, \overline{f}^{\mathbf{F}} \in \mathbb{K}[x_1, \dots, x_n]$, and either $\overline{f}^{\mathbf{F}} = 0$ or $\overline{f}^{\mathbf{F}}$ is a linear combination of monomials, none of which is divisible by any of $\text{LT}(f_i), \dots, \text{LT}(f_s)$. In this context, $\overline{f}^{\mathbf{F}}$, the *remainder* of f on division by $\mathbf{F}$, is called *normal form* of f. Algorithm 3.1 shows how to compute $a_1, \dots, a_s, \overline{f}^{\mathbf{F}}$. A proof for this algorithm can be found in [18, pp. 64].

We are interested in using these concepts to test if a polynomial $f \in \mathbb{K}[x_1, \dots, x_n]$ shares zeros with a set of polynomials $\{f_1, \dots, f_s\} \subset \mathbb{K}[x_1, \dots, x_n]$. This problem can be formulated as determining if $f \in \mathbf{I}$, with $\mathbf{I} = \langle f_1, \dots, f_s \rangle$. Geometrically speaking, using Lemma 1, this is equivalent as deciding whether $\mathbf{V}(\mathbf{I})$ lies on the variety of $\mathbf{V}(f)$. This is known in the literature as the Ideal Membership problem [18]. We can intuitively understand this problem with an example. Let us assume, that we have a system of polynomial equations $\mathbf{F} = \mathbf{0}$, and an polynomial equation $f = 0$ such that $f \in \mathbf{F}$. We are interested in determining if $f = 0$ has common solutions to $\mathbf{F} \setminus \{f\} = \mathbf{0}$. Finding a systematic way to knowing if such solutions exist would be helpful to determine if a system of polynomial equations has a solution. The Ideal Membership problem can be solved using a special kind of bases called *Gröbner Bases*. They can be characterized

Algorithm 3.1: *DivAlg*$(\cdot)$ Division Algorithm in $\mathbb{K}[x_1, \ldots, x_n]$

```
(Taken from [18])
input  : $f_1, \ldots, f_s, f$
output: $a_1, \ldots, a_s, \overline{f}^{\mathbf{F}}$
Initialize $a_1 := 0; \ldots ; a_s := 0; \overline{f}^{\mathbf{F}} := 0$ and $p := f$
while $p \neq 0$ do
    $i := 1$
    $\text{div}_{\text{occ}} := false$
    while $i \leq s$ and $\text{div}_{\text{occ}} := false$ do
        if $\text{LT}(f_i)$ divides $\text{LT}(p)$ then
            $a_i := a_i + \frac{\text{LT}(p)}{\text{LT}(f_i)}$
            $p := p - \frac{\text{LT}(p)}{\text{LT}(f_i)} f_i$
            $\text{div}_{\text{occ}} := true$
        else
            $i := i + 1$
    if $\text{div}_{\text{occ}} = false$ then
        $\overline{f}^{\mathbf{F}} := \overline{f}^{\mathbf{F}} + \text{LT(p)}$
        $p := p - \text{LT}(p)$
```

using the following Theorem:

Theorem 2. ***(Gröbner Basis)*** *Given $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ an ideal, and a fixed monomial order, the following statements are equivalent*

1. $\mathbf{G} = \{g_1, \ldots, g_t\}$ *is a* GB *for* $\mathbf{I}$.
2. *Let* $\text{LT}(\mathbf{I})$ *be the set of leading terms of elements of* $\mathbf{I}$ *and* $\langle \text{LT}(\mathbf{I}) \rangle$ *the ideal generated by those elements, then*
$$\text{LT}(\mathbf{I}) = \langle \text{LT}(g_i), \ldots, \text{LT}(g_t) \rangle. \tag{3.16}$$
3. (Ideal Membership) *For* $f \in \mathbb{K}[x_1, \ldots, x_n]$*,* $f \in \mathbf{I}$ *iff* $\overline{f}^{\mathbf{G}} = 0$*, i.e. the remainder of* f *on division by* $\mathbf{G}$ *is zero.*
4. (Buchberger's Criterion) $\overline{S(g_i, g_j)}^{\mathbf{G}} = 0$ *for all pairs of polynomials* $g_i, g_j \in \mathbf{G}$.

Proof. (Taken from [18])

$1 \Leftrightarrow 2$. Statement 2 is the definition of a GB. To prove that this is a basis for $\mathbf{I}$, i.e.

$$\mathbf{I} = \langle g_1, \ldots, g_t \rangle = \{h \mid h = h_1 g_1 + \cdots + h_t g_t,\ h_i \in \mathbb{K}[x_1, \ldots, x_n]\}, \tag{3.17}$$

Dickinson's Lemma and the Hilbert Basis Theorem are required. These results can be found in [18, pp. 71 and pp. 76, respectively].

$1 \Leftrightarrow 3$. Let's suppose that $f \in \mathbf{I}$. Since $\mathbf{G}$ is a basis for $\mathbf{I}$, f can be written as $f = h = h_1 g_1 + \cdots + h_t g_t$. Comparing with Eq. (3.15) it follows that $\overline{f}^{\mathbf{G}} = 0$. Conversely, if $\overline{f}^{\mathbf{G}} = 0$, and $\mathbf{G}$ is a GB, it is trivial to see that f lies in $\mathbf{I}$.

$3 \Leftrightarrow 4$. Let's assume that $\mathbf{G}$ is a GB of $\mathbf{I}$. Using Eq. (3.14), we can write the S-polynomial as

$$\begin{aligned} S(g_i, g_j) &= \frac{\text{l.c.m}(\text{LM}(g_i), \text{LM}(g_j))}{\text{LT}(g_i)} g_i - \frac{\text{l.c.m}(\text{LM}(g_i), \text{LM}(g_j))}{\text{LT}(g_j)} g_j \\ &= h_i g_i + h_j g_j, \end{aligned} \tag{3.18}$$

which means that $S(g_i, g_j) \in \mathbf{I}$. By Statement 3, this means that $\overline{S(g_i, g_j)}^{\mathbf{G}} = 0$. For the sake of brevity, the proof of the converse statement, i.e. starting from $\overline{S(g_i, g_j)}^{\mathbf{G}} = 0$, show that $\mathbf{G}$ is a GB of $\mathbf{I}$, is omitted here, but it can be found in [18, pp. 85]. □

The third statement of this theorem suggests a method for transforming an arbitrary basis $\mathbf{F}$ of an ideal $\mathbf{I}$ into a Gröbner basis. Such a method was first proposed by Bruno Buchberger in 1965 [19], and is shown in Algorithm 3.2. Nevertheless, this algorithm is computationally expensive since it involves the computation of all pairs of S-polynomials, and therefore heavily depends on a good selection of the monomial ordering [29, 30]. A weak criterion to avoid the computation of some useless S-polynomials, i.e. a criterion to determine beforehand if an S-polynomial is reduced to zero, and thus, avoid computing the remainder using the generalized division algorithm, is presented in the following proposition:

Proposition 1. *Given a finite set* $\mathbf{G} \subset \mathbb{K}[x_1, \ldots, x_n]$ *and polynomials* $f, g \in \mathbf{G}$, *such that* $\text{l.c.m}(\text{LT}(f), \text{LT}(g)) \neq \text{LT}(f), \text{LT}(g)$, *then* $\overline{S(f, g)}^{\mathbf{G}} = 0$.

Algorithm 3.2: *Groebner*$(\cdot)$ Buchberger's Algorithm

```
(Taken from [18])
input  : $\mathbf{F} = \{f_1, \ldots, f_s\}$, a basis for $\mathbf{I}$, and $>$, a monomial ordering
output: $\mathbf{G} = \{g_1, \ldots, g_t\}$ a GB for $\mathbf{I}$
Initialize $\mathbf{G} := \mathbf{F}$
$\mathrm{g}_{\mathrm{eq}} := false$
while $\mathrm{g}_{\mathrm{eq}} = false$ do
    $\mathbf{G}' := \mathbf{G}$
    for each pair $\{p, q\}$, $p \neq q$ in $\mathbf{G}'$ do
        if $\text{l.c.m}(\text{LT}(p), \text{LT}(q)) \neq \text{LT}(p), \text{LT}(q)$ then
            $S := \overline{S(p, q)}^{\mathbf{G}'}$, using $DivAlg(S(p, q), \mathbf{G}')$
            if $S \neq 0$ then
                $\mathbf{G} := \mathbf{G} \cup \{S\}$
    if $\mathbf{G}' = \mathbf{G}$ then
        $\mathrm{g}_{\mathrm{eq}} := true$
```

The proof for this proposition can be found in [18, pp. 104]. More recent alternatives to Buchberger's algorithm like the F_4 algorithm by Faugére [30], include more sophisticated criteria to avoid the computation of useless S-polynomials, as well as linear algebra techniques to improve the performance of the generalized division algorithm. Such methods are included in most commercial computer algebra systems. However it should be noted that the performance of these algorithms heavily depends on $\mathbb{K}$, i.e. the field in which the coefficients of the polynomials lie, and a good selection of a monomial ordering. It has been experimentally shown that computing a GB with respect to the grevlex order is usually faster than computing a GB with respect to the lex order or the grlex order [31]. The Gröbner bases described by Theorem 2 are not unique,

Algorithm 3.3: $rGB(\cdot)$ Reduced Gröbner Basis

```
(Taken from [18])
input  : $\mathbf{G} = \{g_1, \dots, g_t\}$, a GB basis for $\mathbf{I}$
output: $\mathbf{G}^{'} = \{p_1, \dots, p_s\}$ a reduced GB for $\mathbf{I}$

for $g_i \in \mathbf{G}$ do
    $g_i := \frac{1}{\mathrm{LC}(g_i)} g_i$
for $g_i \in \mathbf{G}$ do
    if $g_i \in \langle \mathrm{LT}(\mathbf{G} \setminus \{g_i\}) \rangle$ then
        $\mathbf{G} := \mathbf{G} \setminus \{g_i\}$
for $g_i \in \mathbf{G}$ do
    $p_i := \overline{g_i}^{\mathbf{G} \setminus \{g_i\}}$, using $DivAlg(g_i, \mathbf{G} \setminus \{g_i\})$
return $\mathbf{G}^{'} = \{p_1, \dots, p_s\}$
```

i.e. for an ideal $\mathbf{I}$, several GBs can be found. This leads to the definition of *reduced Gröbner bases*. In addition to the conditions above, further requirements for a GB to be a reduced GB are that $\mathrm{LC}(g) = 1$, and no monomial of g lies in $\langle \mathrm{LT}(\mathbf{G} \setminus \{g\}) \rangle$ for all $g \in \mathbf{G}$. A reduced GB is unique for a given monomial order. A full proof of this conditions lies beyond the scope of this thesis, but it can be found in [18, pp. 92]. An algorithm for computing a reduced GB is shown in Alg. 3.3.

3.3 Hilbert's Nullstellensatz

One of the most important applications of GBs is determining if a system of polynomial equations has a solution. This can be interpreted geometrically as knowing whether the variety generated by such a system is non-empty. Hilbert's Nullstellensatz answers this question.

Theorem 3. ***((Weak) Hilbert's Nullstellensatz)*** *Let $\mathbb{K}$ be an algebraically closed field and let $\mathbf{I} \subset \mathbb{K}[x_1, \dots, x_n]$ be an ideal satisfying $\mathbf{V}(\mathbf{I}) = \emptyset$. Then $\mathbf{I} = \mathbb{K}[x_1, \dots, x_n]$.*

A proof by induction can be found in [18, pp. 170]. A criterion for determining the existence of common zeros of polynomial equations can be expressed as follows:

Corollary 1. *Let $\mathbb{K}$ be an algebraically closed field, and $\{f_1, \dots, f_s\} \subset \mathbb{K}[x_1, \dots, x_n]$ be a set of polynomials. Then if the reduced Gröbner basis of the ideal generated by these polynomials with respect to any ordering is $\mathbf{G} = \{1\}$, they do not have a common zero.*

Proof. (Taken from [18])

The ideal generated by $\mathbf{G}$ is $\mathbf{I} = \langle 1 \rangle$, which can be written as

$$\mathbf{I} = \{h \mid h = h_i \cdot 1,\ h_i \in \mathbb{K}[x_1, \dots, x_n]\}, \tag{3.19}$$

which means that every polynomial $h_i \in \mathbb{K}[x_1, \dots, x_n]$ is also an element of $\mathbf{I}$, and therefore $\mathbb{K}[x_1, \dots, x_n] \subset \mathbf{I}$. Since by definition $\mathbf{I} \subset \mathbb{K}[x_1, \dots, x_n]$, this implies that $\mathbf{I} = \mathbb{K}[x_1, \dots, x_n]$, and therefore, by the weak Nullstellensatz, $\mathbf{V}(\mathbf{I}) = \emptyset$. From Lemma 1, it follows that $\mathbf{V}(\mathbf{I}) = \mathbf{V}(1)$. Since $\mathbf{G}$ is the basis of the ideal generated by $\{f_1, \dots, f_s \in \mathbb{K}[x_1, \dots, x_n]$, i.e. $\mathbf{I} = \langle f_1, \dots, f_s \rangle = \langle 1 \rangle$. Using Lemma 2, $\mathbf{V}(f_1, \dots, f_s) = \mathbf{V}(1) = \emptyset$, and therefore, the polynomials $\{f_1, \dots, f_s\}$ do not have common zeros in $\mathbb{K}$. □

The following proposition allows us to generalize the above result to the case of non-reduced GBs:

Proposition 2. *Let* $\mathbf{G}$ *be a GB for the ideal* $\mathbf{I}$ *with respect to a fixed monomial ordering* $>$ *and let* $c \in \mathbb{K}[x_1, \ldots, x_n]$ *be a constant polynomial. The reduced GB* $\mathbf{G}^{'}$ *of* $\mathbf{I}$ *with respect to the monomial ordering* $>$ *is given by* $\mathbf{G}^{'} = \{1\}$ *iff* $c \in \mathbf{G}$.

Proof. First, we show that if $c \in \mathbf{G}$, it follows that $\mathbf{G}^{'} = \{1\}$. We construct $\mathbf{G}^{'}$ using Algorithm 3.3. Without loss of generality, the GB can be written as $\mathbf{G} = \{f_1, \ldots, c, \ldots, f_s\}$. Dividing all polynomials in $\mathbf{G}$ by their leading coefficients results in

$$\mathbf{G}^{'} = \left\{\frac{f_1}{\mathrm{LC}(f_1)}, \ldots, 1, \ldots, \frac{f_s}{\mathrm{LC}(f_s)}\right\}. \tag{3.20}$$

It is easy to see that $\langle 1 \rangle = \mathbb{K}[x_1, \ldots, x_n]$. Since $1 \in \mathbf{G}^{'}$ and all polynomials $\frac{f_i}{\mathrm{LC}(f_i)} \in \mathbb{K}[x_1, \ldots, x_n]$, it follows that $\frac{f_i}{\mathrm{LC}(f_i)} \in \left\langle \mathrm{LT}(\mathbf{G}^{'} \setminus \left\{\frac{f_i}{\mathrm{LC}(f_i)}\right\}\right\rangle$ for all f_i's in $\mathbf{G}$ not equal to c, and hence, all of these polynomials are discarded from $\mathbf{G}^{'}$. Finally, it follows that the only polynomial remaining in $\mathbf{G}^{'}$ is 1, and therefore, $\mathbf{G}^{'} = \{1\}$.

Now, it we show that if $\mathbf{G}^{'} = \{1\}$, it follows that $c \in \mathbf{G}$. Assuming that $\mathbf{G}^{'} = \{1\}$ implies that the polynomial $\frac{f_i}{\mathrm{LC}(f_i)}$, with $f_i \neq 1$ in the original non-reduced GB $\mathbf{G}$, lies in the ideal generated by the leading monomials of the polynomials in $\mathbf{G}$ minus the leading monomial of the leading monomial of f_i, i.e. $\frac{f_i}{\mathrm{LC}(f_i)} \in \left\langle \mathrm{LT}(\mathbf{G}^{'} \setminus \left\{\frac{f_i}{\mathrm{LC}(f_i)}\right\}\right\rangle$. This is true for all polynomials $f_i \neq 1 \in \mathbf{G}$. Adding these polynomials to $\mathbf{G}^{'}$ results in

$$\mathbf{G}^{'} = \left\{1, \frac{f_1}{\mathrm{LC}(f_1)}, \ldots, \frac{f_s}{\mathrm{LC}(f_s)}\right\}. \tag{3.21}$$

Finally, multiplying every polynomial in $\mathbf{G}^{'}$ by its leading coefficient, which is a constant in $\mathbb{K}$, results in $\mathbf{G} = \{c, f_1, \ldots, f_s\}$, with c a constant. This means that $c \in \mathbf{G}$. □

It follows from this proposition, that if a constant polynomial is an element of the GB of the ideal generated by a system of polynomial equations, the affine variety of such a system is empty, and therefore, the system has no solution.

We are now interested in determining whether an affine variety $\mathbf{V}(\mathbf{I})$ is a finite set. This question can be answered knowing the dimension of this variety, but a more thorough discussion about the concept of dimension of a variety lies beyond the scope of this chapter. A more formal treatment of this concept is provided in Appendix C.

Definition 18. ***(Zero-dimensional Ideal)*** *An ideal* $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ *is called* zero-dimensional *if the affine variety* $\mathbf{V}(\mathbf{I})$ *is a finite set.*

It follows from the previous definition that a system of polynomial equations has only a finite number of solutions if the ideal generated by those polynomials is zero-dimensional. The following theorem provides a method for determining if a system of equations has finitely many solutions:

Theorem 4. *Let* $\mathbf{V}(\mathbf{I}) \subset \mathbb{K}^n$ *be an affine variety and* $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ *be an ideal,* $>$ *be a graded monomial ordering in* $\mathbb{K}[x_1, \ldots, x_n]$*, and* $\mathbf{G}$ *a GB for* $\mathbf{I}$ *with respect to such ordering.* $\mathbf{I}$ *is zero-dimensional iff for each* i*,* $1 \leq i \leq n$*, there is some* $m_i \geq 0$ *such that* $x_i^{m_i} = \mathrm{LT}(g)$ *for some* $g \in \mathbf{G}$.

The proof of this theorem is omitted here, since it uses results from the formal definition of dimension of a variety, but it is included in Appendix C. The following result proposes a quantitative estimate of the number of solutions of a system of polynomial equations:

Proposition 3. *Let* $\mathbf{I} \subset \mathbb{K}[x_1, \dots, x_n]$ *be an ideal in an algebraically closed field with GB* $\mathbf{G} = \{g_1, \dots, g_t\}$ *such that* $\mathrm{LT}(g_i) = x_i^{m_i}$*. Then it follows that the variety* $\mathbf{V}(\mathbf{I})$ *contains at most* $m_1 \times m_2 \times \cdots \times m_n$ *points.*

The proof of this proposition can be found in Appendix C.

3.4 Elimination Theory

Given a system of polynomial equations $\mathbf{F} \subset \mathbb{K}[x_1, \dots, x_n]$, elimination theory allows us to use the theoretical framework of commutative algebra to find the solutions to $\mathbf{F} = \mathbf{0}$ in two steps:

1. (Elimination Step) Find a consequence $g_t(x_n) = 0 \in \mathbb{K}[x_1, \dots, x_n]$ of the original equations in $\mathbf{F}$, which involves only x_n, i.e. eliminates all other variables $x_1, \dots, x_{n-1}$ from the system.
2. (Extension Step) Once $g_t = 0$ is solved, determine values of x_n that could extend these solutions to solutions of the original system $\mathbf{F} = \mathbf{0}$.

In order to generalize these ideas, the following definition is required:

Definition 19. ***(Elimination Ideal)*** *Let* $\mathbf{I} = \langle f_1, \dots, f_2 \rangle \in \mathbb{K}[x_1, \dots, x_n]$ *be an ideal. The* l*-th* elimination ideal $\mathbf{I}_l$ *is the ideal given by*

$$\mathbf{I}_l = \mathbf{I} \cap \mathbb{K}[x_{l+1}, \dots, x_n] \tag{3.22}$$

The l-th elimination ideal consists of all consequences of $f_1 = \cdots = f_s = 0$ which eliminate the variables $x_1, \dots, x_l$, and is an ideal of $\mathbb{K}[x_{l+1}, \dots, x_n]$. Therefore, the elimination of variables $x_1, \dots, x_l$ consists of finding nonzero polynomials in $\mathbf{I}_l$. The following theorem allows us to do that systematically:

Theorem 5. ***(Elimination Theorem)*** *Let* $\mathbf{I} \subset \mathbb{K}[x_1, \dots, x_n]$ *be an ideal, and* $\mathbf{G}$ *its GB with respect to the lex order. Then for every* $0 \leq l \leq n$ *the set*

$$\mathbf{G}_l = \mathbf{G} \cap \mathbb{K}[x_{l+1}, \dots, x_n] \tag{3.23}$$

is the GB of the l*-th elimination ideal* $\mathbf{I}_l$

The proof of this theorem can be found in [18, pp. 117]. Using this result, a strategy for solving systems of polynomial equations $f_1 = \cdots = f_s = 0$ using GBs can be found in Algorithm 3.4. In this way, solving systems of polynomial equations using the formalisms of GBs represents a natural generalization of the Gaussian algorithm for solving systems of linear equations [18].

Compared to the grlex, the grevlex and other monomial orderings, computation of GBs with respect to the lex order is often more difficult. Therefore, a usual strategy for finding the zeros of a system of polynomial equations that generate a zero-dimensional ideal is to compute the GB of such an ideal with respect to a more efficient monomial ordering then to *convert* this GB into a lexicographic GB [29,33]. Among the most popular ordering-conversion methods for GBs

Algorithm 3.4: *PolySolve*!$(\cdot)$ Solutions of a system of polynomials

```
(Taken from [18])
input  : $\mathbf{F} = \{f_1 = 0, \ldots, f_s = 0\}$, a system of polynomial equations in $\mathbb{K}[x_1, \ldots, x_n]$
output: Sols the set solutions of F
Assume: $\mathbf{I} = \langle f_1, \ldots, f_s \rangle$ is zero-dimensional.
Sols := $\emptyset$
G := $Groebner(\mathbf{F}, >_{lex})$, the GB of $\langle f_1, \ldots, f_s \rangle$ with respect to the lex order
G := $rGB(\mathbf{G})$, the reduced GB of $\langle f_1, \ldots, f_s \rangle$
$r := Roots(g_n)$, the roots of generator in $x_n$ by applying one-variable techniques (including numerical methods such as Newton-Raphson [18, 32])
Sols := Sols $\cup \{r\}$
for $i := 1 : n - 1$ do
    Compute $r := Roots(g_{n-i})$ the roots of the generator in $x_i$ applying back substitution.
    Sols := Sols $\cup \{r\}$
return Sols
```

are the Gröbner Walk [33] and the FGLM [31] algorithms. These methods are built-in in most commercial computer algebra systems. Worth mentioning is the work of Jean Charles Faugère, who has not only contributed with the development of the F_4 and FGLM algorithms, but also wrote the standard `Groebner` package in MAPLE.

4

Algebraic Formulation of BP

Using the framework of computational commutative algebra presented in Chapter 3, it is possible to express an alternative formulation of the Belief Propagation algorithm. In order to do so we need to formalize the notion of the system of polynomials that represent the message passing equations:

Definition 20. ***(Associated set of polynomials of a MN)*** *Given a MN $\mathcal{M}$ and its system of MPE defined by the BP algorithm, i.e. the message updates from Eq.* (2.32) *and the normalization constraints from Eq.* (2.33) *(Section 2.5), its* associated set of polynomials *(ASP)* $\mathbf{F}_{\mathcal{M}} \subset \mathbb{K}[\mu_{1\to2}(x_1), \ldots, \mu_{2\to1}(x_n), Z_{1\to2}, \ldots, Z_{2\to1}]$ *is given as*

$$\mathbf{F}_{\mathcal{M}} = \left\{ \begin{array}{l} \mu_{i\to j}(x_j) - Z_{i\to j} \sum_{x_i} \Psi_{i,j}(x_i, x_j) \prod_k \mu_{k\to i}(x_i) \\ \sum_{x_j} \mu_{i\to j}(x_j) - 1 \end{array} : \forall\ \{i,j,k\} \in \mathcal{G},\ x_i \in \mathbf{val}(i) \right\} \tag{4.1}$$

It is important to remark that the variables of the polynomials in $\mathbf{F}_{\mathcal{M}}$ are the messages $\mu_{i\to j}(x_j)$ *and* the normalization constants $Z_{i\to j}$. To avoid cluttered notation, the shorthand notation $\mathbf{F} = \mathbf{F}_{\mathcal{M}}$ will be used, whenever it is clear that the set is associated with the MN $\mathcal{M}$. As an example, the ASP of the MN of the tree shown in Figure 2.1, whose graph $\mathcal{G}_{BTree}$ is given in

Eq. (2.13), with binary variables $\mathbf{val}(\mathbf{X}) = \{x, \bar{x}\}\ \forall X_i \in \mathbf{X}$ is given as

$$\mathbf{F}_{Btree} = \left\{ \begin{array}{l}
\mu_{1\to 2}(x) - Z_{1\to 2}\left(\Psi_{X_1,X_2}(x,x) + \Psi_{X_1,X_2}(\bar{x},x)\right) \\
\mu_{1\to 2}(\bar{x}) - Z_{1\to 2}\left(\Psi_{X_1,X_2}(x,\bar{x}) + \Psi_{X_1,X_2}(\bar{x},\bar{x})\right) \\
\mu_{2\to 1}(x) - Z_{2\to 1}\left(\Psi_{X_1,X_2}(x,x)\mu_{3\to 2}(x)\mu_{4\to 2}(x) + \Psi_{X_1,X_2}(x,\bar{x})\mu_{3\to 2}(\bar{x})\mu_{4\to 2}(\bar{x})\right) \\
\mu_{2\to 1}(\bar{x}) - Z_{2\to 1}\left(\Psi_{X_1,X_2}(\bar{x},x)\mu_{3\to 2}(x)\mu_{4\to 2}(x) + \Psi_{X_1,X_2}(\bar{x},\bar{x})\mu_{3\to 2}(\bar{x})\mu_{4\to 2}(\bar{x})\right) \\
\mu_{2\to 3}(x) - Z_{2\to 3}\left(\psi_{X_2}(x)\Psi_{X_2,X_3}(x,x)\mu_{1\to 2}(x)\mu_{4\to 2}(x) + \Psi_{X_2,X_3}(\bar{x},x)\mu_{1\to 2}(\bar{x})\mu_{4\to 2}(\bar{x})\right) \\
\mu_{2\to 3}(\bar{x}) - Z_{2\to 3}\left(\psi_{X_2}(x)\Psi_{X_2,X_3}(x,\bar{x})\mu_{1\to 2}(x)\mu_{4\to 2}(x) + \Psi_{X_2,X_3}(\bar{x},\bar{x})\mu_{1\to 2}(\bar{x})\mu_{4\to 2}(\bar{x})\right) \\
\mu_{2\to 4}(x) - Z_{2\to 4}\left(\psi_{X_2}(x)\Psi_{X_2,X_4}(x,x)\mu_{1\to 2}(x)\mu_{3\to 2}(x) + \Psi_{X_2,X_4}(\bar{x},x)\mu_{1\to 2}(\bar{x})\mu_{3\to 2}(\bar{x})\right) \\
\mu_{2\to 4}(\bar{x}) - Z_{2\to 4}\left(\psi_{X_2}(x)\Psi_{X_2,X_4}(x,\bar{x})\mu_{1\to 2}(x)\mu_{3\to 2}(x) + \Psi_{X_2,X_4}(\bar{x},\bar{x})\mu_{1\to 2}(\bar{x})\mu_{3\to 2}(\bar{x})\right) \\
\mu_{3\to 2}(x) - Z_{3\to 2}\left(\Psi_{X_2,X_3}(x,x) + \Psi_{X_2,X_3}(x,\bar{x})\right) \\
\mu_{3\to 2}(\bar{x}) - Z_{3\to 2}\left(\Psi_{X_2,X_3}(\bar{x},x) + \Psi_{X_2,X_3}(\bar{x},\bar{x})\right) \\
\mu_{4\to 2}(x) - Z_{4\to 2}\left(\Psi_{X_2,X_4}(x,x) + \Psi_{X_2,X_4}(x,\bar{x})\right) \\
\mu_{4\to 2}(\bar{x}) - Z_{4\to 2}\left(\Psi_{X_2,X_4}(\bar{x},x) + \Psi_{X_2,X_4}(\bar{x},\bar{x})\right) \\
\mu_{1\to 2}(x) + \mu_{1\to 2}(\bar{x}) - 1 \\
\mu_{2\to 1}(x) + \mu_{2\to 1}(\bar{x}) - 1 \\
\mu_{2\to 3}(x) + \mu_{2\to 3}(\bar{x}) - 1 \\
\mu_{2\to 4}(x) + \mu_{2\to 4}(\bar{x}) - 1 \\
\mu_{3\to 2}(x) + \mu_{3\to 2}(\bar{x}) - 1 \\
\mu_{4\to 2}(x) + \mu_{4\to 2}(\bar{x}) - 1
\end{array} \right\}. \tag{4.2}$$

The rest of this chapter is structured as follows: In Section 4.1, we revisit the condition of convergence of the BP in terms of determining the dimension and cardinality of $\mathbf{V}_{\mathcal{M}}$, the affine variety generated by the associated set of polynomials of a MN. In Section 4.2 an equivalent alternative to the BP algorithm using Gröbner basis is proposed.

4.1 Convergence of the (L)BP algorithm

In Chapter 2, it was shown that convergence of the iterative (L)BP algorithm required the message updates calculated with Eq. (2.37) to converge to the normalized messages after enough iterations, i.e. $\mu^{(k)}_{X_i\to X_j}(x_j) = \mu_{X_i\to X_j}(x_j)$. This is equivalent to the following proposition:

Proposition 4. *Let $\mathcal{M}$ be a MN and $\mathbf{F}$ be the ASP of $\mathcal{M}$. The (L)BP algorithm converges to a solution (not necessarily the true marginals), if the system of equations defined by the set $\mathbf{F}$ as vector (i.e. each polynomial $f_i \in \mathbf{F}$ represents the i-th component of vector $\mathbf{F}$) vanishes.*

Proof. From the message update rule from Eq. (2.37), we have that

$$\mu^{(k)}_{X_i\to X_j}(x_j) - Z^{(k)}_{i\to j} \sum_{x_i \in \mathbf{val}(X_i)} \Psi_{X_i,X_j}(x_i,x_j) \prod_{X_l \in \mathbf{Nb}(X_i)\backslash\{X_j\}} m^{(k-1)}_{X_l\to X_i}(x_i) = 0. \tag{4.3}$$

Using Eq. (2.36), we have that $m^{(k-1)}_{X_l\to X_i}(x_i) = \frac{1}{Z^{(k-1)}_{l\to i}} \mu^{(k-1)}_{X_l\to X_i}(x_i)$, which substituting in the above equation results in

$$\mu^{(k)}_{X_i\to X_j}(x_j) - Z^{(k)}_{i\to j} \prod_{l} \frac{1}{Z^{(k-1)}_{l\to i}} \sum_{x_i \in \mathbf{val}(X_i)} \Psi_{X_i,X_j}(x_i,x_j) \prod_{X_l \in \mathbf{Nb}(X_i)\backslash\{X_j\}} \mu^{(k-1)}_{X_l\to X_i}(x_i) = 0. \tag{4.4}$$

By construction, the messages computed with Algorithm 2.1 satisfy that

$$\sum_{x_j \in \mathbf{val}(X_j)} \mu^{(k)}_{X_i \to X_j}(x_j) - 1 = 0. \tag{4.5}$$

Using the convergence criterium $\mu^{(k)}_{X_i \to X_j}(x_j) = \mu_{X_i \to X_j}(x_j)$, we can rewrite Eq. (4.4) and Eq. (4.5) as

$$\begin{aligned} \mu_{X_i \to X_j}(x_j) - Z_{i \to j} \sum_{x_i \in \mathbf{val}(X_i)} \Psi_{X_i,X_j}(x_i, x_j) \prod_{X_k \in \mathbf{Nb}(X_i) \setminus \{X_j\}} \mu_{X_k \to X_i}(x_i) &= 0 \\ \sum_{x_j \in \mathbf{val}(X_j)} \mu_{X_i \to X_j}(x_j) - 1 &= 0, \end{aligned} \tag{4.6}$$

where $Z_{i \to j} = Z^{(k)}_{i \to j} \prod_l \frac{1}{Z^{(k-1)}_{l \to i}}$. This condition is valid for every message. Comparing Eq. (4.6) with Eq. (4.1) implies that convergence of the (L)BP algorithm is ensured if $\mathbf{F} = \mathbf{0}$ has a solution. □

It is easy to see, that using the convergence condition from Proposition 4, we can use computational commutative algebra to analyze conditions for convergence of the (L)BP algorithm. A criterion for convergence can be stated as follows:

Lemma 3. *Let* $\mathbf{F}$ *be the ASP of* $\mathcal{M}$*, a MN, and* $\mathbf{G_F}$ *be the reduced Gröbner basis of* $\mathbf{I}$*, the ideal generated by* $\mathbf{F}$ *with respect to an arbitrary monomial ordering. The BP algorithm does not converge to a solution if:*

1. $\mathbf{G_F} = \{1\}$*, i.e. there is no solution to* $\mathbf{F} = \mathbf{0}$*,*
2. $\mathbf{I}$ *is not zero-dimensional, i.e. there are infinitely many solutions to* $\mathbf{F} = \mathbf{0}$*.*

Proof. Let $\mathbf{V}(\mathbf{I})$ be the affine variety of $\mathbf{I}$. Since $\mathbf{I} = \langle f_1, \ldots, f_D \rangle$ by construction, then, by Lemma 1, $\mathbf{V}(\mathbf{I}) = \mathbf{V}(f_1, \ldots, f_D)$ for $f_i \in \mathbf{F}$, i.e. the set of solutions of the system $\mathbf{F} = \mathbf{0}$.

Assuming that $\mathbf{G_F} = \{1\}$ means that $\mathbf{V}(\mathbf{I}) = \emptyset$, which implies that the system $\mathbf{F} = \mathbf{0}$ do not have a solution, and thus the convergence condition from Proposition 4 is not satisfied. The result follows directly from Corollary 1 and is a consequence of the weak Nullstellensatz (Theorem 3).

If $\mathbf{I}$ is not zero-dimensional then, by definition, the variety $\mathbf{V}(\mathbf{I})$ is not a finite set, and thus, $\mathbf{F} = \mathbf{0}$ has infinitely many solutions. This implies that the BP algorithm doesn't converge to a fixed point. □

Using the above Lemma and properties of the GB, a method for determining the convergence of the BP algorithm can be expressed in the following Theorem:

Theorem 6. ***(Convergence of BP)*** *Let* $\mathbf{F} \subset \mathbb{K}[\mu_1, \ldots, \mu_D]$ *be the ASP of an MN* $\mathcal{M}$*, and* $\mathbf{G_F}$ *be the reduced GB of* $\mathbf{I}$*, the ideal generated by the polynomials in* $\mathbf{F}$*, with respect to some monomial ordering* $>$*. The BP algorithm converges to a solution if for every variable* μ_i *there exists an element* $g_j \in \mathbf{G_F}$ *whose leading term is* $\mathrm{LT}(g_j) = \mu_i^{m_i}$*, and the number of fixed points is bounded by* $m_1 \times \cdots \times m_D$*.*

Proof. If there exists an ordering $>$ such that the leading term of some $g_j \in \mathbf{G_F}$ can be written as $\mathrm{LT}(g_j) = \mu_i^{m_i}$, means that $\langle \mathrm{LT}(\mathbf{I}) \rangle = \mu_i^{m_i}$. From Theorem 4 follows that $\mathbf{I}$ is zero-dimensional. Furthermore, it also follows that $\mathbf{G_F} \neq \{1\}$, since $\mathbf{G_F}$ is a reduced GB. Hence, by Lemma 3, this means that the conditions for existence of solutions of $\mathbf{F} = \mathbf{0}$ are satisfied. From Proposition 3, it follows that the variety $\mathbf{V}(\mathbf{I})$ has at most n_{sols} solutions, with $1 \leq n_{sols} \leq m_1 \times \cdots \times m_D$. Hence, BP converges to a solution. □

Let's illustrate this results with a binary Markov chain with 3 nodes, $\mathcal{M}_{MChain3}$, whose graph is given by Eq. (2.14), (Fig. 2.1). The ASP of $\mathcal{M}_{MChain3}$ is given by

$$\mathbf{F} = \left\{ \begin{array}{l} \mu_{1\to2}(x) - Z_{1\to2}\left(\Psi_{X_1,X_2}(x,x) + \Psi_{X_1,X_2}(\bar{x},x)\right) \\ \mu_{1\to2}(\bar{x}) - Z_{1\to2}\left(\Psi_{X_1,X_2}(x,\bar{x}) + \Psi_{X_1,X_2}(\bar{x},\bar{x})\right) \\ \mu_{2\to1}(x) - Z_{2\to1}\left(\Psi_{X_1,X_2}(x,x)\mu_{3\to2}(x) + \Psi_{X_1,X_2}(x,\bar{x})\mu_{3\to2}(\bar{x})\right) \\ \mu_{2\to1}(\bar{x}) - Z_{2\to1}\left(\Psi_{X_1,X_2}(\bar{x},x)\mu_{3\to2}(x) + \Psi_{X_1,X_2}(\bar{x},\bar{x})\mu_{3\to2}(\bar{x})\right) \\ \mu_{2\to3}(x) - Z_{2\to3}\left(\Psi_{X_2,X_3}(x,x)\mu_{1\to2}(x) + \Psi_{X_2,X_3}(\bar{x},x)\mu_{1\to2}(\bar{x})\right) \\ \mu_{2\to3}(\bar{x}) - Z_{2\to3}\left(\Psi_{X_2,X_3}(x,\bar{x})\mu_{1\to2}(x) + \Psi_{X_2,X_3}(\bar{x},\bar{x})\mu_{1\to2}(\bar{x})\right) \\ \mu_{3\to2}(x) - Z_{3\to2}\left(\Psi_{X_2,X_3}(x,x) + \Psi_{X_2,X_3}(x,\bar{x})\right) \\ \mu_{3\to2}(\bar{x}) - Z_{3\to2}\left(\Psi_{X_2,X_3}(\bar{x},x) + \Psi_{X_2,X_3}(\bar{x},\bar{x})\right) \\ \mu_{1\to2}(x) + \mu_{1\to2}(\bar{x}) - 1 \\ \mu_{2\to1}(x) + \mu_{2\to1}(\bar{x}) - 1 \\ \mu_{2\to3}(x) + \mu_{2\to3}(\bar{x}) - 1 \\ \mu_{3\to2}(x) + \mu_{3\to2}(\bar{x}) - 1 \end{array} \right\} \tag{4.7}$$

If we let the variables be lexicographically ordered as

$$\begin{aligned} &\mu_{2\to3}(\bar{x}) >_{lex} \mu_{2\to3}(x) >_{lex} \mu_{2\to1}(\bar{x}) >_{lex} \mu_{2\to1}(x) >_{lex} Z_{2\to3} >_{lex} Z_{3\to1} \\ &>_{lex} Z_{2\to1} >_{lex} Z_{1\to2} >_{lex} \mu_{3\to2}(\bar{x}) >_{lex} \mu_{3\to2}(x) >_{lex} \mu_{1\to2}(\bar{x}) >_{lex} \mu_{1\to2}(x), \end{aligned} \tag{4.8}$$

the GB of the ideal generated by $\mathbf{F}$ with respect to the lex order is given by

$$\mathbf{G_F} = \left\{ \begin{array}{l} \mu_{1\to2}(x) - \frac{q_1}{(q_1+q_2)} \\ \mu_{1\to2}(\bar{x}) - \frac{q_2}{(q_1+q_2)} \\ \mu_{3\to2}(x) - \frac{q_{11}}{(q_{11}+q_{12})} \\ \mu_{3\to2}(\bar{x}) - \frac{q_{12}}{(q_{11}+q_{12})} \\ Z_{1\to2} - \frac{1}{(q_1+q_2)} \\ Z_{3\to2} - \frac{1}{(q_{11}+q_{12})} \\ Z_{2\to1} - \frac{q_{11}+q_{12}}{(q_{11}q_3+q_{11}q_5+q_{12}q_4+q_{12}q_6)} \\ Z_{2\to3} - \frac{q_1+q_2}{(q_1q_7+q_1q_9+2q_2q_8)} \\ \mu_{2\to1}(x) - \frac{q_{11}q_3+q_{12}q_4}{(q_{11}q_3+q_{11}q_5+q_{12}q_4+q_{12}q_6)} \\ \mu_{2\to1}(\bar{x}) - \frac{q_{11}q_5+q_{12}q_6}{(q_{11}q_3+q_{11}q_5+q_{12}q_4+q_{12}q_6)} \\ \mu_{2\to3}(x) - \frac{q_1q_7+q_2q_8}{(q_1q_7+q_1q_9+2q_2q_8)} \\ \mu_{2\to3}(\bar{x}) - \frac{q_1q_9+q_2q_8}{(q_1q_7+q_1q_9+2q_2q_8)} \end{array} \right\}, \tag{4.9}$$

where the factors $q_1, \dots, q_{12}$ are calculated as

$$\begin{aligned}
q_1 &= \Psi_{X_1,X_2}(x,x) + \Psi_{X_1,X_2}(\bar{x},x)\\
q_2 &= \Psi_{X_1,X_2}(x,\bar{x}) + \Psi_{X_1,X_2}(\bar{x},\bar{x})\\
q_3 &= \Psi_{X_1,X_2}(x,x)\\
q_4 &= \Psi_{X_1,X_2}(x,\bar{x})\\
q_5 &= \Psi_{X_1,X_2}(\bar{x},x)\\
q_6 &= \Psi_{X_1,X_2}(\bar{x},\bar{x})\\
q_7 &= \Psi_{X_2,X_3}(x,x)\\
q_8 &= \Psi_{X_2,X_3}(x,\bar{x})\\
q_9 &= \Psi_{X_2,X_3}(\bar{x},x)\\
q_{10} &= \Psi_{X_2,X_3}(\bar{x},\bar{x})\\
q_{11} &= \Psi_{X_1,X_2}(x,x) + \Psi_{X_1,X_2}(\bar{x},x)\\
q_{12} &= \Psi_{X_1,X_2}(x,\bar{x}) + \Psi_{X_1,X_2}(\bar{x},\bar{x}).
\end{aligned} \tag{4.10}$$

We can see that there is a polynomial $g \in \mathbf{G}$ for each variable $\mu_{i\to j}(k)$, whose leading term can be written in form $\mathrm{LT}(g) = (\mu_{i\to j}(k))^m$, with $m = 1$ for all variables. It follows from Theorem 6 that the BP algorithm for a MN with tree nodes converges to a single solution. We can expand this result to a general finite Markov Chain as follows:

Theorem 7. *The BP algorithm converges to a solution for the MN $\mathcal{M}$ being a Markov Chain.*

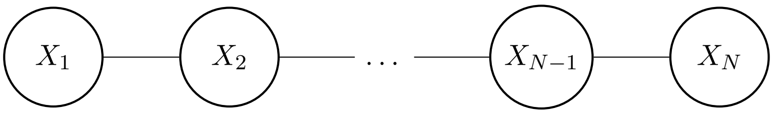


Figure 4.1: Graph of a Markov Chain with N nodes.

Proof. Let $\mathcal{M}$ be the MN of a Markov Chain with N nodes shown in Figure 4.1. We consider $\mathbf{F}$, the ASP of $\mathcal{M}$ with RVs that have m states, to be a set of polynomials that depends only on the message variables. The normalization constants are absorbed by the potentials $\Psi_{j,l}(\cdot,\cdot)$'s, since the normalization constants were introduced to avoid numerical instability, and are, therefore, not strictly necessary. $\mathbf{F}$ can be written as

$$\mathbf{F} = \left\{ \begin{array}{c}
\mu_{1\to 2}(x_1) - \sum_{i=1}^{m} \Psi_{1,2}(x_i, x_1)\\
\vdots\\
\mu_{1\to 2}(x_m) - \sum_{i=1}^{m} \Psi_{1,2}(x_i, x_m)\\
\vdots\\
\mu_{i\to i+1}(x_j) - \sum_{l=1}^{m} \Psi_{i,i+1}(x_l, x_j)\mu_{i-1\to i}(x_l)\\
\vdots\\
\mu_{N\to N-1}(x_j) - \sum_{i=1}^{m} \Psi_{N,N-1}(x_i, x_j)\\
\vdots\\
\mu_{i+1\to i}(x_j) - \sum_{l=1}^{m} \Psi_{i+1,i}(x_l, x_j)\mu_{i+2\to i+1}(x_l)\\
\vdots
\end{array} \right\}. \tag{4.11}$$

Let the message variables be lexicographically ordered as

$$\begin{aligned}&\mu_{N-1\to N}(x_m) >_{lex} \cdots >_{lex} \mu_{N-1\to N}(x_1) >_{lex} \cdots >_{lex} \mu_{2\to 1}(x_j)\\ &>_{lex} \mu_{N\to N-1}(x_m) >_{lex} \cdots >_{lex} \mu_{1\to 2}(x_1).\end{aligned} \tag{4.12}$$

Given the above ordering of the message variables, it can be seen, that we can relabel each message variable and potential by assigning them a unique integer ID. By doing so, in the rest of this proof, we use the shorthand notation μ_i for representing a message variable $\mu_{j\to l}(k)$ and ψ_i to represent a potential $\Psi_{j,l}(\cdot,\cdot)$. In the ASP described in Eq. (4.11) we identify two different forms of polynomials, namely,

I. $\mu_i - \psi_i$, and

II. $\mu_i - \sum_{o=1}^{m} \psi_o\mu_o$.

It can be seen that $\mathrm{LT}(f_i) = \mu_i$ for every polynomial in $\mathbf{F}$, because of the chosen lex order of the message variables. We can compute now the GB with respect to the lex order using Buchberger's algorithm.

First we make $\mathbf{G} = \mathbf{F}$. We proceed to calculate the S-polynomials of all pairs of polynomials in $\mathbf{G}$. From the definition of S-polynomials, given two polynomials f and g, it follows that $S(f,g) = -S(g,f)$. Therefore, we have to compute only three different types of S-polynomials, namely $S(\mu_i - \psi_i, \mu_j - \psi_j)$, $S(\mu_i - \psi_i, \mu_j - \sum_l^m \psi_l\mu_l)$ and $S(\mu_i - \sum_l^m \psi_l\mu_l, \mu_j - \sum_n^m \psi_n\mu_n)$.

1. $S(\mu_i - \psi_i, \mu_j - \psi_j)$

   Both polynomials belong to the nodes in the extrema of the chain. Without loss of generality, we assume that $\mu_j >_{lex} \mu_i$. We notice that $\mathrm{l.c.m}(\mu_i, \mu_j) = \mu_i\mu_j$, and therefore, by Proposition 1, the remainder of the division of S(I,I) by $\mathbf{G}$ is zero. Nevertheless, as an example, we compute this case explicitly. Using Eq. (3.14), the S-polynomial is given by

$$S(\mu_i - \psi_i, \mu_j - \psi_j) = \psi_j\mu_i - \psi_i\mu_j \tag{4.13}$$

   Using the generalized division algorithm, the normal form of S(I,I) can be computed as

$$\begin{array}{r|lll} a_1: & \psi_j & & \\ a_2: & -\psi_l & & \\ \cline{2-3} \mu_i - \psi_i & \psi_j\mu_i - f_i\mu_j & -\psi_i\mu_j & \\ \mu_j - \psi_j & -\psi_j\mu_i & & +\psi_i\psi_j \\ \cline{2-4} \vdots & & -\psi_i\mu_j & +\psi_i\psi_j \\ & & +\psi_i\mu_j & +\psi_i\psi_j \\ \cline{3-4} & & 0 & \to \overline{S(\mu_i - \psi_i, \mu_j - \psi_j)}^{\mathbf{G}}. \end{array} \tag{4.14}$$

2. $S(\mu_i - \psi_i, \mu_j - \sum_l^m \psi_l\mu_l)$

   In this case we have that $\mathrm{l.c.m}(\mu_i, \mu_j) = \mu_i\mu_j$, and hence $\overline{S(\mu_i - \psi_i, \mu_j - \sum_l^m \psi_l\mu_l)}^{\mathbf{G}} = 0$.

3. $S(\mu_i - \sum_l^m \psi_l\mu_l, \mu_j - \sum_n^m \psi_n\mu_n)$

   Finally, in the last case we also have that $\mathrm{l.c.m}(\mu_i, \mu_j) = \mu_i\mu_j$. Therefore, it follows that $\overline{S(\mu_i - \psi_i, \mu_j - \sum_l^m \psi_l\mu_l)}^{\mathbf{G}} = 0$.

Since all of the reductions of the S-polynomials satisfy Buchberger's criterion (Theorem 2), no S-polynomial is added to the GB, hence $\mathbf{G} = \mathbf{F}$ is a (non reduced) GB. Since there is no constant polynomial c in $\mathbf{F}$, it follows that $c \notin \mathbf{G}$. Therefore, by Proposition 2, the reduced GB is not $\{1\}$. Furthermore, since all the leading terms are of the form $\mathrm{LT}(g_i) = \mu_i$ for $g_i \in \mathbf{G}$, the variety $\mathbf{V}(\mathbf{F})$ has only one point, and therefore it converges to a solution. □

It is possible to use the proposed convergence criteria to give an alternative proof to the convergence of the BP algorithm for graphs with a single loop. As already stated in Chapter 2, this is a well known result proved by Weiss in [3], using linear algebra techniques.

Theorem 8. *The BP algorithm converges to a solution for the MN $\mathcal{M}$ being a single loop.*

Proof. Consider the MN of a single loop with N nodes and m-ary variables shown in Figure 4.2. Similarly to the above theorem, we consider $\mathbf{F}$, the ASP of $\mathcal{M}$ be a set of polynomials that depend only on the message variables, i.e. the normalization constants are absorbed by the potentials $\Psi_{j,l}(\cdot,\cdot)$'s. This ASP can be written as

$$\mathbf{F} = \left\{ \begin{array}{c} \mu_{1\to 2}(x_1) - \sum_{l=1}^{m} \Psi_{1,2}(x_l, x_1)\mu_{N\to 1}(x_l) \\ \vdots \\ \mu_{1\to 2}(x_m) - \sum_{l=1}^{m} \Psi_{1,2}(x_l, x_m)\mu_{N\to 1}(x_l) \\ \vdots \\ \mu_{i\to i+1}(x_j) - \sum_{l=1}^{m} \Psi_{i,i+1}(x_l, x_j)\mu_{i-1\to i}(x_l) \\ \vdots \\ \mu_{i+1\to i}(x_j) - \sum_{l=1}^{m} \Psi_{i+1,i}(x_l, x_j)\mu_{i+2\to i+1}(x_l) \\ \vdots \end{array} \right\}. \tag{4.15}$$

Let the message variables be lexicographically ordered as in Eq. (4.12). Using the same shorthand notation for the message variables and factor potentials introduced in Theorem 7, we see that all polynomials in the above ASP are of the form $\mu_i - \sum_l^m \psi_l \mu_l$, but in this case μ_i is not always the leading term.

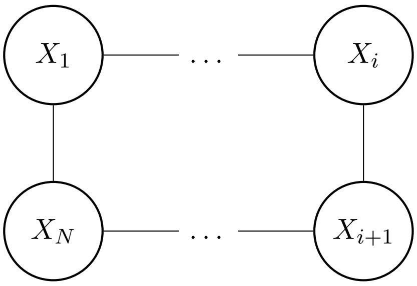


Figure 4.2: Graph of the MN of a single Loop with N variable nodes

We use Buchberger's algorithm to compute the GB with respect to the lexicographical order. First, we make $\mathbf{G} = \mathbf{F}$. Next, we have to calculate the S-polynomials of all pairs of polynomials in $\mathbf{G}$. However, we note that this is similar to the third case of S-polynomials in the proof of Theorem 7. Using the same argument, the least common multiple of the leading terms of each combination of polynomials is

$$\text{l.c.m}\left(\text{LT}\left(\mu_i - \sum_l^m \psi_l \mu_l\right), \text{LT}\left(\mu_j - \sum_p^m \psi_p \mu_p\right)\right) = \begin{cases} \mu_i \mu_j & \text{if } \mu_i, \mu_j \text{ are the LTs,} \\ \psi_w \mu_i \mu_w & \text{if } \mu_i, \psi_w \mu_w \text{ are the LTs,} \\ \psi_w \mu_j \mu_w & \text{if } \psi_w \mu_w, \mu_j \text{ are the LTs,} \\ \psi_w \psi_u \mu_u \mu_w & \text{if } \psi_u \mu_u, \psi_w \mu_w \text{ are the LTs,} \\ \psi_i \mu_i & \text{if } \psi_i \mu_i, \mu_i \text{ are the LTs.} \end{cases} \tag{4.16}$$

By Proposition 1, it follows that the remainder of the division of $S\left(\mu_i - \sum_l^m \psi_l \mu_l, \mu_j - \sum_p^m \psi_p \mu_p\right)$ by $\mathbf{G}$ is zero in all cases except if $\psi_i \mu_i, \mu_i$ are the leading terms of the polynomials. The S-polynomial for this case is given as

$$S\left(\mu_i - \sum_l^m \psi_l \mu_l, \mu_j - \sum_p^m \psi_p \mu_p\right) = \frac{\mu_j}{\psi_i} - \sum_l^m \psi_l \mu_l - \frac{1}{\psi_i} \sum_p \psi_p \mu_p, \tag{4.17}$$

which is not a constant polynomial. Adding this polynomials to $\mathbf{G}$, and noticing that no polynomial in $\mathbf{F}$ is a constant, by Proposition 2, the reduced GB with respect of the lexicographical order is not $\{1\}$. This implies that the system of equations defined by $\mathbf{F} = \mathbf{0}$ has a non-empy set of solutions. Furthermore, for every message variable μ_i there is a polynomial $g \in \mathbf{G}$ such that $\text{LT}(g_i) = \mu_i$. Therefore, by Theorem 6, it follows that the BP algorithm converges in this case. □

Another proof for this theorem using linear algebra and theory of determinants can be found in Appendix D, since solving systems of linear equations using GBs is equivalent to Gaussian elimination. In the following theorem, we show that using the proposed convergence criteria, the convergence of the BP algorithm for graphs with a tree-like structure can be proved:

Theorem 9. ***(Convergence for Trees)*** *The BP algorithm converges to a solution for the MN $\mathcal{M}$ having a tree-like structure.*

Proof. Let $\mathcal{M}$ be the MN of a tree with N nodes and m-ary variables shown in Figure 4.3. Similarly to the above theorems, we consider $\mathbf{F}$, the ASP of $\mathcal{M}$ be a set of polynomials that depend only on the message variables, i.e. the normalization constants are absorbed by the potentials $\Psi_{j,l}(\cdot,\cdot)$'s. The ASP of $\mathcal{M}$ can be written as

$$\mathbf{F} = \left\{ \begin{array}{c} \mu_{1\to I_1}(x_1) - \sum_{l=1}^m \Psi_{1,I_1}(x_l, x_1) \\ \vdots \\ \mu_{1\to I_1}(x_m) - \sum_{l=1}^m \Psi_{1,I_1}(x_l, x_m) \\ \vdots \\ \mu_{o\to o+1}(x_j) - \sum_{l=1}^m \Psi_{o,o+1}(x_l, x_j) \prod_{u\in \mathbf{Nb}(o)\setminus\{o+1\}} \mu_{u\to o}(l) \\ \vdots \end{array} \right\}. \tag{4.18}$$

Let the message variables be lexicographically ordered as

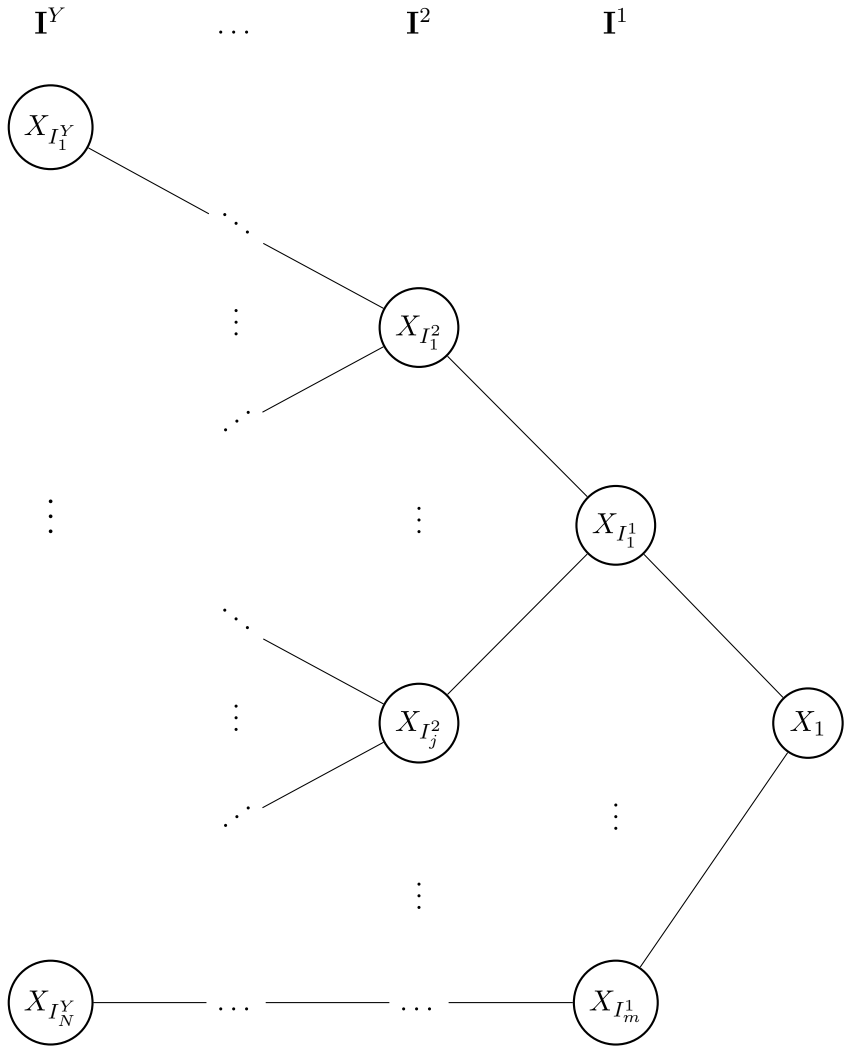


Figure 4.3: Graph of a tree with N nodes.

$$
\begin{aligned}
&\mu_{X_{I_z^{Y-1}} \to X_{I_y^{Y-2}}}(x_m) >_{lex} \cdots >_{lex} \mu_{X_{I_z^{Y-1}} \to X_{I_y^{Y-2}}}(x_1) >_{lex} \mu_{X_{I_z^{Y-1}} \to X_{I_N^Y}}(x_m) >_{lex} \cdots \\
&>_{lex} \mu_{X_{I_z^{Y-1}} \to X_{I_N^Y}}(x_1) >_{lex} \cdots >_{lex} \mu_{X_{I_1^1} \to X_1}(x_m) >_{lex} \cdots >_{lex} \mu_{X_{I_1^1} \to X_1}(x_1) \\
&>_{lex} \mu_{X_{I_1^1} \to X_{I_1^2}}(x_m) >_{lex} \cdots >_{lex} \mu_{X_{I_1^1} \to X_{I_1^2}}(x_1) >_{lex} \cdots >_{lex} \mu_{X_1 \to X_{I_1^1}}(x_m) \\
&>_{lex} \cdots >_{lex} \mu_{X_1 \to X_{I_1^1}}(x_1). \qquad (4.19)
\end{aligned}
$$

Using the same shorthand notation for the message variables and factor potentials introduced in Theorem 7, we see that there are two types of polynomials in the above ASP, namely

I. $\mu_i - \psi_i$, and

II. $\mu_j - \sum_{l=1}^{m} \psi_l \prod_p \mu_p$.

Let us compute the GB with respect to the lex order using Buchberger's algorithm. First, we make $\mathbf{G} = \mathbf{F}$. Since given two polynomials f and g it follows that $S(f, g) = -S(g, f)$, we have to consider only three different types of S-polynomials, namely $S(\mu_i - \psi_i, \mu_j - \psi_j)$,

$S(\mu_i - \psi_i, \mu_j - \sum_l^m \psi_l \prod_p \mu_p)$ and $S(\mu_i - \sum_l^m \psi_l \prod_p \mu_p, \mu_j - \sum_l^m \psi_l \prod_p \mu_p)$. We notice that, given the structure of the graph, the leading term of every polynomial $f_i \in \mathbf{F}$ is $\mathrm{LT}(f_i) = \mu_i$.

1. $S(\mu_i - \psi_i, \mu_j - \psi_j)$

   Both polynomials belong to the nodes in the extrema of the tree. Computing the least common multiple of the these two polynomials result in $\mathrm{l.c.m}(\mu_i, \mu_j) = \mu_i\mu_j$, and therefore, by Proposition 1, the remainder of the division of $S(\mu_i - \psi_i, \mu_j - \psi_j)$ by $\mathbf{G}$ is zero.

2. $S(\mu_i - \psi_i, \mu_j - \sum_l^m \psi_l \prod_p \mu_p)$

   The least common multiple of these two polynomials is $\mathrm{l.c.m}(\mu_i, \mu_j) = \mu_i\mu_j$, and hence, the normal form is $\overline{S(\mu_i - \psi_i, \mu_j - \sum_l^m \psi_l \prod_p \mu_p)}^{\mathbf{G}} = 0$.

3. $S(\mu_i - \sum_l^m \psi_l \prod_p \mu_p, \mu_j - \sum_l^m \psi_l \prod_q \mu_q)$

   Finally, in the last case we also have that the least common multiple of the two polynomials is $\mathrm{l.c.m}(\mu_i, \mu_j) = \mu_i\mu_j$. Therefore, the normal form of $S(\mu_i - \sum_l^m \psi_l \prod_p \mu_p, \mu_j - \sum_l^m \psi_l \prod_q \mu_q)$ vanishes, i.e. $\overline{S(\mu_i - \sum_l^m \psi_l \prod_p \mu_p, \mu_j - \sum_l^m \psi_l \prod_q \mu_q)}^{\mathbf{G}} = 0$.

Since no S-polynomial was added to $\mathbf{G}$, it follows that $\mathbf{G}$ is a non reduced GB with no constant polynomial. From Proposition 2, the reduced GB with respect to the lex order is not equal to $\{1\}$. Hence, the system $\mathbf{F} = \mathbf{0}$ has a solution. We notice that for every message variable μ_i there is a polynomial $g \in \mathbf{G}$ such that $\mathrm{LT}(g) = \mu_i$. By Theorem 6, this result ensures convergence for the BP algorithm for graphs with a tree-like structure. □

4.2 Tarkus Belief Propagation

We are interested in using the methods of commutative algebra, specially Gröbner bases, to compute the beliefs similar to the LBP algorithm. Such method is proposed in the following theorem:

Theorem 10. ***(Tarkus Belief Propagation)*** *Let $\mathcal{M}$, be a MN with variable nodes $\mathbf{X} = \{X_1, \ldots, X_N\}$. The beliefs $\{b_{X_j}(x_j) \mid X_j \in \mathbf{X}\}$ computed with Algorithm 4.1 are a solution to the MPEs described by the (L)BP algorithm.*

Proof. Let $\mathbf{F}$ be the ASP of $\mathcal{M}$. By construction, $\mathbf{G_F}$ is the reduced GB of the ideal generated by $\mathbf{F}$ with respect to the monomial ordering $>$. The function $ConvCrit(\mathbf{G_F})$ returns *true* if for every variable μ_i there is a polynomial g in $\mathbf{G_F}$ whose leading term is of the form $\mathrm{LT}(g) = \mu_i^{m_i}$. This follows directly from Theorem 6, and guarantees that the ideal generated by $\mathbf{F}$ is zero-dimensional, and thus, the BP converges to a solution.

In case $ConvCrit(\mathbf{G_F}) = true$, the GB $\mathbf{G_F}$ is transformed into a reduced GB with respect to the lex order. This can be done using the Gröbner Walk [34], or the FGLM [31] algorithms. The reason to do so, is because computing a basis with respect to the lex order is usually not very efficient [18]. Using $PolySolve(\mathbf{G_F})$ (Alg. 3.4) to compute the zeros of the polynomials in $\mathbf{G_F}$ is equivalent to computing the points in the variety $\mathbf{V}(\mathbf{F})$, the solutions to $\mathbf{F} = \mathbf{0}$, i.e. the solutions to the MPEs. □

The algorithm presented above is interesting, due to the fact that it uses methods coming from seemingly disjoint areas of mathematics (commutative algebra and probability theory) to perform inference in PGMs. Nevertheless, it should be noted that this algorithm is only suited for small problems. The main reason for this, is that the complexity of computing GBs for general ideals is known to be $O(k^{p(N)})$, with $p(N)$ a polynomial on the number of variables N

Algorithm 4.1: $TBP(\cdot)$ Tarkus Belief Propagation

```
input  : $\mathbf{F}_{\mathcal{M}}$, the associated set of polynomials of a MN $\mathcal{M}$
output: the set of beliefs $\{b_{X_i}(x_i) \mid X_i \in \mathbf{X}\}$
$\mathbf{G} := Groebner(\mathbf{F}_{\mathcal{M}}, >)$, the Gröbner basis of the ideal generated by $\mathbf{F}_{\mathcal{M}}$ respect to monomial ordering $>$
$\mathbf{G_F} := rGB(\mathbf{G})$, the reduced GB of the ideal generated by $\mathbf{F}_{\mathcal{M}}$
if $ConvCrit(G_{\mathbf{F}})$ is true then
    $\mathbf{G_F} := tGB(\mathbf{G_F}, >_{lex})$, where $tGB$ is an algorithm to transform $G_{\mathbf{F}}$ into a GB with respect to the lex ordering (like Gröbner Walk or FGLM)
    $\boldsymbol{\mu} = PolySolve!(\mathbf{G_F})$, the solutions of $\mathbf{F} = \mathbf{0}$
    for $X_i \in \mathbf{X}$ do
        $b_{X_i}(x_i) := \prod_{X_j \in \mathbf{Nb}(X_i)} \mu_{X_j \to X_i}(x_i)$
        return $\{b_{X_j}(x_j) \mid X_j \in \mathbf{X}\}$
else
    return UNCONVERGED
function($ConvCrit$)
input  : $\mathbf{G}$, a reduced GB for ideal generated by $\mathbf{F}$ with variables vars $= \{\mu_1, \ldots, \mu_D\}$
output: Boolean
ii := 0
for $\mu_i \in$ vars do
    if $\mathrm{LT}(g) = \mu_i^{m_i}$, for some $g \in \mathbf{G}$ and $m_i \in \mathbb{Z}_{\geq 0}$ then
        ii := ii +1
if $ii = D$ then
    return $true$
else
    return $false$
```

and k a constant [35], while the worst case complexity[3] of computing the true marginals for systems with m-ary RVs is $O(m^N)$ [5]. This means, that using TBP to compute the marginals for a general graph could be a more complex problem than calculating the actual marginal probabilities. Nevertheless, it should be noted, that several more optimistic complexity bounds for computing GBs of ideals with a particular structure have been derived, and newer and more efficient methods for computing GBs are being constantly developed [30, 35].

[3] It has been shown that using the junction trees, the complexity of performing exact inference on an undirected graphical model is exponential with the size of the largest clique [5].

5 Experiments

In this chapter, the we present an experiment to show the performance of the TBP proposed in Chapter 4. In order to do so, we compute the beliefs of a 2 × 2 spin glass using TBP and LBP. This model is simple enough to analytically compute the true marginal probabilities [17].

The rest of this chapter is structured as follows: In Section 5.1 the experimental setup for computing the beliefs using both TBP and LBP is described, and in Section 5.2, a brief overview of spin glasses using the Boltzmann distribution and the Ising model is provided. Section 5.3 compares the results of the beliefs computed with TBP and LBP and the true marginal probabilities for a binary spin glass. We conclude this chapter in Section 5.4, with the discussion of these results.

5.1 Experimental Setup

In order to compute the beliefs, the TBP from Algorithm 4.1 was implemented in MAPLE 18. The build-in F_4 method by Faugère [30] was selected to compute the GBs. This algorithm is known for its efficiency in computing GBs for zero-dimensional ideals with coefficients in $\mathbb{Q}$, the set of rational numbers [36, 37]. The code for this implementation can be found on Appendix A. The standard LBP from Algorithm 2.1 was implemented in MATLAB R2013a.

A 2 × 2 spin glass $\mathcal{M}_{2\times2SG}$, as shown in Figure 5.1, with binary states $x = -1$ and $\bar{x} = 1$, is used for this experiments. The joint probability distribution for this MN is given by the Ising model, described in the following section. The beliefs computed with TPB and LBP are then compared with the analytical closed form solution of the true marginals.

5.2 Spin Glass

Although the theory behind spin glasses model originated in the context of statistical physics, its scope has been expanding far beyond its original goal of explaining actual spin glass materials [38]. This framework is an example of a system that has an extremely complex structure but, nevertheless can be subject of rigorous systematic analyses, and its analytical treatment has been recognized as an important tool in the study of information processing tasks, error correcting codes, image restoration and optimization problems, among others [38]. A *spin glass* is an example of a *disordered system*, where the term “glass” comes from the analogy of the system with the positional disorder of atoms in a conventional glass [38]. Classically, it refers to a disordered magnet, in which the spins are stochastically positioned[4].

The *Boltzmann distribution*, for particles (represented by binary random variables $X_1, \dots, X_N$) with energy $E(X_1, \dots, X_N)$ is given by

$$p(X_1, \dots, X_N) = \frac{1}{Z} \exp\left(-E(X_1, \dots, X_N)\right), \tag{5.1}$$

where Z is a normalization constant. From classical analytical mechanics [39, 40], the total energy of a system can be represented by its *Hamiltonian function* H, which can be understood as is the sum of kinetic and potential energy. Spin glasses belong to the class of *quenched disordered systems*, in which the disorder in the system is explicitly present in the Hamiltonian, under the form of random couplings J, θ, i.e. $H = H(J, \theta, x_1, \dots, x_N)$, with x_i representing the state of X_i.

The Hamiltonian of a 2D spin glass with pairwise interactions in an external field using the Ising model [15, 17] is given by

$$H(J, \theta, x_i, \dots, x_N) = -\sum_{j>i}^{N} J_{ij} x_i x_j - \sum_{i}^{N} \theta_i x_i, \tag{5.2}$$

where J_{ij} are the couplings of the interactions X_i and X_j and θ_i represents the interaction of an external field with X_i. Substituting Eq. (5.2) in the Boltzmann distribution from Eq. (5.1), the joint probability distribution results in

$$p(X_1, \dots, X_N) = \frac{1}{Z} \exp\left(\sum_{j>i}^{N} J_{ij} x_i x_j + \sum_{i}^{N} \theta_i x_i\right). \tag{5.3}$$

This joint distribution can be represented using the formalisms of a MN $\mathcal{M}$, where the every particle X_i represents a variable node. Comparing Eq. (5.3) with Eq. (2.16), we can write the potentials as

$$\Psi_{X_i, X_j}(x_i, x_j) = \phi_{i,j}(x_i, x_j)\phi_j(x_j), \tag{5.4}$$

where the factors $\phi_{i,j}(x_i, x_j)$ and $\phi_j(x_j)$ are given by

$$\phi_{i,j}(x_i, x_j) = \exp(J_{ij} x_i x_j) \qquad \text{and} \qquad \phi_j(x_j) = \exp(\theta_j x_j). \tag{5.5}$$

5.3 2×2 Spin Glass

In Figure 5.1 the passing of the messages between the variable nodes is explicitly shown. We assume that the couplings and the external field are uniform, i.e. $J_{ij} = J$ if there is an edge

[4] Spin is an intrinsic form of angular momentum carried by elementary particles and atomic nuclei. Associated with this angular momentum is a magnetic moment [39].

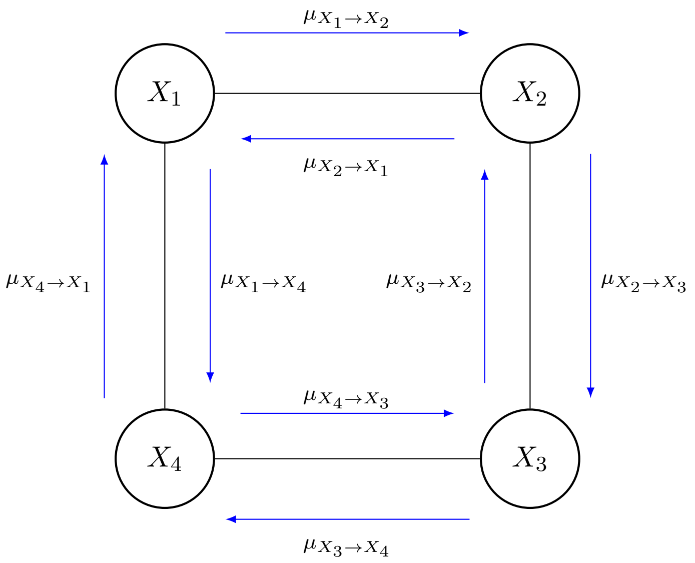


Figure 5.1: Message passing on a 2 × 2 Spin Glass.

between X_i and X_j and $\theta_i = \theta$ [17]. The ASP for this spin glass is

$$\mathbf{F}_{2\times 2SG} = \left\{ \begin{array}{l} \mu_{1\to 2}(x) - Z_{1\to 2}\left(\Psi_{X_1,X_2}(x,x)\mu_{4\to 1}(x) + \Psi_{X_1,X_2}(\bar{x},x)\mu_{4\to 1}(\bar{x})\right) \\ \mu_{1\to 2}(\bar{x}) - Z_{1\to 2}\left(\Psi_{X_1,X_2}(x,\bar{x})\mu_{4\to 1}(x) + \Psi_{X_1,X_2}(\bar{x},\bar{x})\mu_{4\to 1}(\bar{x})\right) \\ \mu_{1\to 4}(x) - Z_{1\to 4}\left(\Psi_{X_1,X_4}(x,x)\mu_{2\to 1}(x) + \Psi_{X_1,X_4}(\bar{x},x)\mu_{2\to 1}(\bar{x})\right) \\ \mu_{1\to 4}(\bar{x}) - Z_{1\to 4}\left(\Psi_{X_1,X_4}(x,\bar{x})\mu_{2\to 1}(x) + \Psi_{X_1,X_4}(\bar{x},\bar{x})\mu_{2\to 1}(\bar{x})\right) \\ \mu_{2\to 1}(x) - Z_{2\to 1}\left(\Psi_{X_1,X_2}(x,x)\mu_{3\to 2}(x) + \Psi_{X_1,X_2}(x,\bar{x})\mu_{3\to 2}(\bar{x})\right) \\ \mu_{2\to 1}(\bar{x}) - Z_{2\to 1}\left(\Psi_{X_1,X_2}(\bar{x},x)\mu_{3\to 2}(x) + \Psi_{X_1,X_2}(\bar{x},\bar{x})\mu_{3\to 2}(\bar{x})\right) \\ \mu_{2\to 3}(x) - Z_{2\to 3}\left(\Psi_{X_2,X_3}(x,x)\mu_{1\to 2}(x) + \Psi_{X_2,X_3}(\bar{x},x)\mu_{1\to 2}(\bar{x})\right) \\ \mu_{2\to 3}(\bar{x}) - Z_{2\to 3}\left(\Psi_{X_2,X_3}(x,\bar{x})\mu_{1\to 2}(x) + \Psi_{X_2,X_3}(\bar{x},\bar{x})\mu_{1\to 2}(\bar{x})\right) \\ \mu_{3\to 2}(x) - Z_{3\to 2}\left(\Psi_{X_2,X_3}(x,x)\mu_{4\to 3}(x) + \Psi_{X_2,X_3}(x,\bar{x})\mu_{4\to 3}(\bar{x})\right) \\ \mu_{3\to 2}(\bar{x}) - Z_{3\to 2}\left(\Psi_{X_2,X_3}(\bar{x},x)\mu_{4\to 3}(x) + \Psi_{X_2,X_3}(\bar{x},\bar{x})\mu_{4\to 3}(\bar{x})\right) \\ \mu_{3\to 4}(x) - Z_{3\to 4}\left(\Psi_{X_3,X_4}(x,x)\mu_{2\to 3}(x) + \Psi_{X_3,X_4}(\bar{x},x)\mu_{2\to 3}(\bar{x})\right) \\ \mu_{3\to 4}(\bar{x}) - Z_{3\to 4}\left(\Psi_{X_3,X_4}(x,\bar{x})\mu_{2\to 3}(x) + \Psi_{X_3,X_4}(\bar{x},\bar{x})\mu_{2\to 3}(\bar{x})\right) \\ \mu_{4\to 1}(x) - Z_{4\to 1}\left(\Psi_{X_1,X_4}(x,x)\mu_{3\to 4}(x) + \Psi_{X_1,X_4}(x,\bar{x})\mu_{3\to 4}(\bar{x})\right) \\ \mu_{4\to 1}(\bar{x}) - Z_{4\to 1}\left(\Psi_{X_1,X_4}(\bar{x},x)\mu_{3\to 4}(x) + \Psi_{X_1,X_4}(\bar{x},\bar{x})\mu_{3\to 4}(\bar{x})\right) \\ \mu_{4\to 3}(x) - Z_{4\to 3}\left(\Psi_{X_3,X_4}(x,x)\mu_{1\to 4}(x) + \Psi_{X_3,X_4}(x,\bar{x})\mu_{1\to 4}(\bar{x})\right) \\ \mu_{4\to 3}(\bar{x}) - Z_{4\to 3}\left(\Psi_{X_3,X_4}(\bar{x},x)\mu_{1\to 4}(x) + \Psi_{X_3,X_4}(\bar{x},\bar{x})\mu_{1\to 4}(\bar{x})\right) \\ \mu_{1\to 2}(x) + \mu_{1\to 2}(\bar{x}) - 1 \\ \mu_{1\to 4}(x) + \mu_{1\to 4}(\bar{x}) - 1 \\ \mu_{2\to 1}(x) + \mu_{2\to 1}(\bar{x}) - 1 \\ \mu_{2\to 3}(x) + \mu_{2\to 3}(\bar{x}) - 1 \\ \mu_{3\to 2}(x) + \mu_{3\to 2}(\bar{x}) - 1 \\ \mu_{3\to 4}(x) + \mu_{3\to 4}(\bar{x}) - 1 \\ \mu_{4\to 1}(x) + \mu_{4\to 1}(\bar{x}) - 1 \\ \mu_{4\to 3}(x) + \mu_{4\to 3}(\bar{x}) - 1 \end{array} \right\}, \tag{5.6}$$

where the potentials $\Psi_{X_i,X_j}(x_i, x_j)$ are given by Eq. (5.4).

Let the message and normalization constant variables ve lexicographically ordered as

$$\begin{aligned}
&Z_{4\to1}Z_{4\to3} >_{lex} Z_{3\to4} >_{lex} Z_{3\to2} >_{lex} Z_{2\to3} >_{lex} Z_{2\to1}\\
&>_{lex} Z_{1\to4} >_{lex} Z_{1\to2}\mu_{4\to3}(\bar{x}) >_{lex} \mu_{4\to3}(x)\mu_{4\to1}(\bar{x})\\
&>_{lex} \mu_{4\to1}(x) >_{lex} \mu_{3\to4}(\bar{x}) >_{lex} \mu_{3\to4}(x) >_{lex} \mu_{3\to2}(\bar{x})\\
&>_{lex} \mu_{3\to2}(x) >_{lex} \mu_{2\to3}(\bar{x})\mu_{2\to3}(x) >_{lex} \mu_{2\to1}(\bar{x})\\
&>_{lex} \mu_{2\to1}(x) >_{lex} \mu_{1\to4}(\bar{x}) >_{lex} \mu_{1\to4}(x) >_{lex} \mu_{1\to2}(\bar{x}) >_{lex} \mu_{1\to2}(x).
\end{aligned} \tag{5.7}$$

As an example of the TBP algorithm, we show the explicit computation of the belief $b_{X_1}(x)$, i.e. the belief that the RV X_1 is in state x, for $J = \theta = 0$. The GB with respect to the lex order of the ideal generated by $\mathbf{F}_{2\times2SG}$ for this particular choice of the couplings J and θ is given by

$$\mathbf{G_F} = \left\{ \begin{array}{c}
\mu_{1\to2}(x) - \frac{1}{2}\\
\mu_{1\to2}(\bar{x}) - \frac{1}{2}\\
\mu_{1\to4}(x) - \frac{1}{2}\\
\mu_{1\to4}(\bar{x}) - \frac{1}{2}\\
\mu_{2\to1}(x) - \frac{1}{2}\\
\mu_{2\to1}(\bar{x}) - \frac{1}{2}\\
\mu_{2\to3}(x) - \frac{1}{2}\\
\mu_{2\to3}(\bar{x}) - \frac{1}{2}\\
\mu_{3\to2}(x) - \frac{1}{2}\\
\mu_{3\to2}(\bar{x}) - \frac{1}{2}\\
\mu_{3\to4}(x) - \frac{1}{2}\\
\mu_{3\to4}(\bar{x}) - \frac{1}{2}\\
\mu_{4\to1}(x) - \frac{1}{2}\\
\mu_{4\to1}(\bar{x}) - \frac{1}{2}\\
\mu_{4\to3}(x) - \frac{1}{2}\\
\mu_{4\to3}(\bar{x}) - \frac{1}{2}\\
Z_{1\to2} - \frac{1}{2}\\
Z_{2\to1} - \frac{1}{2}\\
Z_{3\to2} - \frac{1}{2}\\
Z_{2\to3} - \frac{1}{2}\\
Z_{3\to4} - \frac{1}{2}\\
Z_{4\to3} - \frac{1}{2}\\
Z_{1\to4} - \frac{1}{2}\\
Z_{4\to1} - \frac{1}{2}
\end{array} \right\}. \tag{5.8}$$

Using Eq. (2.24), the $b_{X_1}(x)$ is computed as

$$\begin{aligned}
b_{X_1}(x) &= \frac{1}{Z} \prod_{X_i \in \mathbf{Nb}(X_1)} \mu_{X_i\to X_1}(x)\\
&= \frac{1}{Z}\mu_{X_4\to X_1}(x)\mu_{X_2\to X_1}(x)\\
&= \frac{1}{Z} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{4Z}.
\end{aligned} \tag{5.9}$$

Similarly, the belief for X_1 being in state $\bar{x}$ is given by

$$\begin{aligned}
b_{X_1}(\bar{x}) &= \frac{1}{Z} \prod_{X_i \in \mathbf{Nb}(X_1)} \mu_{X_i \to X_1}(\bar{x}) \\
&= \frac{1}{Z} \mu_{X_4 \to X_1}(\bar{x}) \mu_{X_2 \to X_1}(\bar{x}) \\
&= \frac{1}{Z} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{4Z}. \qquad (5.10)
\end{aligned}$$

Since $b_{X_1}(x) + b_{X_1}(\bar{x}) = 1$, the normalization constant is $Z = \frac{1}{2}$. Substituting Z in $b_{X_1}(x)$ results in

$$b_{X_1}(x) = \frac{1}{4 \times \frac{1}{2}} = \frac{1}{2}. \qquad (5.11)$$

Given the simplicity of this spin glass, the true marginal probability RV X_1 in state x can be analytically computed using a naïve approach. Substituting Eq. (5.3) in Eq. (2.6), and performing all the summations, this marginal probability is given by

$$\begin{aligned}
p(X_1) &= \sum_{x_2} \sum_{x_3} \sum_{x_4} p_{2 \times 2SG}(X_1, X_2, X_3, X_4) \\
&= \frac{e^{2(-Jx+J+\theta x+\theta)} \left(2e^{2(Jx+J+2\theta)} + e^{4(Jx+J+3\theta)} + 2e^{2(Jx+J+4\theta)} + e^{4Jx+8\theta} + e^{4J} + e^{4\theta} \right)}{4e^{4(J+\theta)} + e^{8(J+2\theta)} + 4e^{4(J+3\theta)} + 4e^{4J+8\theta} + e^{8J} + 2e^{8\theta}}. \qquad (5.12)
\end{aligned}$$

As stated in Chapter 2, this approach to perform exact inference becomes infeasible for a large number of variables. More efficient alternatives for performing exact inference are transforming the graph into a Junction tree and the Variable elimination algorithm for triangulation of graphs [11].

For this experiment, we computed the belief of RV X_1 being in state x using TBP and LBP, with the coupling with the external field, θ, ranging from -3 to 3 with a step-size of 0.2 and the coupling of the pairwise interactions between RVs, J, ranging from -2 to 2 with a step-size of 0.2. These beliefs are shown in the the top row of Figure 5.2. On the bottom row of this figure, the absolute error between the computed beliefs and the true marginal probabilities is shown, as well as the absolute difference between the beliefs computed with TBP and the beliefs computed with LBP. In Figure 5.3, the number of solutions of the TBP algorithm for couplings J and θ in the above mentioned ranges is shown.

5.4 Discussion

In Figure 5.2, it can be seen that for positive couplings, both TBP and LBP converge to the true marginals. This is an expected result, since both TBP and LBP are not necessarily supposed to converge to the true marginals, but usually offer a relatively good approximation. The mean squared error (MSE) for the TBP compared with the true marginals is 0.16, while the MSE for the LBP compared with the true marginals is only 0.03. However, the MSE for the TBP compared with the LBP is only 0.06.

We can see in Figure 5.3, that there are 4 solutions to the MPE in the range $J \in [-2, 2]$ and $\theta \in [-3, 3]$, exempt for the line at $\theta = 0$, where there is only one solution. This shows the existence of a finite number of solutions, as proved in Corollary 8.

As commented previously, one of the main issues with the TBP algorithm is that it has the com-

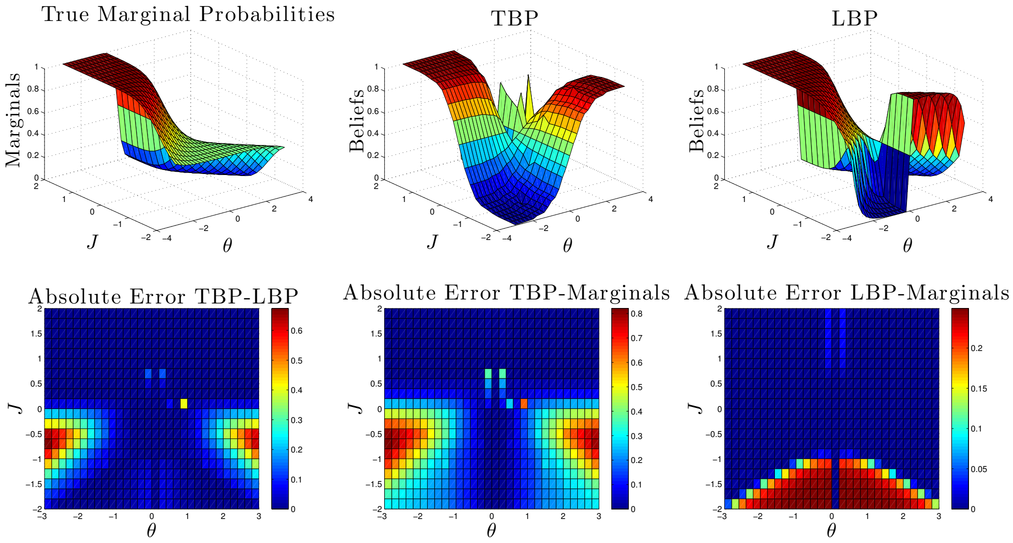


Figure 5.2: Comparison of Beliefs calculated by TBP, LBP and the true marginals for the 2×2 spin glass.

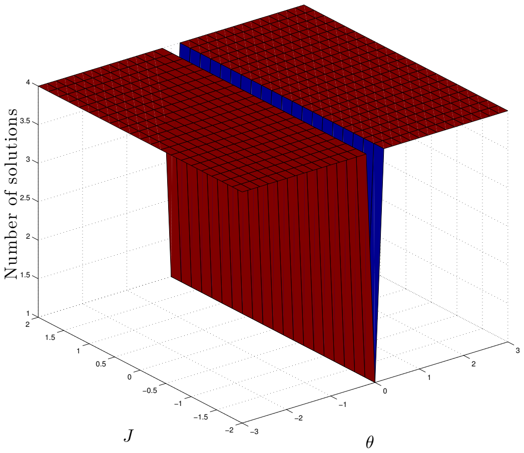


Figure 5.3: Number of solutions of the TBP algorithm for the 2×2 spin glass.

putational limitations of the algorithms to calculate GBs. In particular, most of the algorithms in MAPLE only work with rational numbers. In order to avoid this kind of errors, specially since the potentials are defined in terms of transcendental functions, a rounded version of the potential factors was introduced as

$$\phi_{i,j}(x_i, x_j) = \frac{\text{round}\left(\gamma \exp(J_{ij} x_i x_j)\right)}{\gamma} \qquad \text{and} \qquad \phi_j(x_j) = \frac{\text{round}\left(\gamma \exp(\theta_j x_j)\right)}{\gamma}, \tag{5.13}$$

where γ is a parameter that controls the precision. This introduces a round-off error in the coefficients of the polynomials in the ASP, which means, that we are actually finding the solutions for a slightly different system of equations than the original.

6 Conclusions

In this thesis, the task of inference using probabilistic graphical models was reviewed using the framework of computational commutative algebra. In particular, we used the concepts of ideals, varieties and Gröbner bases, to determine a convergence criterion for Belief Propagation, and to formulate a new algorithm for computing the marginals probabilities of distributions given by a Markov network.

It should be remarked that convergence of the BP algorithm in the case of Markov chains, Trees and Single Loops are well known results in the literature [3, 5, 11]. Nevertheless, it is interesting to see that these known results can also be explained using the framework of commutative algebra.

However, as noted in Section 4.2, the proposed TBP can only be used for solving small toy-problems. This is due to the fact that Gröbner bases are in general very expensive to compute [35], and they heavily depend on a good selection of the monomial ordering. Another more technical-related problem is that (almost) all commercial computer algebra packages have only support for polynomial ideals with coefficients in $\mathbb{Q}$ [37], instead of floating point numbers.

6.1 Conjectures and Future Work

The preliminary results obtained during the realization of this thesis motivate the following conjectures and propositions for future work:

- The problem of finding a good monomial ordering for TBP (including a good lexicographical order of the variables of the polynomials in the ASP) is equivalent to the problem of finding a good random variable ordering in the Variable Elimination (VE) algorithm for triangulation of graphs [11].

  The VE algorithm represents an alternative to MPAs for performing exact inference in PGMs. VE uses the factorization properties induced by the conditional independence of the random in the joint probability distribution defined by a PGM to iteratively *eliminate* random variables from the joint distribution, and therefore, marginalize it. It can be seen that the process of elimination could be also explained with the semantics of elimination theory. This could lead to the use of criteria and heuristic methods from VE to find optimal orderings of the message variables in the set of polynomials associated with a graphical model (and vise versa).

- Relationship between algebraic independence of the polynomials of ASP $\mathbf{F}$ and statistical independence of the variables in $\mathcal{M}$.

  It is said that elements $\phi_1, \ldots, \phi_r \in \mathbb{K}[\mathbf{V}]$ are *algebraically independent* over $\mathbb{K}$, the collection of all polynomial functions $\phi \colon \mathbf{V} \mapsto \mathbb{K}$, if there is no nonzero polynomial $p(\phi_1, \ldots, \phi_r) \in \mathbb{K}[\mathbf{V}]$ with coefficients in $\mathbb{K}$ that vanishes, i.e. $p(\phi_1, \ldots, \phi_r) = 0$ [18]. It can be (non-trivially) proved, that the maximal number of algebraically independent elements in the coordinate ring $\mathbb{K}[\mathbf{V}]$ equals the dimension of the variety $\mathbf{V} \subset \mathbb{K}^n$ (see [18, pp. 478]). Using the concept of coordinate ring (see Appendix C) it can be seen that the messages computed as in Eq. 2.32 are coordinate functions of the variety $\mathbf{V}(\mathbf{F})$. This notion is related to the dimension of $\mathbf{V}(\mathbf{F})$, and hence, the notion of algebraic independence in $\mathbb{K}[\mathbf{V}]$.

  It would be interesting to explore the relationship of statistical independence of the random variables in $\mathcal{M}$ and the algebraic independence of the polynomials in $\mathbf{F}$, and to derive conditions that relate the graph structure to the dimension of the variety of the ASP generated by such graph.

- Border Basis-based Tarkus Belief Propagation.

  As pointed out throughout Chapter 3 and Chapter 4, the computation of Gröbner Bases relies on both a good selection of the monomial ordering and the field in which the coefficients of the polynomials lie. An interesting alternative to GBs are Border Bases (BB). BBs use the concept of order ideal $\mathcal{O}$, which is a subset of the set of monomials in a polynomial ring that is closed under divisors, i.e. if $f \in \mathcal{O}$ and the residual of the division of f by g is zero imply that $g \in \mathcal{O}$ [41]. It has been shown that given a monomial ordering $>$, the elements of a reduced GB with respect to that ordering are exactly the border basis polynomials corresponding to the minimal generators of the border term ideal [41, 42].

  BBs have some advantages over GBs, since they are independent of the monomial ordering, and they are numerically more stable than GBs. BB are more robust agains slight changes in the coefficients of some polynomials generating an ideal $\mathbf{I}$. [41, 43]. These properties would address directly several issues of the GB-based TBP discussed in Section 5.4.

Bibliography

[1] E. Sudderth and W. T. Freeman, "Signal and Image Processing with Belief Propagation [DSP Applications]," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 114–141, Feb. 2008.

[2] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[3] Y. Weiss, "Belief propagation and revision in networks with loops," 1997.

[4] A. L. A. Yuille, "CCCP algorithms to minimize the Bethe and Kikuchi free energies: convergent alternatives to belief propagation." *Neural Computation*, vol. 14, no. 7, pp. 1691–1722, Jun. 2002.

[5] F. Pernkopf, R. Peharz, and S. Tschiatschek, "Introduction to Probabilistic Graphical Models," Graz University of Technology, 2012.

[6] T. Hazan and A. Shashua, "Convergent message-passing algorithms for inference over general graphs with convex free energies," *arXiv preprint arXiv:1206.3262*, 2012.

[7] J. Pearl, "Reverend Bayes on Inference Engines," 1982.

[8] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[9] S. J. Rennie, J. R. Hershey, and P. A. Olsen, "Hierarchical variational loopy belief propagation for multi-talker speech recognition," *Audio, Transactions of the IRE Professional Group on*, pp. 176–181, Dec. 2008.

[10] W. T. Freeman and E. C. Pasztor, "Learning low-level vision," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on.* IEEE, 1999, pp. 1182–1189.

[11] D. Koller and N. Friedman, *Probabilistic graphical models*, ser. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2009.

[12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Pub, 1988.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and B. Scholkopf, Eds. Microsoft Research Ltd.: Springer Verlag, 2009.

[14] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Bethe free energy, Kikuchi approximations, and belief propagation algorithms," Mitshubishi Electric Laboratories, Inc, Tech. Rep., May 2001.

[15] F. Schwabl, *Statistical Mechanics*, 2nd ed. Springer, Sep. 2006.

[16] A. T. Ihler, J. W. Fisher, and A. S. Willsky, “Loopy belief propagation: Convergence and effects of message errors,” *Journal of Machine Learning Research (JMLR)*, vol. 6, no. 1, p. 905, 2006.

[17] J. M. Mooij and H. J. Kappen, “Sufficient conditions for convergence of theSum-Product Algorithm,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4422–4437, Dec. 2007.

[18] D. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, 3rd ed. New York: Springer, 2007.

[19] B. Buchberger, “Bruno Buchberger’s PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal,” *Journal of Symbolic Computation*, vol. 41, no. 3-4, pp. 475–511, Mar. 2006.

[20] ——, “Gröbner Bases:A Short Introduction for Systems Theorists,” pp. 1–19, Sep. 2001.

[21] Z. Lin, L. Xu, and N. K. Bose, “A tutorial on Gröbner bases with applications in signals and systems,” *Circuits and Systems I: Regular Papers*, 2008.

[22] S. Tschiatschek, C. E. Cancino Chacon, and F. Pernkopf, “Bounds for Bayesian Network Classifiers with Reduced Precision Parameters,” *2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1–5, Nov. 2012.

[23] S. Russell and P. Norvig, *Künstliche Intelligenz*, 3rd ed., ser. Ein moderner Ansatz. Munich: Pearson Deutschland GmbH, 2012.

[24] T. Hazan and A. Shashua, “Norm-Product Belief Propagation: Primal-Dual Message-Passing for Approximate Inference,” *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6294–6316.

[25] F. E. Miranda and E. Viso, “Matematicas discretas,” *México: UNAM Facultad de Ciencias*, 2010.

[26] M. Khosla, “Message Passing Algorithms,” Ph.D. dissertation, Universität des Saarlandes., Max Planck Institut für Informatik, 2009.

[27] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Constructing free-energy approximations and generalized belief propagation algorithms,” *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2282–2312, 2005.

[28] I. Ajwa, Z. Liu, and P. Wang, “Gröbner Bases Algorithm,” Kent State University, Tech. Rep., Feb. 2003.

[29] P. D. David A Cox, J. B. Little, and D. B. O’Shea, *Graduate Texts in Mathematics Volume 185 : Using Algebraic Geometry.* Springer Verlag, 1998.

[30] J. C. Faugere, “A new efficient algorithm for computing Gröbner bases (F 4),” *Journal of pure and applied algebra*, vol. 139, no. 1-3, pp. 61–88, 1999.

[31] P. Benge, V. Burks, and N. Cobar, “Groebner Basis Conversion Using the FGLM Algorithm,” *math.lsu.edu.*

[32] R. L. Burden and J. D. Faires, *Análisis Numérico*, 7th ed. Youngstown State University: Thomson Learning, 2002.

[33] S. Collart, M. Kalkbrener, and D. Mall, “Converting bases with the Gröbner walk,” *Journal of Symbolic Computation*, vol. 24, no. 3-4, pp. 465–469, 1997.

[34] J. Buchmann, A. Pyshkin, and R.-P. Weinmann, “A Zero-Dimensional Gröbner Basis for AES-128,” *Fast Software Encryption*, vol. 4047, no. Chapter 6, pp. 78–88, 2006.

[35] E. W. Mayr, “Some Complexity Results for Polynomial Ideals,” *Journal of Complexity*, vol. 13, no. 3, pp. 303–325, Dec. 1996.

[36] A. Kondratyev, H. J. Stetter, and S. Winkler, “Numerical computation of Gröbner bases,” *. . . Algebra in Scientific Computing)*, 2004.

[37] J.-C. Faugère, *Mathematical Software – ICMS 2010*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6327, ch. FGb: A Library for Computing Gröbner Bases, pp. 84–87.

[38] H. Nishimori, *Statistical physics of spin glasses and information processing: an introduction.* Oxford University Press, 2001, vol. 111.

[39] A. Beiser, *Concepts Of Modern Physics*, 5th ed. New York: McGraw-Hill, Inc, 1995.

[40] H. Goldstein and C. Poole, *Classical Mechanics*, 3rd ed. Addison-Wesley, Sep. 2002.

[41] A. Kehrein and M. Kreuzer, “Computing border bases,” *Journal of pure and applied algebra*, vol. 205, no. 2, pp. 279–295, Dec. 2005.

[42] D. Heldt, M. Kreuzer, S. Pokutta, and H. Poulisse, “Approximate computation of zero-dimensional polynomial ideals,” *. . . of Symbolic Computation*, vol. 44, no. 11, pp. 1566–1591, 2009.

[43] M.-L. Torrente, “Applications of Algebra in the Oil Industry,” Ph.D. dissertation, Scuola Normale Superiore, May 2009.

[44] M. R. Spiegel and J. M. Liu, *Mathematical Handbook of Formulas and Tables*, ser. Schaum’s outlines. McGraw-Hill Companies, 1999.

[45] H. Rincón, *Álgebra lineal*, 2nd ed., ser. las prensas de ciencias. Universidad Nacional Autónoma de México, 2006.

A

Maple Code of the TBP for the 2×2 spin glass

```
# Convergence Criterion
ConvCrit := proc (Eq, Vars)
global V, IsZD, NSols;
V := Groebner[SuggestVariableOrder](Eq, Vars);
IsZD := Groebner[IsZeroDimensional](Eq, plex(V));
if IsZD = true then
NSols := PolynomialIdeals[NumberOfSolutions]
(PolynomialIdeals[PolynomialIdeal](Eq, variables = {V}))
else
 NSols := infinity
 end if;
 if NSols <> 0 then true else false
 end if
 end proc;
 # TBP for 2 by 2 spin glass
 TBP2x2 := proc (J, th)
 global x1, x2, Dig, F, f, F12, F14,
 F23, F34, f1, f2, f3, f4, EqsNorm, vars, Marg,
 MargVars, MargNVars, GB, mssgs, mrgs, nsols;

 # Potentials
 x1 := 1;
 x2 := 2;
 Dig := 10;
 F := proc (X, Y) options operator, arrow;
 round(evalf(exp(J*X*Y))*Dig)/Dig
 end proc;
 f := proc (X) options operator, arrow;
 round(evalf(exp(th*X))*Dig)/Dig
 end proc;
 F12 := Matrix(2, 2, [F(-1, -1), F(1, -1), F(-1, 1), F(1, 1)]);
```

```
F14 := Matrix(2, 2, [F(-1, -1), F(1, -1), F(-1, 1), F(1, 1)]);
F23 := Matrix(2, 2, [F(-1, -1), F(1, -1), F(-1, 1), F(1, 1)]);
F34 := Matrix(2, 2, [F(-1, -1), F(1, -1), F(-1, 1), F(1, 1)]);
f1 := Matrix(2, 1, [f(-1), f(1)]);
f2 := Matrix(2, 1, [f(-1), f(1)]);
f3 := Matrix(2, 1, [f(-1), f(1)]);
f4 := Matrix(2, 1, [f(-1), f(1)]);

#Associated set of polynomials
EqsNorm := [-a1*m12x1+f1(x1)*F12(x1, x1)*m41x1+f1(x2)*F12(x2, x1)*m41x2,
-a1*m12x2+f1(x1)*F12(x1, x2)*m41x1+f1(x2)*F12(x2, x2)*m41x2,
-a2*m14x1+f1(x1)*F14(x1, x1)*m21x1+f1(x2)*F14(x2, x1)*m21x2,
-a2*m14x2+f1(x1)*F14(x1, x2)*m21x1+f1(x2)*F14(x2, x2)*m21x2,
-a3*m21x1+f2(x1)*F12(x1, x1)*m32x1+f2(x2)*F12(x1, x2)*m32x2,
-a3*m21x2+f2(x1)*F12(x2, x1)*m32x1+f2(x2)*F12(x2, x2)*m32x2,
-a4*m23x1+f2(x1)*F23(x1, x1)*m12x1+f2(x2)*F23(x2, x1)*m12x2,
-a4*m23x2+f2(x1)*F23(x1, x2)*m12x1+f2(x2)*F23(x2, x2)*m12x2,
-a5*m32x1+f3(x1)*F23(x1, x1)*m43x1+f3(x2)*F23(x1, x2)*m43x2,
-a5*m32x2+f3(x1)*F23(x2, x1)*m43x1+f3(x2)*F23(x2, x2)*m43x2,
-a6*m34x1+f3(x1)*F34(x1, x1)*m23x1+f3(x2)*F34(x2, x1)*m23x2,
-a6*m34x2+f3(x1)*F34(x1, x2)*m23x1+f3(x2)*F34(x2, x2)*m23x2,
-a7*m41x1+f4(x1)*F14(x1, x1)*m34x1+f4(x2)*F14(x1, x2)*m34x2,
-a7*m41x2+f4(x1)*F14(x2, x1)*m34x1+f4(x2)*F14(x2, x2)*m34x2,
-a8*m43x1+f4(x1)*F34(x1, x1)*m14x1+f4(x2)*F34(x1, x2)*m14x2,
-a8*m43x2+f4(x1)*F34(x2, x1)*m14x1+f4(x2)*F34(x2, x2)*m14x2,
-1+m12x1+m12x2,
-1+m14x1+m14x2,
-1+m21x1+m21x2,
-1+m23x1+m23x2,
-1+m32x1+m32x2,
-1+m34x1+m34x2,
-1+m41x1+m41x2,
-1+m43x1+m43x2];
# Variables for the ASP
vars := [m12x1, a1, m12x2, m14x1, a2, m14x2, m21x1, a3, m21x2, m23x1,
a4, m23x2, m32x1, a5, m32x2, m34x1, a6, m34x2, m41x1, a7,
m41x2, m43x1, a8, m43x2];
# Marginal equations
Marg := [m21x1*m41x1-Marg1x1, m21x2*m41x2-Marg1x2, m12x1*m32x1-Marg2x1,
m12x2*m32x2-Marg2x2, m23x1*m43x1-Marg3x1, m23x2*m43x2-Marg3x2,
m14x1*m34x1-Marg4x1, m14x2*m34x2-Marg4x2];
MargVars := {Marg1x1, Marg1x2, Marg2x1, Marg2x2,
Marg3x1, Marg3x2, Marg4x1, Marg4x2};
MargNVars := [-MargN1x1+Marg1x1/(Marg1x1+Marg1x2),
-MargN1x2+Marg1x2/(Marg1x1+Marg1x2),
-MargN2x1+Marg2x1/(Marg2x1+Marg2x2),
-MargN2x2+Marg2x2/(Marg2x1+Marg2x2),
-MargN3x1+Marg3x1/(Marg3x1+Marg3x2),
-MargN3x2+Marg3x2/(Marg3x1+Marg3x2),
-MargN4x1+Marg4x1/(Marg4x1+Marg4x2),
-MargN4x2+Marg4x2/(Marg4x1+Marg4x2)];
```

```
# TBP
if ConvCrit(EqsNorm, vars) = true then
# Compute GB with respect to lex ordering using the built-in F4 algorithm
GB := Groebner[Basis](EqsNorm, plex(V), method = maplef4);
nsols := NSols; mssgs := solve(GB, {V});
# Compute the solutions for the messages
mssgs := evalf(mssgs);
mrgs := evalf(subs(mssgs, Marg));
# Calculate the beliefs
mrgs := solve(mrgs, MargVars);
mrgs := evalf(subs(mrgs, MargNVars));
mrgs := solve(mrgs, indets(mrgs));
evalf(subs(mrgs, MargN1x1))
end if
end proc:
```

B Formal definitions of mathematical structures

B.1 Probability theory

In this section, the formal definitions of probability distribution and random variable are provided. These definitions are taken from [5].

Definition 21. ***(Outcome Space, Event Space, Probability Distribution,Probability Space)****. Let Ω be an* outcome space*, the set of all outcomes of an experiment, and Σ an* event space *over Ω, the set of all possible outcomes. A* probability distribution *over (Ω, Σ) is a function $P\colon \Sigma \mapsto \mathbb{R}$ that satisfies Kolmogorov's axioms:*

1. $0 \leq P(\sigma) \leq 1$, $\forall \sigma \in \Sigma$.

2. $P(\Omega) = 1$.

3. *If* $\sigma_i \cap \sigma_j = \emptyset$, *then* $P(\sigma_i \cup \sigma_j) = P(\sigma_i) + P(\sigma_j)$, $\forall \sigma_i, \sigma_j \in \Sigma$, $i \neq j$.

A probability space *is formed by the triplet (Ω, Σ, P).*

Definition 22. ***(Random Variable)****. Given a probability space (Ω, Σ, P), a* random variable *(RV) is a function $X: \Omega \mapsto \mathbb{R}$ which satisfies the following properties:*

- *"$X \leq x$" $= \{\omega \in \Omega \mid X(\omega) \leq x\}$ is an event for every $x \in \mathbb{R}$,*

- *for the events "$X = -\infty$" $= \{\omega \in \Omega\colon X(\omega) = -\infty\}$ and "$X = \infty$" $= \{\omega \in \Omega \mid X(\omega) = \infty\}$ it must hold that $P(X = -\infty) = P(X = \infty) = 0$,*

- $\mathbf{val}(X) = \{x \in \mathbb{R} \mid \exists \omega \in \Omega\colon X(\omega) = x\}$ *is the image of a RV X.*

$\mathbf{X} = \{X_1, \ldots, X_n\}$ *represents an ordered set of n RVs, and* $\mathbf{x} = \{x_1, \ldots, x_n\}$ *an ordered set of n values from $\mathbb{R}$. The set of values which can be assumed by a set of random variables $\mathbf{X}$ is denoted as* $\mathbf{val}(\mathbf{X})$.

B.2 Algebraic structures

In this section, the formal definitions of the algebraic structures used in this work is provided. These definitions are taken from [45].

Definition 23. ***(Operation).*** *An operation $\cdot$ on a set $\mathbb{X}$ is a function $\cdot : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{X}$.*

Definition 24. ***(Semigroup).*** *The tuple $(\mathbb{X}, \cdot)$ is called a* semigroup *if the operation $\cdot : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{X}$ is* associative*, i.e.*

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z, \ \forall x, y, z \in \mathbb{X} \tag{B.1}$$

Definition 25. ***(Neutral element).*** *Let $\cdot$ be an associative operation on $\mathbb{X}$.*

- *$e \in \mathbf{X}$ is a* left neutral element *for $\cdot$, if $e \cdot x = x \ \forall x \in \mathbb{X}$,*
- *$e \in \mathbf{X}$ is a* right neutral element *for $\cdot$, if $x \cdot e = x \ \forall x \in \mathbb{X}$,*
- *$e \in \mathbf{X}$ is a* neutral element *for $\cdot$, if e is both left and right neutral element for $\cdot$.*

Definition 26. ***(Monoid)*** *A triple $(\mathbb{X}, \cdot, e)$ is a* monoid*, if $(\mathbb{X}, \cdot)$ is a semigroup, and e is a neutral element for $\cdot$.*

Definition 27. ***(Inverse)*** *Let $(\mathbb{X}, \cdot, e)$ be a monoid, and let*

$$a \cdot b = e, \tag{B.2}$$

for $a, b \in \mathbb{X}$. We say that a is a left inverse *of b and b is a* right inverse*of a. Furthermore, if x is a left inverse of a, and z is a right inverse for a, it follows that $x = z$.*

It can be proved, that if the inverse of an element of a monoid exists, it is unique [45, pp. 5].

Definition 28. ***(Group, Commutative group)*** *A triple $(\mathbb{X}, \cdot, e)$ is a* group*, if every element in $\mathbb{X}$ has an inverse. Furthermore, if $\cdot$ is a commutative operation, we say that the group is a* commutative group.

Using these concepts, the formal definitions of a ring and a field can be expressed as follows:

Definition 29. ***(Ring, commutative ring)*** *A quintuple $(\mathbb{A}, +, \cdot, 0, 1)$ is a* ring *if the following conditions are satisfied:*

1. *$(\mathbb{A}, +, 0)$ is a commutative group.*
2. *$(\mathbb{A}, \cdot, 1)$ is a monoid.*
3. *$\cdot$ distributes over $+$ for both sides, i.e.*

$$r \cdot (s + t) = (r \cdot s) + (r \cdot t), \ \forall r, s, t \in \mathbb{A} \tag{B.3}$$

and

$$(s + t) \cdot r = (s \cdot r) + (t \cdot r) \ \forall r, s, t \in \mathbb{A}. \tag{B.4}$$

If operation $\cdot$ is commutative, the ring is called a commutative *ring.*

Definition 30. ***(Field)*** *A quintuple* $(\mathbb{K}, +, \cdot, 0, 1)$ *is a* field *if* $(\mathbb{K}, +, \cdot, 0, 1)$ *is a commutative ring and* $(\mathbb{K} \setminus \{0\}, \cdot, 1)$ *is a commutative group.*

C Dimension of a Variety

In this appendix, a formal definition of the dimension of an affine variety, as well as proofs for some of the results presented in Chapter 3 are provided. All the notation and definitions are taken from the standard text by Cox et. al. [18]. We start this discussion presenting the concept of quotients of polynomial rings.

Definition 31. ***(Congruence modulo*** $\mathbf{I}$***, Equivalence classes, Quotient of a polynomial ring)*** *Let* $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ *be an ideal, and* $f, g \in \mathbb{K}[x_1, \ldots, x_n]$*. If* $f - g \in \mathbf{I}$*,* f *and* g *are said to be* congruent modulo $\mathbf{I}$*, denoted as*

$$f \equiv g \mod \mathbf{I}. \tag{C.1}$$

Congruence modulo $\mathbf{I}$ *partitions* $\mathbb{K}[x_1, \ldots, x_n]$ *into a collection of disjoint sets called* equivalence classes*. For any* $f \in \mathbb{K}[x_1, \ldots, x_n]$*, the class of* f *is the set*

$$[f] = \{g \in \mathbb{K}[x_1, \ldots, x_n] \colon g \equiv f \mod \mathbf{I}\}. \tag{C.2}$$

The set of equivalence classes for congruence modulo $\mathbf{I}$*, given by*

$$\mathbb{K}[x_1, \ldots, x_n]/\mathbf{I} = \{[f] \colon f \in \mathbb{K}[x_1, \ldots, x_n]\}, \tag{C.3}$$

is called the quotient *of* $\mathbb{K}[x_1, \ldots, x_n]$ *modulo* $\mathbf{I}$*.*

Since $\mathbb{K}[x_1, \ldots, x_n]$ is a ring, it is easy to see that the sum and product of equivalence classes in $\mathbb{K}[x_1, \ldots, x_n]$ yield the equivalence classes of the sum and the product of the elements of the classes, i.e.

$$\begin{aligned} &[f] + [g] = [f + g] \\ &[f] \cdot [g] = [f \cdot g]. \end{aligned} \tag{C.4}$$

Definition 32. ***(Coordinate ring, coordinate function)*** *Let* $\mathbf{V} \subset \mathbb{K}^n$ *be an affine variety. The* coordinate ring*, denoted by* $\mathbb{K}[\mathbf{V}]$*, represents the collection of all polynomial functions* $\phi \colon \mathbf{V} \mapsto \mathbb{K}$*.*

The i*-th* coordinate function *on* $\mathbf{V}$ *is a function* $[x_i] \colon \mathbf{V} \mapsto \mathbb{K}$*, such that at each variable* x_i *from* *the polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$ give a polynomial function whose value at point $p \in \mathbf{V}$ is the i-th coordinate of p.*

The sum and product of ideals can be expressed in terms of the intersection and union of varieties, respectively. More formally, this means, given ideals $\mathbf{I} = \langle f_1, \ldots, f_s \rangle, \mathbf{J} = \langle g_1, \ldots, g_t \rangle \subset \mathbb{K}[x_1, \ldots, x_n]$ and their respective varieties $\mathbf{V}(\mathbf{I}), \mathbf{V}(\mathbf{J}) \subset \mathbb{K}$, the sums and products of ideals are given by

$$\begin{aligned} \mathbf{I} + \mathbf{J} = \langle f_1, \ldots, f_s, g_1, \ldots, g_t \rangle \quad &\longrightarrow \quad \mathbf{V}(\mathbf{I}) \cap \mathbf{V}(\mathbf{J}) \\ \mathbf{I} \cdot \mathbf{J} = \langle f_i g_j \colon 1 \leq i \leq r, 1 \leq j \leq s \rangle \quad &\longrightarrow \quad \mathbf{V}(\mathbf{I}) \cup \mathbf{V}(\mathbf{J}). \end{aligned} \tag{C.5}$$

Using these results, we can proceed to define the dimension of a variety as follows:

Definition 33. ***(Coordinate subspace, Dimension of a Variety)*** *A* coordinate subspace *is a vector subspace in $\mathbb{K}^n$ defined by setting some subset of variables $x_1, \ldots, x_n$ equal to zero.*

Let $\mathbf{V}$ be a variety which is the union of a finite number of coordinate subspaces of $\mathbb{K}^n$. Then the dimension *of $\mathbf{V}$, denoted by* $\dim\mathbf{V}$*, is the largest of the dimensions of the subspaces.*

It can be proved that the variety of every monomial ideal is a finite union of coordinate subspaces in $\mathbb{K}^n$ [18, pp. 440]. We can now explore how to compute $\dim\mathbf{V}$ and its connection with the number of solutions of a system of polynomial equations.

Proposition 5. ***(Affine Hilbert function)*** *Let $\mathbf{I}$ be a proper monomial ideal of $\mathbb{K}[x_1, \ldots, x_n]$, and let ${}^aHF_{\mathbf{I}}(s)$, known as the* affine Hilbert function*, be the number of monomials of total degree $\leq s$ that do not lie in $\mathbf{I}$ for all $s \geq 0$, then*

1. *for all s sufficiently large, the affine Hilbert function is equal to a polynomial, known as* affine Hilbert polynomial*, given by*

$$ {}^aHF_{\mathbf{I}}(s) = \sum_{i=0}^{d} b_i \binom{s}{d-i}, \tag{C.6}$$

   where $b_i \in \mathbb{Z}$ and b_0 is positive.

2. *The degree of the affine Hilbert polynomial is the maximum of the dimensions of the coordinate subspaces contained in $\mathbf{V}(\mathbf{I})$*
3. *Let $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ and $>$ be a graded monomial order. Then $\mathbf{I}$ has the same affine Hilbert function as the monomial ideal $\langle \mathrm{LT}(\mathbf{I}) \rangle$*

A full proof of this proposition is beyond the scope of this thesis, but the interested reader can find it in [18, pp. 458]. This leads to the following theorem:

Theorem 11. ***(Dimension theorem)*** *Let $\mathbf{V}(\mathbf{I})$ be an affine variety, where $\mathbf{I} = \mathbf{I}(\mathbf{V}) \subset \mathbb{K}[x_1, \ldots, x_n]$. Using a graded monomial ordering on $\mathbb{K}[x_1, \ldots, x_n]$, the dimension of the variety is given by*

$$\begin{aligned} \dim \mathbf{V}(\mathbf{I}) &= \deg\ {}^aHF_{\langle \mathrm{LT}(\mathbf{I}) \rangle}(s) \\ &= \textit{maximum dimension of a coordinate subspace in } \mathbf{V}(\langle \mathrm{LT}(\mathbf{I}) \rangle). \end{aligned} \tag{C.7}$$

Proof. (Taken from [18])

Using Lemma 1, and third statements of Proposition 5, it follows that the affine Hilbert polynomial of $\mathbf{I}$ is the same as the respective function from the monomial ideal $\langle \mathrm{LT}(\mathbf{I})\rangle$ i.e.

$$^{a}HF_{\mathbf{I}}(s) = \; ^{a}HF_{\langle \mathrm{LT}(\mathbf{I})\rangle}(s). \tag{C.8}$$

From here, using the second part of Proposition 5, by definition, the dimension of the variety $\mathbf{V}(\mathbf{I})$ is given by the degree of the affine Hilbert function of $\mathbf{I}$, and thus, equal to the degree of the affine Hilbert function of $\langle \mathrm{LT}(\mathbf{I})\rangle$. □

Finally, the above theorem helps establishing a criterion to determine whether an affine variety has finitely many points:

Corollary 2. *Let* $\mathbf{V}(\mathbf{I}) \subset \mathbb{K}^n$ *be an affine variety. Then the cardinality of* $\mathbf{V}(\mathbf{I})$*, i.e. the number of elements in* $\mathbf{V}(\mathbf{I})$*, is finite iff* $\dim \mathbf{V}(\mathbf{I}) = 0$.

Proof. (Taken from [18])

Suppose that $\mathbf{V}(\mathbf{I})$ has finitely many elements $a_1, \ldots, a_m \in \mathbb{K}^n$, and let $>$ be a graded monomial ordering on $\mathbb{K}[x_1, \ldots, x_n]$. Then the polynomial in variable x_i and the i-th components of the elements of $\mathbf{V}(\mathbf{I})$, given as

$$f = \prod_{j=1}^{m}(x_i - a_{i_j}), \tag{C.9}$$

lies in $\mathbf{I}(\mathbf{V})$, which means that $\mathrm{LT}(f) = x_i^m \in \langle \mathrm{LT}(\mathbf{I}(\mathbf{V}))\rangle$, for all $1 \leq i \leq n$. This means that $\mathbf{V}(\langle \mathrm{LT}(\mathbf{I}(\mathbf{V}))\rangle) = \{0\}$, i.e. the dimension of all coordinate subspaces in $\mathbf{V}(\langle \mathrm{LT}(\mathbf{I})\rangle)$ is zero. By Eq. (C.7) of Theorem 11, this implies that $\dim \mathbf{V}(\mathbf{I}) = 0$.

Let's now suppose that $\dim \mathbf{V}(\mathbf{I}) = 0$. To show that $\mathbf{V}(\mathbf{I})$ is finite, it suffices to show that for each i, $1 \leq i \leq n$, there can be only finitely many distinct i-th coordinates for the points of $\mathbf{V}(\mathbf{I})$. From Proposition 5, and the Dimension Theorem, this means that the affine Hilbert polynomial is a constant C for s sufficiently large. This mean that the classes $[1], [x_i], [x_i^2], \ldots, [x_i^s]$ are $s+1$ vectors in a vector space of dimension $C \leq s$ and hence, they must be linearly dependent, that is, there are constants a_i such that

$$[0] = \sum_{j=0}^{s} a_{i_j}[x_i^j] = \left[\sum_{j=0}^{s} a_{i_j} x_i^j\right]. \tag{C.10}$$

However this implies that $\sum_{j=0}^{s} a_{i_j} x_i^j$ is a nonzero polynomial in $\mathbf{I}(\mathbf{V})_{\leq s}$, the ideal of polynomials that vanish on $\mathbf{V}$ of degree at most s, which vanishes on $\mathbf{V}$. This implies that that there are only finitely manly distinct i coordinates among the points of $\mathbf{V}$, for all $1 \leq i \leq n$, which means that the cardinality of $\mathbf{V}$ must be finite. □

We can use these results to prove Theorem 4 and Proposition 3. For easiness of reading, the full formulation of these results is reproduced in this appendix.

Theorem 12. *Let* $\mathbf{V}(\mathbf{I}) \subset \mathbb{K}^n$ *be an affine variety and* $\mathbf{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ *be an ideal,* $>$ *be a graded monomial ordering in* $\mathbb{K}[x_1, \ldots, x_n]$*, and* $\mathbf{G}$*, a GB for* $\mathbf{I}$ *with respect to such ordering.* $\mathbf{I}$ *is zero-dimensional iff for each* i*,* $1 \leq i \leq n$*, there is some* $m_i \geq 0$ *such that* $x_i^{m_i} = \mathrm{LT}(g)$ *for some* $g \in \mathbf{G}$.

Proof. (Taken from [18])

If $\mathbf{I}$ is zero-dimensional, $\dim \mathbf{V}(\mathbf{I}) = 0$. It follows from the proof of Corollary 2, that for each variable x_i, $1 \leq i \leq n$, there is some m_i for which $x_i^{m_i} \in \langle \mathrm{LT}(\mathbf{I}) \rangle$. Since $\mathbf{G}$ is a GB for $\mathbf{I}$, therefore, from Theorem 2, $\langle \mathrm{LT}(\mathbf{I}) \rangle = \langle \mathrm{LT}(\mathbf{G}) \rangle$, which means that $x_i^{m_i} \in \langle \mathrm{LT}(\mathbf{G}) \rangle = \langle \mathrm{LT}(g_1), \dots, \mathrm{LT}(g_t) \rangle$, for $g_1, \dots, g_t \in \mathbf{G}$. Hence, $x_i^{m_i} = \mathrm{LT}(g)$ for some $g \in \mathbf{G}$.

Assuming now, that for each i, $1 \leq i \leq n$, there is some $m_i \geq 0$ such that $x_i^{m_i} = \mathrm{LT}(g)$ for some $g \in \mathbf{G}$. This implies that $x_i^{m_i} \in \langle \mathrm{LT}(\mathbf{G}) \rangle$. Since $\mathbf{G}$ is a GB for $\mathbf{I}$, it follows that $\langle \mathrm{LT}(\mathbf{G}) \rangle = \langle \mathrm{LT}(\mathbf{I}) \rangle$. From the proof of Corollary 2, this means that $\dim \mathbf{V}(\mathbf{I}) = 0$, and therefore, $\mathbf{I}$ is zero-dimensional. □

Proposition 6. *Let* $\mathbf{I} \subset \mathbb{K}[x_1, \dots, x_n]$ *be an ideal in an algebraically closed field with GB* $\mathbf{G} = \{g_1, \dots, g_t\}$ *such that* $\mathrm{LT}(g_i) = x_i^{m_i}$. *Then it follows that the variety* $\mathbf{V}(\mathbf{I})$ *contains at most* $m_1 \times m_2 \times \cdots \times m_n$ *points.*

Proof. (Taken from [18])

Since $\mathbf{G}$ is a GB for $\mathbf{I}$, it follows that $x_i^{m_i} \in \langle \mathrm{LT}(\mathbf{I}) \rangle$ for each i, $1 \leq i \leq n$. Then, by definition, the monomials $x_1^{\alpha_1} \dots x_n^{\alpha_n}$ for $\alpha_i \geq m_i$ are all in $\langle \mathrm{LT}(\mathbf{I}) \rangle$, which means that the monomials which are not in the ideal generated by $\langle \mathrm{LT}(\mathbf{I}) \rangle$ must have $\alpha_i \leq m_1 - 1$ for each i. This means that there can be at most $m_1 \times \cdots \times m_n$ monomials not generated by $\langle \mathrm{LT}(\mathbf{I}) \rangle$. □

D

Alternative proof of Theorem 8

To show the convergence of the BP algorithm in case of single loops, here we use techniques from linear algebra, which provides a slightly different flavor to the proof presented in Chapter 4. Nevertheless, as stated in Chapter 3, for linear systems, solving a system of equations using GBs is equivalent to Gaussian elimination. For easiness of reading, the full text of the theorem is reproduced here.

Theorem 13. *The BP algorithm converges to a solution for the MN $\mathcal{M}$ being a single loop.*

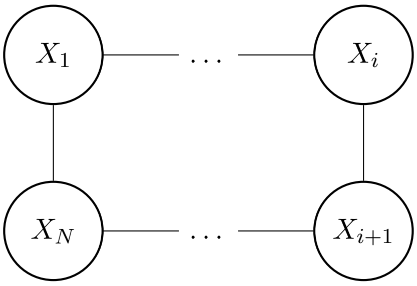


Figure D.1: Graph of the MN of a single Loop with N variable nodes

Proof. Let $\mathcal{M}$ be the MN of a single loop with N variable m-ary variable nodes shown in Figure D.1. If we consider $\mathbf{F}$, the ASP of $\mathcal{M}$ to be a system of polynomials of the message variables only, i.e. the normalization constants are absorbed by the factors, the ASP of $\mathcal{M}$ is given as

$$\mathbf{F} = \left\{ \begin{array}{c} \mu_{1\to 2}(x_1) - \sum_{l=1}^{m} \Psi_{1,2}(x_l, x_1)\mu_{N\to 1}(x_l) \\ \vdots \\ \mu_{1\to 2}(x_m) - \sum_{l=1}^{m} \Psi_{1,2}(x_l, x_m)\mu_{N\to 1}(x_l) \\ \vdots \\ \mu_{i\to i+1}(x_j) - \sum_{l=1}^{m} \Psi_{i,i+1}(x_l, x_j)\mu_{i-1\to i}(x_l) \\ \vdots \\ \mu_{i+1\to i}(x_j) - \sum_{l=1}^{m} \Psi_{i+1,i}(x_l, x_j)\mu_{i+2\to i+1}(x_l) \\ \vdots \end{array} \right\}. \tag{D.1}$$

This system of equations can be written in vector form as

$$\vec{\mu} = \boldsymbol{\Psi}\vec{\mu}, \tag{D.2}$$

where $\vec{\mu}$ is a vector containing all messages, and $\boldsymbol{\Psi}$ is a matrix containing the potentials, respectively given as

$$\vec{\mu} = \begin{pmatrix} \mu_{X_1\to X_2}(x_1) \\ \vdots \\ \mu_{X_N\to X_1}(x_m) \\ \mu_{X_1\to X_N}(x_1) \\ \vdots \\ \mu_{X_2\to X_1}(x_m) \end{pmatrix}, \qquad \text{and} \qquad \boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi_2} \end{pmatrix}, \tag{D.3}$$

and the block matrices $\boldsymbol{\Psi_1}$ and $\boldsymbol{\Psi_2}$, which represent the potentials in the MPEs of the messages that go in the clockwise direction and in the counter-clockwise direction respectively, are given as

$$\boldsymbol{\Psi_1} = \begin{pmatrix} \mathbf{0} & \dots & & \mathbf{A}_1 \\ \mathbf{A}_2 & & & \\ & \mathbf{A}_3 & & 0 \\ 0 & & \ddots & \\ & & & \mathbf{A}_N \end{pmatrix} \qquad \boldsymbol{\Psi_2} = \begin{pmatrix} \mathbf{0} & \dots & & \mathbf{A}'_1 \\ \mathbf{A}'_2 & & & \\ & \mathbf{A}'_3 & & 0 \\ 0 & & \ddots & \\ & & & \mathbf{A}'_N \end{pmatrix}. \tag{D.4}$$

The matrices $\mathbf{A}_i \in \mathbb{R}^{m\times m}$ comprise the pairwise potentials $\Psi_{X_i,X_j}(x_i, x_j)$, given as

$$\mathbf{A}_i = \begin{pmatrix} \Psi_{X_i,X_j}(x_{i_1}, x_{j_1}) & \dots & \Psi_{X_i,X_j}(x_{i_1}, x_{j_n}) \\ \vdots & & \vdots \\ \Psi_{X_i,X_j}(x_{i_n}, x_{j_1}) & \dots & \Psi_{X_i,X_j}(x_{i_n}, x_{j_n}) \end{pmatrix} \tag{D.5}$$

From linear algebra, we know that system has a solution as long as $\det(\boldsymbol{\Psi}) \neq 0$. Expanding the determinant of $\mathbf{\Psi}$ we have

$$\begin{aligned}
\det(\Psi) &= \det(\mathbf{\Psi_1})\det(\mathbf{\Psi_2}) \\
&= \det\left[\begin{pmatrix} \mathbf{0} & \cdots & & & \mathbf{A}_1 \\ \mathbf{A}_2 & & & & \\ & \mathbf{A}_3 & & & 0 \\ 0 & & \ddots & & \\ & & & \mathbf{A}_N & \end{pmatrix}\right] \det\left[\begin{pmatrix} \mathbf{0} & \cdots & & & \mathbf{A}'_1 \\ \mathbf{A}'_2 & & & & \\ & \mathbf{A}'_3 & & & 0 \\ 0 & & \ddots & & \\ & & & \mathbf{A}'_N & \end{pmatrix}\right] \qquad \text{(D.6)}
\end{aligned}$$

From basic linear algebra we know, that switching a row on a matrix inverts the sign of its determinant. By switching the rows corresponding to the matrix $\mathbf{A}$ and $\mathbf{A'}$ to the end, and moving all other rows upwards, we end up having two diagonal block matrices. Since the same number of switches where performed in matrices $\mathbf{\Psi_1}$ and $\mathbf{\Psi_2}$, then the sign of the determinant remains the same, i.e.

$$\begin{aligned}
&= \det\left[\begin{pmatrix} \mathbf{A}_2 & & & & 0 \\ & \mathbf{A}_3 & & & \\ & & \ddots & & \\ 0 & & & \mathbf{A}_N & \\ & & & & \mathbf{A}_1 \end{pmatrix}\right] \det\left[\begin{pmatrix} \mathbf{A}'_2 & & & & 0 \\ & \mathbf{A}'_3 & & & \\ & & \ddots & & \\ 0 & & & \mathbf{A}'_N & \\ & & & & \mathbf{A}'_1 \end{pmatrix}\right] \\
&= \prod_{i=1}^{N} \det(\mathbf{A}_i) \prod_{i=1} \det(\mathbf{A}'_i) \\
&= \prod_{i=1}^{N} \det(\mathbf{A}_i)\det(\mathbf{A}'_i) \qquad \text{(D.7)}
\end{aligned}$$

From here we can see, that $\det(\mathbf{\Psi}) \neq 0$ since by construction $\det(\mathbf{A}_i), \det(\mathbf{A}'_i) \neq 0$. Therefore, the system converges to a solution. □