On Generalized Survey Propagation: Normal Realization and Sum-Product Interpretation

Ronghui Tu, Yongyi Mao and Jiying Zhao
School of Information Technology and Engineering, University of Ottawa
800 King Edward Ave., Ottawa, Ontario, Canada K1N 6N5
Email: {rtu, yymao, jyzhao}@site.uottawa.ca

***Abstract*— A celebrated algorithmic discovery in solving constraint satisfaction problems, survey propagation (SP) and its generalization have recently demonstrated their power in areas of communications and data compression. It is known that under certain Markov Random Field (MRF) formalism of k-SAT problems, SP may be interpreted as an instance of belief propagation (BP). In this paper, we borrow the notion of generalized states from system theory and coding theory, and introduce a new MRF formalism — normal realization — for k-SAT problems. We show that when BP applies to this MRF, generalized SP is resulted. This new MRF formalism appears to be simpler than the existing one and the interpretation of SP as BP in this framework also appears more transparent.**

I. Introduction

At the heart of the theory of computation, k-SAT problems are classical NP-complete problems [1]. Extensive study has been carried out in order to understand the hardness of these problems and to develop efficient solvers. A recent celebrated result in k-SAT problems is the work of Mézard, Parisi and Zecchina [2], published in *Science*, where a highly efficient algorithm — derived from techniques in statistical physics — was introduced. As shown in [2] and various follow-up developments (see, e.g., [3]), this algorithm remains effective even for very large and difficult instances of random k-SAT problems. The central part of this algorithm is an iterative message-passing procedure — known as *survey propagation* — that operates on the *factor-graph* representation [4] of the problem instance.

As its astonishing performance is yet to be fully understood, survey propagation (referred to as SP hereafter), as a general methodology for solving constraint-satisfaction problems, has been applied to various other settings (see, e.g., [5]). In the context of communications and data compression, very recently, Yu and Aleksic have applied SP to coding for a special class of broadcast channels [6], and Wainwright and Maneva have applied a generalized version of SP [3] to source-coding problems [7]; in both cases, great successes were demonstrated.

Introduced by Maneva, Mossel and Wainwright [3], generalized SP is a family of message-passing algorithms — which we will denote by SP(γ) in this paper — parametrized by a real number $\gamma \in [0, 1]$. It is shown in [3] that when $\gamma = 1$, generalized SP reduces to the standard SP, and it is observed that for some $\gamma < 1$, SP(γ) often outperforms SP(1). In addition to the introduction of generalized SP, an important contribution of [3] is an interpretation of generalized SP as an instance of the well-known *belief propagation* (BP) or *sum-product* algorithm: by defining a Markov Random Field (MRF) on the space of "extended" configurations, where each variable is allowed to take an additional "joker" symbol ($*$), the authors of [3] show that the SP message-passing rule for the original k-SAT problem is equivalent to the sum-product message-passing rule on the extended MRF. The importance of this perspective is at least two-fold. On one hand, it allows the well-developed analytic tools for the sum-product algorithm to be used in the understanding of the SP and generalized SP algorithms. On the other hand, since a partial ordering can be naturally defined for the extended configurations, the valid extended configurations (namely, those with non-zero probability mass under the MRF model) form a hierarchical (lattice) structure that bridges the original satisfying configurations; that is, lifting the k-SAT problem from the original configuration space to the extended configuration space provides a combinatorial framework, potentially enabling a deeper understanding of the inherent structure of the problem.

In this paper, we present a different MRF representation for k-SAT problems, also on the extended configuration space. For a given instance of k-SAT problem, the global probability mass function (PMF) under our MRF model is identical to that of [3], and the SP message-passing rule for the original k-SAT problem can also be reduced from the sum-product message-passing rule on our MRF model. The advantage of our MRF lies in the following aspects. First, using a "normal realization" represented by a Forney graph [8], our representation emphasizes the notion of state, (more precisely, the notion of generalized state [8]) familiar to the coding-theory community. As is well known in coding theory, the concept of state is fundamental in understanding the structure of codes. We expect that our introduction of (generalized) states may potentially be useful for understanding the combinatorial structure and properties of the k-SAT problems. Second, although it is arguable that our MRF representation is equivalent to that of [3] in terms of the valid extended configurations and the defined PMF, our representation is conceptually simpler and perhaps more natural. In particular, each generalized state in our MRF takes values only from a set of six elements (and for valid extended configurations, only four of the elements are used). Moreover, on our Forney-graph representation, which has similar structure as the original factor graph, the explicit

1-4244-0504-1/06/$20.00 ©2006 IEEE

introduction of state variables makes the reduction of sum-product messages to SP messages more transparent.

Attracting significant research interest, it appears that the methodology of SP brings great promise to solving a large variety of communication problems, which share with k-SAT problems the common nature of finding constraint-satisfying configurations. We believe that the normal realization introduced in this paper allows this important algorithm to be more accessible to the wider engineering community. For this reason, we have chosen to present this paper in a way slightly deviating from the "canonical" computer-science language on this subject, with the hope that it may suit better the communication and coding-theory community.

The remainder of this paper is organized as follows. In Section II, we introduce the k-SAT problem and generalized SP. In Section III, we present our normal realization formalism for the k-SAT problems and its representation using Forney graphs. In Section IV, we show that SP message-passing rule can be reduced from the BP message-passing rule on the Forney graphs. We briefly conclude the paper in Section V. Length constraints often preclude our desire of elaboration, and for the same reason, all proofs are omitted.

II. The k-SAT Problems and generalized SP

An instance of k-SAT problem may be formulated as follows.

Let V be a finite set indexing a set of binary $\{0,1\}$-valued variables $\{x_v : v \in V\}$ and C be another finite set indexing a set of constraints $\{z_c : c \in C\}$. Using a standard notation in graphical-model literature, for any subset $U \subseteq V$, x_U denotes the variable set $\{x_v : v \in U\}$. For any $c \in C$, we use $V(c)$ to denote the set of indices that index the variables involved in constraint z_c. Likewise, for any $v \in V$, we use $C(v)$ to denote the set of indices that index the constraints involving variable x_v. Given k, in any instance of k-SAT problem, the cardinality $|V(c)|$ of $V(c)$ is precisely k for every $c \in C$. Each constraint z_c is specified via a binary vector $\{L_{v,c} : v \in V(c)\}$, where $L_{v,c}$ is the *preferred value* of x_v by constraint z_c. Constraint z_c is then defined as that at least one variable in $x_{V(c)}$ take its preferred value by z_c. When — under a slight abuse of notation — treating constraint z_c as the indicator function defining the constraint, z_c may be written as $z_c(x_{V(c)}) := \left[\sum_{v\in V(c)}[x_v = L_{v,c}] > 0\right]$, where the notation $[P]$ for any boolean proposition P is the Iverson's Convention [4], namely, evaluating to 1 if P, and to 0 otherwise. The k-SAT problem is then to determine whether all constraints $\{z_c : c \in C\}$ can be satisfied simultaneously and if so to find a configuration of x_V that satisfies all the constraints. Using a function notation, the global constraint associated with the problem can be represented by the indicator function

$$\prod_{c\in C} z_c(x_{V(c)}). \qquad (1)$$

Then the k-SAT problem may be equivalently phrased as finding a solution for equation $\prod_{c\in C} z_c(x_{V(c)}) = 1$ or conversely concluding no such solution exists. Clearly the factorization of (1) may be represented by a factor graph [4]. It is convenient to treat each $L_{v,c}$ as the label for edge (x_v, z_c) on the factor graph, and use dashed edge to represent label 0 and solid edge to represent label 1. The factor-graph representation of a toy 3-SAT problem is shown in Figure 1.

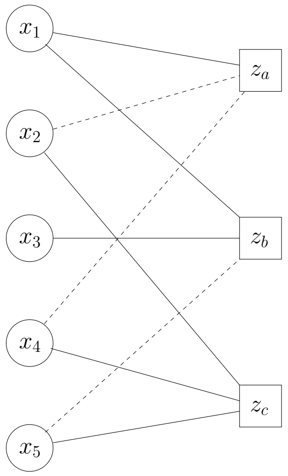

Fig. 1. A factor graph for 3-SAT problem specified by formula $(x_1 \vee \overline{x}_2 \vee \overline{x}_4) \wedge (x_1 \vee x_3 \vee \overline{x}_5) \wedge (x_2 \vee x_4 \vee x_5)$. Logic operation notations are used here to define the problem, where $\vee$ denotes logic OR, $\wedge$ denotes logic AND, and the horizontal bar on a variable denotes the negation of the variable. The function represented by the factor graph is $z_a(x_1, x_2, x_4) \cdot z_b(x_1, x_3, x_5) \cdot z_c(x_3, x_4, x_5)$.

As a solver for k-SAT problems, generalized SP, or SP(γ), is a family of message-passing algorithms parametrized by a real number $\gamma \in [0, 1]$. Similar to BP, in SP(γ), messages are passed between variable nodes and function nodes in the above-defined factor graph. For the purpose of describing the SP message-passing rules and for future use, we introduce the following notations. For any edge (x_v, z_c) in the factor graph, $C_c^{\mathtt{u}}(v)$ denotes the set $\{b \in C(v) \setminus \{c\} : L_{v,b} \neq L_{v,c}\}$, and $C_c^{\mathtt{s}}(v)$ denotes the set $\{b \in C(v) \setminus \{c\} : L_{v,b} = L_{v,c}\}$. For any $v \in V$, $C^+(v)$ denotes the set $\{b \in C(v) : L_{v,b} = 1\}$, and $C^-(v)$ denotes the set $\{b \in C(v) : L_{v,b} = 0\}$.

The SP message-passing rule is then as follows.

The message passed from variable node x_v to function node z_c is a triplet of real numbers $(\Pi^{\mathtt{u}}_{v\to c}, \Pi^{\mathtt{s}}_{v\to c}, \Pi^{*}_{v\to c})$, where

$$\Pi^{\mathtt{u}}_{v\to c} = \left(1 - \gamma \prod_{b\in C_c^{\mathtt{u}}(v)} (1 - \eta_{b\to v})\right) \prod_{b\in C_c^{\mathtt{s}}(v)} (1 - \eta_{b\to v}) \qquad (2)$$

$$\Pi^{\mathtt{s}}_{v\to c} = \left(1 - \prod_{b\in C_c^{\mathtt{s}}(v)} (1 - \eta_{b\to v})\right) \prod_{b\in C_c^{\mathtt{u}}(i)} (1 - \eta_{b\to v}) \qquad (3)$$

$$\Pi^{*}_{v\to c} = \prod_{b\in C_c^{\mathtt{s}}(v)} (1 - \eta_{b\to v}) \prod_{b\in C_c^{\mathtt{u}}(v)} (1 - \eta_{b\to v}) \qquad (4)$$

The message passed from constraint node z_c to variable node x_v is a real number $\eta_{c\to v} \in [0, 1]$, given by

$$\eta_{c\to v} = \prod_{u\in V(c)\setminus\{v\}} \frac{\Pi^{\mathtt{u}}_{u\to c}}{\Pi^{\mathtt{u}}_{u\to c} + \Pi^{\mathtt{s}}_{u\to c} + \Pi^{*}_{u\to c}}. \qquad (5)$$

The initialization of SP messages is usually random.

Similar to the sum-product algorithm, upon convergence, SP also computes a "summary message", a triplet $(\mu_v(0), \mu_v(1), \mu_v(*))$ of real numbers, at each variable node x_v according to the following rule.

$$\begin{aligned}
\mu_v(1) &\propto \left(1-\gamma \prod_{b\in C^+(v)} (1-\eta_{b\to v})\right) \prod_{b\in C^-(v)} (1-\eta_{b\to v}) \\
\mu_v(0) &\propto \left(1-\gamma \prod_{b\in C^-(v)} (1-\eta_{b\to v})\right) \prod_{b\in C^+(v)} (1-\eta_{b\to v}) \\
\mu_v(*) &\propto \gamma \prod_{b\in C^+(v)} (1-\eta_{b\to v}) \prod_{b\in C^-(v)} (1-\eta_{b\to v})
\end{aligned}$$

When $\gamma = 1$, generalized SP reduces to the standard SP of [5].

Usually, SP is carried out in conjunction with a "decimation" procedure. In the decimation procedure, the *bias* $B(v) := |\mu_v(0) - \mu_v(1)|$ at each $v \in V$ is calculated, and the variable with the highest bias is fixed to 0 or 1 according to the sign of $\mu_v(0)-\mu_v(1)$: x_v is set to 0 if $\mu_v(0)-\mu_v(1) > 0$, and to 1 otherwise. The k-SAT formula is then simplified and SP is applied again. This process iterates until the reduced problem is simple enough for a local search algorithm.

It has been demonstrated that the generalized SP algorithm described above is highly effective even for large and difficult instances of random k-SAT problems. For a more comprehensive review of SP and its generalization, the reader is referred to [5] and [3].

III. Normal Realization of k-SAT Problem

For a given instance of k-SAT problem specified by a factor graph G with variable nodes $\{x_v : v \in V\}$, function nodes $\{z_c : c \in C\}$, edge set $E(G)$ and edge labels $\{L_{v,c} : (x_v, z_c) \in E(G)\}$, we now construct a "normal realization" of the problem on the space of extended configurations, where we borrow the term "normal realization", introduced in [8], from system theory and coding theory. The result of this exercise is another factor graph (with slightly modified semantics), known as a *Forney graph*. The Forney graph is defined via a set of *extended symbol variables*, a set of *generalized state variables* and a set of *local functions*.

A. Extended symbol variables and generalized state variables

For each variable x_v in the given problem instance, we introduce an extended symbol variable (referred to as symbol variable hereafter) y_v, taking values from the alphabet $\{0, 1, *\}$ — with one extra symbol $*$ — which we will denote by $\mathcal{A}^*$. A configuration of y_V is intended to indicate a "cluster" of configurations of x_V, where the symbol $*$ at any $v \in V$ indicates that variable x_v in the configurations can take value either 0 or 1 in the cluster.

We also introduce two other alphabets $\mathcal{L} := \{\mathtt{F}, \overline{\mathtt{F}}, *\}$ and $\mathcal{R} := \{\mathtt{F}, *\}$. Then for each edge (x_v, z_c) in $E(G)$, we introduce a *generalized state variable* (referred to as state variable hereafter) $s_{v,c}$ in the normal realization, where $s_{v,c}$ takes values from $\mathcal{L} \times \mathcal{R}$. We call the $\mathcal{L}$-component of $s_{v,c}$ the *left state*, denoted by $s^L_{v,c}$, and the $\mathcal{R}$-component of $s_{v,c}$ the *right state*, denoted by $s^R_{v,c}$.

It is worth noting that in the definitions of $\mathcal{L}$ and $\mathcal{R}$, we have made an abuse of notation concerning letters $\mathtt{F}$ and $\overline{\mathtt{F}}$, in order to simplify the upcoming presentation. Letters $\mathtt{F}$ and $\overline{\mathtt{F}}$ may be interpreted in two ways — either *symbolically* or *numerically*: in the *symbolic* interpretation, they are simply interpreted as symbols, without any numerical meaning; in a *numerical* interpretation, $\mathtt{F}$ and $\overline{\mathtt{F}}$ are interpreted as numerical values in $\{0, 1\}$, where $\mathtt{F}$ and $\overline{\mathtt{F}}$ are negation of each other, namely, if $\mathtt{F} = 0, \overline{\mathtt{F}} = 1$, and if $\mathtt{F} = 1, \overline{\mathtt{F}} = 0$. Clearly, there are two different ways to interpret $\mathcal{L}$ and $\mathcal{R}$ numerically, and we will resolve this ambiguity by specifying the value of $\mathtt{F}$.

B. Local functions

The set of local functions consist of *left functions* — each for an $x_v, v \in V$, and denoted by g_v — and a set of *right functions* — each for a constraint $z_c, c \in C$, and denoted by f_c.

To define these local functions, it is useful to introduce the following notations.

For any arbitrary finite set U, any vector t of arbitrary length consisting of the elements in U, and any given element $e \in U$, we use notation $n(t; e)$ to denote the number of occurrences of symbol e in t.

Let mapping $r : \mathcal{L}^{k-1} \to \mathcal{R}$ be defined as

$$r(t) := \begin{cases} \mathtt{F}, & \text{if } n(t; \overline{\mathtt{F}}) = k-1 \\ *, & \text{otherwise} \end{cases}$$

This mapping will be used in defining the right functions.

Let mapping $l : (\mathcal{A}^*)^m \to \mathcal{A}^* \cup \{\diamond\}$ for an arbitrary m be defined as

$$l(t) := \begin{cases} 1, & \text{if } n(t;1) > 0, \text{ and } n(t;0) = 0 \\ 0, & \text{if } n(t;0) > 0, \text{ and } n(t;1) = 0 \\ *, & \text{if } n(t;1) = 0, \text{ and } n(t;0) = 0 \\ \diamond, & \text{otherwise} \end{cases}$$

This mapping will be used in defining the left functions. The symbol $\diamond$ is used to indicate that there exists a conflict in vector t, in the sense that t contains both letter 1 and letter 0.

For any $v \in V$ and any subset $C' \subseteq C(v)$, we use $s_{v,C'}$ to denote the variable set $\{s_{v,c} : c \in C'\}$. Likewise, for any $c \in C$ and any subset $V' \subseteq V(c)$, we use $s_{V',c}$ to denote the variable set $\{s_{v,c} : v \in V'\}$. Similar notations apply to left and right states.

With these notations, the local functions are defined as follows.

The right function f_c for each $c \in C$ is defined as

$$f_c(s_{V(c),c}) := \prod_{v\in V(c)} [r(s^L_{V(c)\setminus\{v\},c}) = s^R_{v,c}].$$

We note that in the right functions, both left states and right states are interpreted symbolically, where $\mathtt{F}$ indicates that the symbol variable is *forced* to take the preferred value by z_c, and

$\overline{\mathtt{F}}$ indicates that the variable is *forced* to take the *opposite* of the preferred value. The preferred value itself is irrelevant. The right function then dictates that at least one symbol variable involved in z_c takes its preferred value by z_c.

The left function g_v for each $v \in V$ is defined as

$$g_v(y_v, s_{v,C(v)}) := W(y_v, s_{v,C(v)}) \cdot [y_v \sim l(s^R_{v,C(v)})] \times \prod_{c \in C(v)} [s^L_{v,c} = l(y_v, s^R_{v,C(v)\setminus\{c\}})],$$

where

$$W(y_v, s_{v,C(v)}) := \begin{cases} \omega_*, & \text{if } n(s^R_{v,C(v)}; *) = |C(v)| \\ & \text{and } y_v = * \\ \omega_{\mathtt{u}}, & \text{if } n(s^R_{v,C(v)}; *) = |C(v)| \\ & \text{and } y_v \neq * \\ 1, & \text{otherwise,} \end{cases}$$

for some pre-specified non-negative ω_* and $\omega_{\mathtt{u}}$, and a boolean proposition $a \sim b$ is defined to be true if $a = b$ or $b = *$, and false otherwise. We note that in the left functions, both left states and right states are interpreted numerically — with $\mathtt{F}$ interpreted as the preferred value — so that the mapping $l(\cdot)$ is well defined. More precisely, for any edge $(x_v, z_c) \in E(G)$, the corresponding left and right state (those indexed by (v, c)) take the numerical interpretation with $\mathtt{F} := L_{v,c}$. The left function then dictates that all left and right states involved in g_v are consistent with each other and with y_v. The role of $W(\cdot)$ is to adjust the weight, or probability mass, of the symbol configurations (i.e., the clusters of x_V).

C. Normal realization and Forney graph

Let $s_{V,C}$ denote collectively the set of all state variables. We define a global function F via the above-defined local functions:

$$F(y_V, s_{V,C}) := \prod_{v \in V} g_v(y_v, s_{v,C(v)}) \cdot \prod_{c \in C} f_c(s_{V(c),c}). \quad (6)$$

Then function F is readily expressed as a factor graph. Notice that in this factor graph, all symbol variables have degree one and all state variables have degree two, giving rise to a normal realization [8]. Following [8], such a factor graph can be more compactly represented as a Forney graph, where each state variable is represented as an edge, each symbol variable represented as a terminal or "half-edge", and local functions represented as nodes. The Forney graph representing the normal realization of the toy 3-SAT problem in Figure 1 is shown in Figure 2. It is apparent that the Forney graph representing the normal realization resembles the original factor graph describing the k-SAT problem.

As is standard, when treating function F (upon normalization) as a PMF, the factorization structure of F defines an MRF.

We say that a symbol-state configuration of $(y_V, s_{V,C})$ is *valid* if it gives rise to non-zero value for function F. The following lemma is easy to prove.

Lemma 1: If $(y_V, s_{V,C})$ is a valid symbol-state configuration under F, then

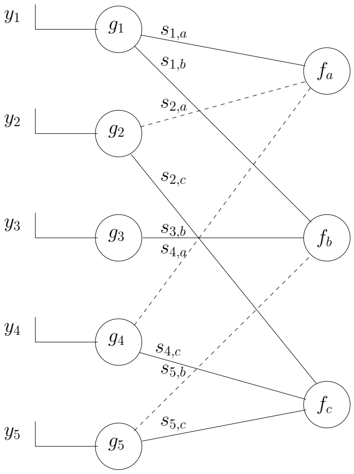

Fig. 2. The Forney graph representing the normal realization of the toy problem in Figure 1.

1) for every state configuration $s_{v,c}$, it holds that $s_{v,c} \neq \overline{\mathtt{F}}\mathtt{F}$ and that $s_{v,c} \neq *\mathtt{F}$, and
2) $F(y_V, s_{V,C}) = \omega_*^{n_*(y_V, s_{V,C})} \cdot \omega_{\mathtt{u}}^{n_{\mathtt{u}}(y_V, s_{V,C})}$, where $n_*(y_V, s_{V,C})$ and $n_{\mathtt{u}}(y_V, s_{V,C})$ are respectively the cardinalities of set $\{v \in V : n(s^R_{v,C(v)}; *) = |C(v)|, y_v = *\}$ and set $\{v \in V : n(s^R_{v,C(v)}; *) = |C(v)|, y_v \neq *\}$.

The first part of this lemma suggests that there are in fact only four values that a state variable may take in valid symbol-state configurations. As a consequence, we will see that when BP is applied to the Forney graph, messages are all quadruplets.

The second part of this lemma suggests that the PMF under this MRF model is identical to that of [3], since an equivalent result is shown for the MRF in [3]. As such, the reader is referred to [3] for a combinatorial interpretation of the valid symbol configurations.

It is remarkable that in this formalism, the left states and right states are "entangled" in the following sense. For any given state variable $s_{v,c}$, its right state must be compatible with all other left states involved in f_c; and its left state must be compatible with all other right states involved in g_v and with the configuration of symbol variable y_v. Clearly, as in code realizations, the state variables introduced here completely capture the interaction between constraints. Fundamental in studying the structure of codes, it is our belief that the notions of state and normal realization will also play an important role in understanding the combinatorial structure of k-SAT problems.

IV. SP AS AN INSTANCE OF BP

Now we consider applying the sum-product message-passing rule to the Forney graphs, where we will use $\rho_{c\to v}$ to denote the message passed from a right function f_c to a left function g_v, and use $\lambda_{v\to c}$ to denote the message passed from left function g_v to right function f_c. We note that both $\rho_{c\to v}$ and $\lambda_{v\to c}$ are functions on the state space $\mathcal{L} \times \mathcal{R}$. The first part of Lemma 1 implies that $\rho_{c\to v}(\overline{\mathtt{F}}\mathtt{F}) = \rho_{c\to v}(*\mathtt{F}) = \lambda_{v\to c}(\overline{\mathtt{F}}\mathtt{F}) = \lambda_{v\to c}(*\mathtt{F}) = 0$. This makes each message a quadruplet, the form of which is summarized in Lemma 2.

Lemma 2: The sum-product message-passing rule applied on the Forney graph defined earlier gives rise to:

$$
\begin{aligned}
\rho_{c\to v}(\mathtt{FF}) &= \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\overline{\mathtt{F}}*),\\
\rho_{c\to v}(\overline{\mathtt{F}}*) &= \prod_{u\in V(c)\setminus\{v\}} \left(\lambda_{u\to c}(\mathtt{F}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\overline{\mathtt{F}}*)\right)\\
&\quad + \sum_{u\in V(c)\setminus\{v\}} \left(\lambda_{u\to c}(\mathtt{FF}) - \lambda_{u\to c}(\mathtt{F}*) - \lambda_{u\to c}(**)\right)\times\\
&\quad \prod_{w\in V(c)\setminus\{u,v\}} \lambda_{w\to c}(\overline{\mathtt{F}}*) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\overline{\mathtt{F}}*),\\
\rho_{c\to v}(\mathtt{F}*) &= \prod_{u\in V(c)\setminus\{v\}} \left(\lambda_{u\to c}(\mathtt{F}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\overline{\mathtt{F}}*)\right)\\
&\quad - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\overline{\mathtt{F}}*),\\
\rho_{c\to v}(**) &= \prod_{u\in V(c)\setminus\{v\}} \left(\lambda_{u\to c}(\mathtt{F}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\overline{\mathtt{F}}*)\right)\\
&\quad - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\overline{\mathtt{F}}*),
\end{aligned}
$$

and

$$
\begin{aligned}
\lambda_{v\to c}(\mathtt{FF}) &= \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\overline{\mathtt{F}}*)\\
&\quad \times \prod_{b\in C_c^{\mathtt{s}}(v)} \left(\rho_{b\to v}(\mathtt{FF}) + \rho_{b\to v}(\mathtt{F}*)\right),\\
\lambda_{v\to c}(\overline{\mathtt{F}}*) &= \prod_{b\in C_c^{\mathtt{s}}(v)} \rho_{b\to v}(\overline{\mathtt{F}}*)\\
&\quad \times \left(\prod_{b\in C_c^{\mathtt{u}}(v)} \left(\rho_{b\to v}(\mathtt{F}*) + \rho_{b\to v}(\mathtt{FF})\right)\right.\\
&\quad \left. -(1-\omega_{\mathtt{u}}) \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\mathtt{F}*)\right),\\
\lambda_{v\to c}(\mathtt{F}*) &= \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\overline{\mathtt{F}}*)\\
&\quad \times \left(\prod_{b\in C_c^{\mathtt{s}}(v)} \left(\rho_{b\to v}(\mathtt{F}*) + \rho_{b\to v}(\mathtt{FF})\right)\right.\\
&\quad \left. -(1-\omega_{\mathtt{u}}) \prod_{b\in C_c^{\mathtt{s}}(v)} \rho_{b\to v}(\mathtt{F}*)\right),\\
\lambda_{v\to c}(**) &= \omega_* \prod_{b\in C_c^{\mathtt{u}}(v)\cup C_c^{\mathtt{s}}(v)} \rho_{b\to v}(**).
\end{aligned}
$$

Theorem 1: Let ω_* equal γ, the parameter in generalized SP. If $\omega_* + \omega_{\mathtt{u}} = 1$, and on every edge (g_v, f_c) in the Forney graph the BP messages are initialized such that $\rho_{c\to v}(\mathtt{F}*) = \rho_{c\to v}(**) = \rho_{c\to v}(\overline{\mathtt{F}}*)$ and $\rho_{c\to v}(\overline{\mathtt{F}}*) + \rho_{c\to v}(\mathtt{FF}) = 1$, then BP messages and SP messages are related by

$$
\begin{aligned}
\rho_{c\to v}(\mathtt{FF}) &= \eta_{c\to v},\\
\lambda_{v\to c}(\overline{\mathtt{F}}*) &= \Pi^{\mathtt{u}}_{v\to c},
\end{aligned}
$$

and

$$\lambda_{v\to c}(\mathtt{F}*) + \lambda_{v\to c}(**) = \Pi^{\mathtt{s}}_{v\to c} + \Pi^{*}_{v\to c}.$$

That is, under the condition of this theorem, the SP(γ) message-passing rule is completely recovered.

It is worth noting that any state variable in normal realizations may be viewed as a data structure serving as a "communication channel". In the context of sum-product message passing, right state $s^R_{v,c}$ serves the purpose of receiving messages from all other left-states involved in f_c, and left state $s^L_{v,c}$ serves the purpose of receiving messages from all other right states (and the symbol y_v) involved in g_c. Realizing this, the derivation of sum-product message-passing rule in Lemma 2 and reduction of BP messages to SP messages in Theorem 1 are nearly transparent.

V. Conclusion

In this paper, we present a new simple MRF formalism for k-SAT problems, based on the notion of normal realization. We show that generalized SP may be interpreted as an instance of BP on this MRF model, and that the reduction from BP messages to SP messages appears simpler than that of [3]. In addition, we expect that the explicit exhibition of "states" in this formalism may provide additional insights for understanding the structures of the studied problems, and may potentially be extended to settings beyond k-SAT problems. An effort along this direction and some results are presented in [9].

As many problems in communications and data compression may be formulated as constraint-satisfaction problems, the methodology of survey propagation has the potential of impacting various related areas. With this paper, it is our hope that we have made this important algorithm more easily accessible to the wider community of information theorists, coding theorists and communication engineers.

References

[1] S. A. Cook, "The complexity of theorem-proving procedures," in *3rd Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, 1971, pp. 151–158.

[2] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems," *Science*, , no. 297, pp. 812–815, 2002.

[3] E. Maneva, E. Mossel, and M. J. Wainwright, "A new look at survey propagation and its generalizations," in *SODA*, 2005, pp. 1089–1098.

[4] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[5] A. Brauntein, M. Mézard, M. Weight, and R. Zecchina, *Constraint satisfaction by survey propagation*, Sep. 2003, http://arxiv.org/abs/cond-mat/0212451.

[6] W. Yu and M. Aleksic, "Coding for the Blackwell channel: a survey propagation approach," in *IEEE International Symposium on Information Theory (ISIT)*, Adelaide, Australia, 2005, pp. 1583–1587.

[7] M. J. Wainwright and E. Maneva, "Lossy source coding via message-passing and decimation over generalized codewords of LDGM codes," in *IEEE International Symposium on Information Theory (ISIT)*, Adelaide, Australia, 2005, pp. 1493–1497.

[8] G. D. Forney, Jr., "Codes on graphs: normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[9] R. Tu, Y. Mao, and J. Zhao, "Towards a unified solution for constraint-satisfaction problems: a survey-propagation approach based on normal realizations," in *Proc. 23rd Biennial Symposium on Communications*, May 2006.