Is SP BP?

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—In this paper, we investigate the problem how SP may be reduced from BP under certain Markov random field formalism of constraint-satisfaction problems. We show that unlike the case of $k$-SAT problems, certain extra manipulation of BP messages as well as some condition on the structure of the problem may be needed for BP to reduce to SP.

I. INTRODUCTION

Having its origin in statistical physics, survey propagation (SP) [1] has lately been regarded as a revolutionary algorithm in solving classes of hard constraint-satisfaction problems (see, e.g., [1], [2]). The discovery and first application of SP are in the context of solving classic NP-complete problems, the $k$-SAT problems [1], where SP was shown to be efficient even in the well-known hard regime. This celebrated discovery has since motivated the application of SP to other hard constraint-satisfaction problems (CSPs), such as the graph-coloring (or $q$-COL) problems [2], as well as problems in source coding [3] and channel coding [4], where great successes are demonstrated.

As a message-passing algorithm, it has been realized that SP shares much similarity with another recent algorithmic synthesis, belief propagation (BP) or the sum-product algorithm [5], developed for decoding error correction codes and having revolutionized the field of communications. In the continuing research efforts of understanding the algorithmic nature of SP, one natural question arising is how SP is related to BP.

To that end, SP for $k$-SAT problems has been understood as a special case of BP [6]. Extending the results of [6], the authors of [7] introduced a Markov random field (MRF) formalism of $k$-SAT problems, from which a generalized SP algorithm reduces from the BP message-passing rules. Following up on [7], in [8] we showed that an equivalent but cleaner formulation of the MRF in [7] can be achieved via the notion of normal realizations, or Forney graphs, where a key ingredient is the introduction of generalized states of a particular form. In [8], the Forney-graph MRF formalism allows a more transparent reduction of SP from BP for $k$-SAT problems, although it remains intriguing that a peculiar initialization condition is required for the reduction, similar to that in [7].

These results seem to suggest, convincingly to many, that rather than being understood as an algorithm of a new kind, SP should be treated simply as a member of the BP family. Examining whether such an interpretation is valid — namely, to what extent “SP is BP” — forms the central interest of this paper.

The line of research presented in this paper was initiated by our recent observation [9] that for 3-COL problems, the MRF formalism in the style of [7] or [8] does not naturally allow BP to reduce to SP, which is different from the case of $k$-SAT problems. Specifically, in 3-COL problems, certain extra manipulation of BP messages, referred to as state decoupling, is needed for BP messages to reduce to SP messages.

In this paper, we investigate the general rules for BP to reduce to SP for arbitrary CSPs when the MRF underlying BP is formalized in the style of [7] or [8]. We show that even the state-decoupling operation may not guarantee the message reduction in the general case. We give a sufficient condition, in terms of the structure of the CSP, under which state-decoupling operation allows BP to reduce to SP. It appears that both $k$-SAT and 3-COL problems satisfy this condition, allowing for the previously reported reduction in [8] and [9].

The development of this work relies on several of our previous results concerning SP. One of those results is a simpler formulation of SP message-passing rule as “probabilistic token passing” (PTP) [10], which will be reviewed in this paper. We note that prior to [10], SP was mainly formulated using the language of statistical physics (see, e.g., [11], [12]), rather distant from the vocabulary of computer scientists or electrical engineers. In the PTP formulation of SP, a token is a subset of the variable alphabet, and PTP is essentially a rule of passing random tokens on the factor graph representation of the problem. A key perspective of this formulation is the extension of variable alphabets to their power sets. We believe that this simple perspective, which is only implicitly upon in some earlier literature (see, e.g., [12]), is an important characteristic of SP algorithms.

The structure of this paper is outlined as follows. Aiming at a full generality, in Section II, we present a generic formulation of CSPs and explain the SP algorithms using 3-COL problems. In Section III, we introduce the “probabilistic token passing” algorithm as a unification and alternative formulation of SP for arbitrary CSPs. We then show in Section IV how BP may be reduced to SP on a properly defined Forney graph, where state decoupling and a sufficient condition on the problem structure are presented. Finally we conclude the paper in Section V, where we attempt to answer the central question of this paper: “is SP BP?”
II. CONSTRAINT-SATISFACTION PROBLEMS AND SP

Let $V$ be a finite set indexing a set of variables $\{x_v : v \in V\}$, where each variable $x_v$ takes on values from some set $\chi_v$. For any subset $U \subseteq V$, we will use $x_U$ to denote the variable set $\{x_v : v \in U\}$. We note that depending on the context, $x_U$ may also be interpreted as a configuration in Cartesian product $\chi_U := \prod_{v \in U} \chi_v$.

Let $C$ be another finite set indexing a set of constraints $\{c : c \in C\}$, the form of which will be specified subsequently. For each $c \in C$, let $V(c)$ be some subset of $V$, indexing the set of all variables constrained by $\Gamma_c$. Symmetrically, for each $v \in V$, we will denote the set $\{c : v \in V(c)\}$ by $C(v)$, namely, $C(v)$ indexes the set of all constraints involving variable $x_v$. Each constraint $\Gamma_c$ applies only on variables $x_{V(c)}$, and we will identify constraint $\Gamma_c$ as a subset of the Cartesian product $\chi_{V(c)}$. Thus a constraint-satisfaction problem (CSP) may be taken so as a subset of the Cartesian product $\chi_V = \prod_{v \in V} \chi_v$. Each constraint $\Gamma_c$ connects vertices $x_v \in \chi_v$ indexed by $V$, and we will denote the set $\chi_c \subseteq \prod_{v \in V} \chi_v$ indexed by $C$.

Thus a constraint-satisfaction problem (CSP) may be specified by $(V, C, \{\chi_v : v \in V\}, \{V(c) : c \in C\}, \{\Gamma_c : c \in C\})$, with the objective of finding a solution for equation

$$\prod_{c \in C} [x_{V(c)} \in \Gamma_c] = 1. \quad (1)$$

Here the notation $[P]$ for any Boolean proposition $P$, is the Iverson’s convention [5], namely, evaluating to 1 if $P$ is true, and to 0 otherwise. Clearly, (1) can be represented by a factor graph [5], with variable vertices indexed by $V$ and function vertices indexed by $C$.

Using (1), the graph-coloring problem, or $q$-COL problem, on an undirected graph $(\Delta, \Xi)$ with vertex set $\Delta$ and edge set $\Xi$ (where each edge in $\Xi$ connecting vertices $a$ and $b$ in $\Delta$ is identified with set $\{a, b\}$) is defined by $V := \Delta$, $C := \Xi$, $\chi := \{1, 2, \ldots, q\}$, $V(c) := c$, $\Gamma_c := \chi_{V(c)} \setminus \{(r, r) : r = 1, 2, \ldots, q\}$.

Figure 1(b) shows the factor-graph representation of a $q$-COL problem on the undirected graph shown in Figure 1(a).

![Figure 1](image_url)

**Fig. 1.** (a) An undirected graph. (b) The factor graph for a $q$-COL problem on graph (a).

SP has been developed for several classes of CSPs as a message-passing algorithm on the factor graph representing the problem. Previous expositions of SP, such as [7], [11], have emphasized the role of a special symbol, namely the “joker” symbol (often denoted by “#” or “*”). In $k$-SAT problems, $x_v$ equal to the joker indicates that it is free to take any value in the alphabet, and that $x_v$ equal to a non-joker symbol indicates that it is constrained to taking the designated value. It is shown that in hard $k$-SAT problems, the “joker” symbol connects the satisfying configurations, which would otherwise form a large number of disconnected “clusters”, making local search strategies fail (see, e.g., [7]).

As in next section, we will present an alternative formulation of SP and revise the interpretation of the joker symbol, here we give a description of SP for 3-COL problems as an example of the SP algorithm, following the exposition in [2], where we will make little effort interpreting the update equations. The reader is referred to [12] for a generic formulation of SP, although we feel that the token-passing formulation presented [10] and reviewed in next section is more easily accessible.

For 3-COL problems, it is worth noting that each constraint vertex has degree 2. This allows the combination of the message passed from variable $x_v$ to a neighboring constraint, say $\Gamma_c$, with the message passed from constraint $\Gamma_c$ to the other neighbor, say $x_{v'}$, of $\Gamma_c$. As a consequence, $\Gamma_c$ will be suppressed in the factor graph, and messages are directly passed between variable vertices that are distance 2 apart (or equivalently, messages are passed on graph $(\Delta, \Xi)$). Following [2], a compact version of SP message-passing rule is given as follows, where the message passed from variable $x_{u}$ to variable $x_v$ is a quadruplet of real numbers $(\eta_{u \rightarrow v}^1, \eta_{u \rightarrow v}^2, \eta_{u \rightarrow v}^3, \eta_{u \rightarrow v}^*).

$$\eta_{u \rightarrow v}^r = \prod_{w \in N(u) \setminus \{v\}} (1 - \eta_{u \rightarrow w}^r) - \sum_{p \neq r} \sum_{w \in N(u) \setminus \{v\}} (\eta_{u \rightarrow w}^p \eta_{w \rightarrow u}^r + \eta_{w \rightarrow u}^r) + \prod_{w \in N(u) \setminus \{v\}} \eta_{w \rightarrow u}^r \quad (2)$$

for every $r \in \{1, 2, 3\}$, where $N(u)$ is the set $\{v : v \in V, \{u, v\} \in \Xi\}$, and $\sum_p$ is the short form of $\sum_{p=1,2,3}$, and

$$\eta_{u \rightarrow v}^* = 1 - \sum_p \eta_{u \rightarrow v}^p. \quad (3)$$

The SP messages are usually initialized randomly. Upon convergence, SP computes a summary message, a quadruplet $(\mu_1, \mu_2, \mu_3, \mu_*)$ of real numbers, at each variable node $x_v$, according to the following rule.

$$\mu_r = \sum_{u \in N(v)} \prod_{p \neq r} \prod_{w \in N(u) \setminus \{v\}} (\eta_{u \rightarrow w}^p \eta_{w \rightarrow u}^r + \eta_{w \rightarrow u}^r) + \prod_{w \in N(u) \setminus \{v\}} \eta_{w \rightarrow u}^r \quad (4)$$

for every $r \in \{1, 2, 3\}$, and

$$\mu_* = 1 - \sum_p \mu_r^p. \quad (5)$$

We note that the summary message at variable $x_v$ may be interpreted respectively as the probability or “bias” of each symbol in $\{1, 2, 3, *\}$.

Usually, SP is combined with a **decimation procedure**. In the decimation procedure of 3-COL problems, a variable is
fixed to a color \( r \in \{1,2,3\} \) if it is highly “biased” to this color. The 3-COL problem is then simplified and SP is applied again. This process iterates until the reduced problem is simple enough for a local search algorithm.

III. SP AS PROBABILISTIC TOKEN PASSING

For each variable \( x_v \) in a given problem, we define an extended alphabet \( \chi_v^+ \) as the power set of \( \chi_v \) (i.e., \( \chi_v^+ = \{ a : a \subseteq \chi_v \} \)). For 3-COL problems, \( \chi_v^+ \) is then the set \( \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\} \} \) for every \( v \).

For each non-empty element \( t \) of \( \chi_v^+ \), we also write it as a string containing the elements of \( t \). For example, we may write \( \{1,2\} \) as 12. For any subset \( U \subseteq V \), a configuration \( y_U := \{ y_v \in \chi_v^+ : v \in U \} \) is referred to as a rectangle on \( \Gamma \) and understood as the Cartesian product \( \prod_{v \in U} y_v \). Given rectangle \( y_U \) and some \( v \in U \), we refer to its \( v \)-component \( y_v \) as the \( v \)-side of \( y_U \). Clearly, if any component \( y_v \) in the Cartesian product is \( \emptyset \), then rectangle \( y_U \) is \( \emptyset \). We denote by \( \chi_v^U \) the set of all rectangles on \( \Gamma \), and clearly \( \chi_v^U = \prod_{v \in U} \chi_v^+ \).

For any \( U \) and \( S \) with \( S \subseteq U \subseteq V \) and any \( \Omega \subseteq \chi_v^U \), we denote \( \Omega_S \) := \{ \{ \alpha, \gamma \} \in \Omega : \exists \beta \in \chi_v \} \) as the projection of \( \Omega \) on \( S \).

Given an index \( v \in V \) and a configuration \( y_{V \backslash \{v\}} \), we define \( \mathcal{F}_v(y_{V \backslash \{v\}}) := \{ (y_{V \backslash \{v\}} \times \chi_v \} \cap \Gamma_v \) as the forced \( v \)-side by rectangle \( y_{V \backslash \{v\}} \) and constraint \( \Gamma_v \). That is, \( \mathcal{F}_v(y_{V \backslash \{v\}}) \) is the largest subset of \( \chi_v \) in which every element, when paired with some sequence in \( y_{V \backslash \{v\}} \), makes constraint \( \Gamma_v \) satisfied.

We now define the deterministic token passing procedure on the factor-graph representation of the CSP as follows. Tokens are passed along the edges of the factor graph and the token passed from and to each variable \( x_v \) is an element of \( \chi_v^+ \). For a pair of neighboring vertices \( x_v \) and \( \Gamma_v \), on the factor graph, the token \( t_{v \rightarrow c} \) passed from variable \( x_v \) to constraint \( \Gamma_v \) depends on all incoming tokens passed to \( x_v \), except that passed from \( \Gamma_v \). Similarly, the token \( t_{c \rightarrow v} \) passed from constraint \( \Gamma_v \) to variable \( x_v \) depends on all incoming tokens passed to \( \Gamma_v \), except that passed from \( x_v \). The token passing rules are given as follows.

\[
t_{v \rightarrow c} = \bigcap_{b \in \mathcal{F}_v \\Gamma_v} t_{b \rightarrow v}, \tag{4}
\]

\[
t_{c \rightarrow v} = \mathcal{F}_v \left( \prod_{u \in \mathcal{F}_c \{v\}} t_{u \rightarrow c} \right). \tag{5}
\]

Examples of token passing for a 3-COL problem are illustrated in Figure 2.

A “summary token”, \( v \)-side \( y_v \), may be computed for each \( v \in V \), by

\[
y_v = \bigcap_{b \in \mathcal{F}_v} t_{b \rightarrow v}.
\]

and these \( v \)-sides \( \{ y_v : v \in V \} \) form a rectangle \( y_V := \prod_{v \in V} y_v \).

Summary token at each \( v \) is evaluated during token passing iterations so as to check convergence or to claim the \( v \)-side of the rectangle \( y_V \). If the factor graph is a tree, it is possible to show that deterministic token passing always converges. Additionally, on such factor graphs, it is possible to interpret the claimed rectangle \( y_V \) upon convergence in certain meaningful way. Here we however choose not to elaborate on this, since CSPs described by a cycle-free factor graph can be solved using local search algorithms, making SP-style algorithms useless.

The deterministic token passing procedure can be simply extended to the probabilistic token passing (PTP) algorithm on the same factor graph, where token \( t \) passed along each edge is treated as a random variable not allowed to be \( \emptyset \). That is, in PTP, instead of passing token \( t \) on an edge, the passed message is the distribution of \( t \) conditioned on \( t \neq \emptyset \). Specifically, it is assumed that 1) all tokens passed to a given vertex are independent; 2) an outgoing token is generated from the incoming tokens according to the functional dependency of the outgoing token on the incoming tokens given in (4) and (5). 3) the distribution of the random token on any edge (conditioned that it is not the empty set) is used as the message passed along the edge.

More precisely, the PTP message-passing rule is given as follows. We will use \( \lambda_{c \rightarrow v} \) to denote the message passed from a variable \( x_v \) to a constraint \( \Gamma_c \), and use \( \rho_{v \rightarrow c} \) to denote the message passed from a constraint \( \Gamma_c \) to a variable \( x_v \).

\[
\lambda_{c \rightarrow v}(t_{c \rightarrow v}) = \sum_{t_{v \rightarrow c} \in \mathcal{F}_v \\Gamma_c} \left( t_{v \rightarrow c} = \bigcap_{b \in \mathcal{F}_v \\Gamma_c} t_{b \rightarrow v} \right) \cdot \prod_{u \in \mathcal{F}_v \\Gamma_c} \rho_{v \rightarrow u}(t_{v \rightarrow u}), \tag{6}
\]

\[
\rho_{v \rightarrow c}(t_{v \rightarrow c}) = \sum_{t_{c \rightarrow v} \in \mathcal{F}_c \{v\}} \left( t_{c \rightarrow v} = \mathcal{F}_v \left( \prod_{u \in \mathcal{F}_c \{v\}} t_{u \rightarrow c} \right) \right) \cdot \prod_{u \in \mathcal{F}_c \{v\}} \lambda_{u \rightarrow c}(t_{u \rightarrow c}). \tag{7}
\]

We note that in (6) and (7), \( t_{c \rightarrow v} \) and \( t_{v \rightarrow c} \) range over all elements of \( \chi_v^+ \) except \( \emptyset \), and proper normalization is required to make the messages distributions.

Parallel to the deterministic token passing procedure, in PTP, the “summary message” \( \mu_v(y_v) \) at each \( v \), which is a distribution of \( v \)-side, may be computed as

\[
\mu_v(y_v) = \sum_{t_{c \rightarrow v} \in \mathcal{F}_c \{v\}} \left( y_v = \bigcap_{c \in \mathcal{F}_v} t_{c \rightarrow v} \right) \cdot \prod_{c \in \mathcal{F}_v} \rho_{v \rightarrow c}(t_{v \rightarrow c}).
\]
On 3-COL problems, PTP, like SP, can be made more compact. However, instead of passing messages between variable vertices, the PTP messages more naturally reduce to messages passed between constraint vertices that are distance 2 apart. Note that for any two constraint vertices \( \Gamma_c \) and \( \Gamma_d \), there is a unique variable vertex \( x_v \), for which vertices \( \Gamma_c \), \( x_v \), and \( \Gamma_d \) form a path of length 2 from \( \Gamma_c \) to \( \Gamma_d \). We then use \( \mathcal{M}(c, d) \) to denote the index, \( v \), of the unique variable \( x_v \) between \( \Gamma_c \) and \( \Gamma_d \) on the path. We now denote the message passed from constraint \( \Gamma_c \) to variable \( x_{\mathcal{M}(c, d)} \) by \( \rho_{c \rightarrow d} \). Then PTP message update rules for 3-COL problems can be completely described, as in the following lemma, by the update of \( \rho_{c \rightarrow d} \) for every pair of constraint vertices \( \Gamma_c \) and \( \Gamma_d \) that are distance 2 apart.

**Lemma 1:** The support of \( \rho_{c \rightarrow d} \) is \( \{12, 13, 23, 123\} \), and when using \( \{i, j, k\} \) to represent the three distinct elements of \( \{1, 2, 3\} \)

\[
\rho_{c \rightarrow d}(ij) = Z \left( \prod_{b \in N(c) \setminus \{d\}} (p_{b \rightarrow c}(ik) + p_{b \rightarrow c}(jk) + p_{b \rightarrow c}(123)) \right) - \prod_{b \in N(c) \setminus \{d\}} (p_{b \rightarrow c}(ik) + p_{b \rightarrow c}(123)) - \prod_{b \in N(c) \setminus \{d\}} (p_{b \rightarrow c}(jk) + p_{b \rightarrow c}(123) + \prod_{b \in N(c) \setminus \{d\}} p_{b \rightarrow c}(123)) \right),
\]

and

\[
\rho_{c \rightarrow d}(123) = Z \left( \prod_{b \in N(c) \setminus \{d\}} (p_{b \rightarrow c}(12) + p_{b \rightarrow c}(123)) + \prod_{b \in N(c) \setminus \{d\}} (p_{b \rightarrow c}(13) + p_{b \rightarrow c}(123)) + \prod_{b \in N(c) \setminus \{d\}} (p_{b \rightarrow c}(23) + p_{b \rightarrow c}(123)) - 2 \times \prod_{b \in N(c) \setminus \{d\}} p_{b \rightarrow c}(123) \right),
\]

where \( N(c) = \{b : c \cap b \neq \emptyset, b \in \Xi\} \), and \( Z \) is a normalization constant such that \( \sum_{t} \rho_{c \rightarrow d}(t) = 1 \).

Matching the equations in this lemma with (2) and (3), the equivalence between SP and PTP rules for 3-COL problems is evident, as formulated in the following theorem.

**Theorem 1:** The correspondence between update equations (8) and (9) and equations (2) and (3) is: \( \rho_{c \rightarrow d}(ij) = \eta_{u \rightarrow v} \) and \( \rho_{c \rightarrow d}(123) = \eta_{u \rightarrow v}^* \), where \( v = \mathcal{M}(c, d) \) and \( u \) is the other neighbor of \( c \) besides \( \mathcal{M}(c, d) \).

We note that on \( k \)-SAT problems the equivalence between SP and PTP can also be shown similarly. That is, one may view PTP as a simple alternative formulation of SP, generally applicable to arbitrary CSPs. It is worth noting that PTP formulated above can be made in precise correspondence with the generic formulation of SP presented in [12]. But in PTP, the extension of the variable alphabets to their power sets is made more explicit, which we think is an important aspect of SP. In light of this perspective, the role of the joker symbol should be de-emphasized. That is, the joker symbol is merely one of the many subsets of a variable alphabet, namely, the original variable alphabet.

**IV. THE REDUCTION OF BP TO SP**

In the style of [7], [8], an MRF can be defined for arbitrary CSPs as follows. For each edge \( (x_v, \Gamma_c) \) on the factor-graph representation, a state variable \( s_{v,c} \), taking on values in \( \chi_v^c \times \chi_c^* \), is introduced. Each state variable \( s_{v,c} \) will also be written as an ordered pair \( (s_v^L, v) \), where the first component is referred to as the left state, and the second component is referred to as the right state. For any \( v \in V \) and any subset \( C' \subseteq \mathcal{C}(v) \), we use \( s_{v,C'} \) to denote the variable set \( \{s_{v,c} : c \in C'\} \). Similarly, for any \( c \in C \) and any subset \( V' \subseteq V(c) \), we use \( s_{V',c} \) to denote the variable set \( \{s_{v,c} : v \in V'\} \). In addition, we will use \( s_{V,C} \) to denote the set of all state variables. Similar notations apply to left and right states. Instead of on \( \chi_V \), the MRF is defined on \( \chi_V^* \), where each variable \( x_v \) is replaced by an extended variable \( y_v \), taking values in \( \chi_v^* \). The local potential functions of the MRF consist of left functions — each for a \( v \in V \) and denoted by \( g_v \) — and a set of right functions — each for a \( c \in C \) and denoted by \( f_c \). The left function \( g_v \) for each \( v \in V \) is defined as

\[
g_v(y_v, s_{v,C(v)}) := [y_v = \bigcup_{c \in \mathcal{C}(v)} s_v^R] \cdot [y_v \neq \emptyset] \cdot \prod_{c \in \mathcal{C}(v)} [s_v^c = y_v].
\]

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

\[
f_c(s_{V(c),c}) := \prod_{v \in \mathcal{V}(c)} [s_v^c = F_c(s_{V(c) \setminus \{v\}, c})].
\]

The global function \( F \) is defined by

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

Then function \( F \), upon normalization, is an MRF, readily expressed as a special kind of factor graph — a normal realization or Forney graph [13]. In the Forney graph, each \( y_v \) is suppressed to a “half edge”; each state variable, having degree 2, is suppressed to an edge, and all vertices on the graph represent functions. The Forney graph representing the normal realization of the \( q \)-COL problem in Figure 1 is shown in Figure 3.

Such an MRF formalism, in the context of \( k \)-SAT problems, is equivalent to the one presented in [7]. With this formalism,
we show in [8] that the BP message-passing rule reduces to
SP for k-SAT problems when an appropriate initialization of
messages — similar to that of [7] — is imposed. However, we
observe that for 3-COL problems, such a formalism does
not reduce to SP as naturally [9].

We now define a modified BP message-passing rule on such
a Forney graph, which we refer to as the state-decoupled
belief propagation (SD-BP). The purpose of introducing this
“new” message-passing rule is to arrive at a unified reduction
mechanism for SP to reduce from BP (or more precisely from
SD-BP).

Identical to BP at local function vertices, SD-BP differs
from BP in that messages passed from the right functions
need an additional processing, which we call state
decoupling, before they are passed to the left functions. More precisely,
there are three kinds of messages: right message $\rho_{c \rightarrow v}$ is computed at local function $f_c$ to pass along the edge to
gc; state-decoupled right message $\rho^{*}_{c \rightarrow v}$ is computed at the
dge connecting $f_c$ and gc, which depends only on the right message $\rho_{c \rightarrow v}$ on the same edge and is passed to left function
gc; left message $\lambda_{v \rightarrow c}$ is computed at the left function gc
to pass along the edge connecting to $f_c$. We note that the state-
decoupled right message $\rho^{*}_{c \rightarrow v}$ is a function on the support of $\rho_{c \rightarrow v}$ defined by

$$
\rho^{*}_{c \rightarrow v}(s^{L}_{v,c}, s^{R}_{c \rightarrow v}) := \rho_{c \rightarrow v}(s^{R}_{c \rightarrow v}(s^{L}_{v,c})).
$$

(11)

An intuitive explanation of state decoupling is to result in that
the (state-decoupled) right messages only depend on their right states.

Then the SD-BP message-passing rule can be written as the
following updates of left messages and state-decoupled right
messages:

$$
\lambda_{v \rightarrow c}(s^{L}_{v,c}, s^{R}_{c \rightarrow v}) = 
\sum_{s^{L}_{v,c}(e), (e)} \left[ \left( s^{L}_{v,c} = \bigcap_{b \in C(e) \setminus \{c\}} s^{R}_{b \rightarrow v} \right) \cdot \prod_{b \in C(e) \setminus \{c\}} \rho^{*}_{b \rightarrow v}(s^{L}_{v,c}, s^{R}_{b \rightarrow v}) \right]
$$

(12)

$$
\rho^{*}_{c \rightarrow v}(s^{L}_{v,c}, s^{R}_{c \rightarrow v}) = 
\sum_{s^{L}_{v,c}(e), (e)} \left[ \left( s^{L}_{v,c} = F_{c}(s^{L}_{v,c}(e)) \right) \cdot \prod_{u \in V(e) \setminus \{v\}} \lambda_{u \rightarrow c}(s^{L}_{u,c}, F_{c}(s^{R}_{v,c}(e), s^{L}_{u,c})) \right]
\times \prod_{u \in V(e) \setminus \{v\}} \lambda_{u \rightarrow c}(s^{L}_{u,c}, F_{c}(s^{R}_{v,c}(e), s^{L}_{u,c}))
$$

(13)

We now introduce a sufficient condition under which SD-
BP messages reduce to PTP or equivalently to SP. To simplify
expressions, for each constraint $C$ and $v \in V(e)$, we denote by $A_{c}(v)$ the set of all possible forced $v$-sides by a rectangle
$V(e) \setminus \{v\}$ and $G_{e}$. That is,

$$
A_{c}(v) := \{F_{c}(y_{V(e) \setminus \{v\}}) : y_{V(e) \setminus \{v\}} \in \chi_{V(e) \setminus \{v\}} \}.
$$

Theorem 2: Suppose that a CSP satisfies the following
condition: for any $c \in C$, $v \in V(e)$, $u \in V(e) \setminus \{v\}$,

$$
a_{v,c} \in A_{c}(v), \text{ and } y_{V(e) \setminus \{u,v\}} \in \chi_{V(e) \setminus \{u,v\}}.
$$

Then the correspondence of SD-BP message-passing rules (12)
and (13) and PTP message-passing rules (6) and (7) is

$$
\rho^{*}_{c \rightarrow v}(a_{v,c}, a_{v,c}) = \rho_{c \rightarrow v}(PTP)(a_{v,c}),
$$

for every $(a_{v,c}, a_{v,c})$ in the support of $\rho^{*}_{c \rightarrow v}$.

Proof: We can first merge SD-BP message-passing rules
(12) and (13) into one, which is shown as follow.

$$
\rho^{*}_{c \rightarrow v}(s^{L}_{v,c}, s^{R}_{c \rightarrow v}) = 
\sum_{s^{L}_{v,c}(e), (e)} \left[ \left( s^{L}_{v,c} = F_{c}(s^{L}_{v,c}(e)) \right) \cdot \prod_{u \in V(e) \setminus \{v\}} \lambda_{u \rightarrow c}(s^{L}_{u,c}, F_{c}(s^{R}_{v,c}(e), s^{L}_{u,c})) \right]
\times \prod_{u \in V(e) \setminus \{v\}} \lambda_{u \rightarrow c}(s^{L}_{u,c}, F_{c}(s^{R}_{v,c}(e), s^{L}_{u,c}))
$$

(14)

$$
\sum_{t_{u \rightarrow c} = \bigcap_{b \in C(u) \setminus \{c\}} t_{b \rightarrow u}} \left[ t_{u \rightarrow c} = \bigcap_{b \in C(u) \setminus \{c\}} t_{b \rightarrow u} \right] \prod_{b \in C(u) \setminus \{c\}} \rho_{b \rightarrow u}(PTP)(t_{b \rightarrow u})
$$

(15)

where $Q = F_{c}(s^{L}_{v,c}, s^{L}_{V(e) \setminus \{u,v\}, c}) \cap \bigcap_{b \in C(u) \setminus \{c\}} s^{R}_{b \rightarrow u}$.

Similarly, we can merge PTP message-passing rules (6) and
(7) as follow.

$$
\rho_{c \rightarrow v}(PTP)(t_{c \rightarrow v}) = 
\sum_{t_{c \rightarrow v} = \bigcap_{v \in V(e)} t_{v \rightarrow c}} \left[ t_{c \rightarrow v} = \bigcap_{v \in V(e)} t_{v \rightarrow c} \right] \prod_{u \in V(e) \setminus \{v\}} \rho_{c \rightarrow v}(PTP)(t_{v \rightarrow c})
$$

(16)

$$
\sum_{t_{u \rightarrow c} = \bigcap_{b \in C(u) \setminus \{c\}} t_{b \rightarrow u}} \left[ t_{u \rightarrow c} = \bigcap_{b \in C(u) \setminus \{c\}} t_{b \rightarrow u} \right] \prod_{b \in C(u) \setminus \{c\}} \rho_{b \rightarrow u}(PTP)(t_{b \rightarrow u})
$$

(17)

Since for any $c \in C$ and $\{u,v\} \subseteq V(e)$, we have

$$
F_{c}(a_{v,c}, y_{V(e) \setminus \{u,v\}}) \supseteq \bigcap_{b \in C(u) \setminus \{c\}} a_{u,b},
$$

then in SD-BP message-passing rule (16), we have

$$
Q = F_{c}(a_{v,c}, s^{L}_{V(e) \setminus \{u,v\}, c}) \cap \bigcap_{b \in C(u) \setminus \{c\}} s^{R}_{b \rightarrow u}
$$

=} \bigcap_{b \in C(u) \setminus \{c\}} s^{R}_{b \rightarrow u}.
$$

(18)
Clearly, under this condition, (16) and (17) are identical when $s_{v,c}^L$ and $s_{v,c}^R$ in (16) are identified with $t_{v,c}$ and $t_{v,c}$ in (17). The equivalence $\rho^*_{e\rightarrow d}(a_{v,c},a_{v,c}) = \rho_{e\rightarrow d}^{(PTP)}(a_{v,c})$ is then proved.

This theorem essentially presents a sufficient condition for SD-BP to reduce to PTP or SP in arbitrary CSPs. We conjecture that this condition is also necessary for general CSPs in the sense that for any weaker condition it is possible to construct a CSP satisfying the condition but disallowing SD-BP to reduce to PTP.

Now it is easy to verify that for 3-COL problems, the condition of Theorem 2 is satisfied. As a corollary, SD-BP for 3-COL problems reduces to PTP or SP. We now elaborate on this result without using Theorem 2.

For 3-COL problems, similar to PTP, the messages of SD-BP can be compactly represented using state-decoupled right messages passed between every pair of right function vertices, say $f_e$ and $f_a$, that are distance 2 apart. That is, letting $\rho^*_{e\rightarrow d} := \rho^*_{e\rightarrow d|N(c)}$, we may write the update rules only in terms of $\rho^*_{e\rightarrow d}$. Additionally, we note that $\rho^*_{e\rightarrow d}(s)$ is completely specified by $\rho^*_{e\rightarrow d}(s)$ with state $s$ taking the form of $(i,j,i)$ or $(123, 123)$.

**Lemma 2:** For 3-COL problem with above defined MRF and state-decoupling operation, the update rule of message $\rho^*_{e\rightarrow d}$ is:

\[
\rho^*_{e\rightarrow d}(i,j,i) = \prod_{b \in N(c) \setminus \{d\}} (\rho^*_{b\rightarrow c}(ik,ik) + \rho^*_{b\rightarrow c}(jk,jk) + \rho^*_{b\rightarrow c}(123, 123)) - \prod_{b \in N(c) \setminus \{d\}} (\rho^*_{b\rightarrow c}(ik,ik) + \rho^*_{b\rightarrow c}(123, 123))
\]

and

\[
\rho^*_{e\rightarrow d}(123, 123) = \prod_{b \in N(c) \setminus \{d\}} (\rho^*_{b\rightarrow c}(12, 12) + \rho^*_{b\rightarrow c}(123, 123))
\]

where $N(c) = \{b : V(b) \cap V(c) \neq \emptyset, b \in C\}$.

The equivalence of SD-BP messages to PTP (and hence to SP) messages can be characterized in the following theorem.

**Theorem 3:** The correspondence between PTP update equations (8) and (9) and SD-BP update equations (18) and (19) is

\[
\rho^*_{e\rightarrow d}(i,j,i) = J \rho_{e\rightarrow d}(i,j) \text{ and } \rho^*_{e\rightarrow d}(123, 123) = J \rho_{e\rightarrow d}(123),
\]

for some normalization constant $J$.

We remark that for $k$-SAT problems, it is possible to show that the same state-decoupling operation also reduces SD-BP to PTP or SP. However, in that case, such an operation need not be carried out explicitly, since its effect can be equivalently induced by a proper initialization condition — the peculiar condition in [7] or [8] that appeared intriguing prior to this work. On the other hand, for 3-COL problems, the state-decoupling operation need to be enforced, without which BP does not seem to reduce to PTP or SP.

### V. Conclusion

In this paper, we present a simple formulation of SP for general CSPs in terms of “probabilistic token passing”, where we stress the role of extending variable alphabets. Furthermore, we show that for SP to reduce from BP under an MRF formalism in the style of [7], or [8], imposing an extra “state-decoupling” operation on the BP messages is needed. In addition, the reduction requires certain condition on the structure of the CSP. Accidentally, the previously studied $k$-SAT and 3-COL problems both have this condition satisfied.

Although it may still be too early to conclude that SP is not BP, the evidence presented in this paper may to a large extent suggest that SP is different from BP at least in the sense that for general CSPs, there may not exist an MRF formalism under which BP is SP. — The MRF formalism in the style of [7] and [8] fails to achieve this purpose.

Finally, given the complication in the connection between SP and BP presented in this paper, one may wonder whether a change of perspective in studying SP is necessary. That is, instead of asking “is SP BP?”, the right question may be: “what is SP?” or “what is the right graphical model underlying SP?”

### References

[1] M. Mézard, G. Parisi, and R. Zecchina, “Analytic and algorithmic solution of random satisfiability problems,” Science, vol. 297, pp. 812–821, 2002.

[2] A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, “Polynomial iterative algorithms for coloring and analyzing random graphs,” Physical Review E, vol. 68, no. 3, p. 036702, 2003.

[3] M. J. Wainwright and E. Maneva, “Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes,” in Proc. IEEE Int. Symp. Inform. Theory, Adelaide, Australia, 2005, pp. 1493–1497.

[4] W. Yu and M. Aksakalli, “Coding for the blackwell channel: a survey propagation approach,” in Proc. IEEE Int. Symp. Inform. Theory, Adelaide, Australia, 2005, pp. 1583–1587.

[5] 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.

[6] A. Braunstein and R. Zecchina, “Survey propagation as local equilibrium equations,” J. Stat. Mech., June 2004.

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

[8] R. Tu, Y. Mao, and J. Zhao, “On generalized survey propagation: normal realization and sum-product interpretation," in Proc. IEEE Int. Symp. Inform. Theory, 2006.

[9] R. Tu, Y. Mao, and J. Zhao, “On the interpretation of survey propagation,” in Proc. 10th Canadian Workshop on Inform. Theory, 2007, pp. 77–80.

[10] R. Tu, Y. Mao, and J. Zhao, “Survey propagation as “probabilistic token passing”," IEICE Transactions on Information and Systems, Special Section on Foundations of Computer Science.

[11] A. Braunstein, Mézard, and R. Zecchina, “Survey propagation: An algorithm for satisfiability," Random Structures and Algorithms, vol. 27, no. 2, pp. 201–226, 2005.

[12] A. Braunstein, M. Mézard, M. Weigt, and R. Zecchina, Constraint satisfaction by survey propagation, Sep. 2003, http://arxiv.org/abs/cond-mat/0212451.

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