Is SP BP?

Ronghui Tu, **Yongyi Mao** and **Jiying Zhao**

School of Information Technology and Engineering

University of Ottawa

800 King Edward Avenue

Ottawa, Ontario, K1N 6N5, Canada

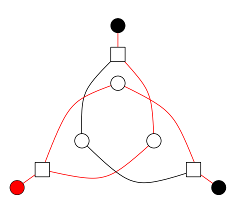
Email: {rtu, yymao, jyzhao}@site.uottawa.ca

Abstract

The Survey Propagation (SP) algorithm for solving k-SAT problems has been shown recently as an instance of the Belief Propagation (BP) algorithm. In this paper, we show that for general constraint-satisfaction problems, SP may not be reducible from BP. We also establish the conditions under which such a reduction is possible. Along our development, we present a unification of the existing SP algorithms in terms of a probabilistically interpretable iterative procedure — weighted Probabilistic Token Passing.

Index Terms

Survey Propagation, Belief Propagation, constraint satisfaction, Markov random field, factor graph, message-passing algorithm, k-SAT, q-COL



I. Introduction

Survey Propagation (SP) [1] is a recent algorithmic breakthrough in solving certain hard families of constraint satisfaction problems (CSPs). Derived from statistical physics, SP first demonstrated its power in solving classic prototypical NP-complete problems, the k-SAT problems [2]. — For random instances of these problems in the hard regime, SP is shown to be the first efficient solver [1]. Recently, SP has also been applied to other CSPs, including other NP-complete problem families such as graph coloring (or q-COL) problems [3], as well as problems arising in communications and data compressions, some examples being coding for Blackwell channels [4] and quantization of Bernoulli sequences [5]. In all these cases, great successes have been demonstrated.

Powerful as it appears, SP however largely remains as a heuristic algorithm to date, where analytic understanding of its algorithmic nature and rigorous characterization of its performance are widely open and of great curiosity and research importance.

Similar to the well-known Belief Propagation (BP) algorithm used in iterative decoding [6] and statistical inference [7], SP operates by iteratively passing "messages" in a factor graph representation [8] of the problem instance, where each variable vertex corresponds to a variable whose value is to be decided and each function vertex corresponds to a local constraint imposed on the variables. This observation has inspired a recent research effort in understanding whether SP may be viewed as a special case of BP. — The significance of questions of such a kind has been witnessed repeatedly in the history of communication research, for example, in understanding the Viterbi algorithm as a dynamic programming algorithm [9], in understanding the turbo decoding algorithm [10] as an instance of Belief Propagation [11], and in unifying the BCJR algorithm [12] and the Viterbi algorithm under the umbrella of the generalized distributive law [13], etc. These unified frameworks have on one hand provided additional insights into the nature of the algorithms, and on the other hand allowed an easier access of the algorithm by much wider research communities. Specific to the question "is SP BP", if SP may be understood as an instance of BP, then the existing analytic techniques of BP are readily applicable to analyzing SP; if SP can not be characterized as a special case of BP, one is then motivated to seek a different algorithmic framework to which SP belongs or to discover the unique algorithmic nature of SP.

The first result reporting that SP is an instance of BP is the work of [14] in the context of k-

SAT problems. This result is generalized in [15] to an extended version of SP for solving k-SAT problems. Briefly, the authors of [15] present a Markov Random Field (MRF) [16] formalism for k-SAT problems; a parameter, denoted by γ in this paper, is used to parametrize the MRF. When the BP algorithm is derived on such an MRF, the BP message-update equations result in a *family* of SP algorithms, referred to as *weighted SP* or SP(γ) in this paper, parametrized by $\gamma \in [0, 1]$; and when $\gamma = 1$, SP(γ) is the original (non-weighted) SP. In addition to extending SP — in the context of k-SAT problems — to a family of SP algorithms with tunable performance, another significance of this result is a conclusive answer to the titular question in that context, namely that SP is BP for the k-SAT problem family. This result was re-developed in our earlier work [17] where a simpler MRF formalism using Forney graphs [18] is presented and a more transparent reduction of BP messages to weighted SP messages is given.

The objective of this paper is to answer the question whether SP and more generally weighted SP are special cases of BP for arbitrary CSPs beyond k-SAT problems. It is worth noting that weighted SP has only been presented for k-SAT problems, although its principle may be extended to designing other CSPs involving *binary* variables (see, e.g., [5]). Furthermore, resulting from BP on a properly defined MRF, weighted SP, unlike the original (non-weighted) SP, does not have a probabilistic interpretation that *does not* rely on the MRF constructed in the style of [15] or [17] and the derived BP algorithm thereby. Thus to answer the question whether weighted SP is BP for general CSPs, it is necessary to formulate weighted SP for arbitrary CSPs that generalizes non-weighted SP without relying on any MRF and BP formalism. For this reason, this research and hence the structure of this paper roughly split into two parts. The first part answers the question what SP and weighted SP exactly are by presenting a probabilistically interpretable formulation of both non-weighted and weighted SP for arbitrary CSPs. The second part presents a MRF formalism for general CSPs in the style of [15] or [17], derives the BP update equations, and answers the question whether and how BP under such MRF formalism may be reduced to SP, if at all.

Although this paper focuses on the second part, namely, on answering whether SP algorithms are instances of BP on a properly defined MRF, our effort in establishing what SP algorithms are and how to formulate these algorithms for general CSPs is noteworthy.

First, the notion of weighted SP, as noted earlier, has only been presented for k-SAT problems as in [15] and in sporadic example applications involving only *binary* variables such as in [5].

As will become clear in this paper, the design philosophy of weighted SP for CSPs involving binary variables (such as in [15] and [5]) is not readily extendable to arbitrary CSPs with arbitrary variable alphabets, since an important notion underlying SP, namely, an *appropriate* extension of variable alphabets, is blurred in the binary special cases.

Second, for non-weighted SP, we note that its formulation in the context of general CSPs primarily exists in the literature of statistical physics (see, e.g., [19]). Although its design recipe has been laid out for arbitrary CSPs, its exposition in statistical physics language has made it rather difficult for readers with primarily engineering or computer science background.

Thus, in addition to serving as the basis for the investigation of BP-to-SP reduction, the first part of the paper also aims at providing a clean, transparent and easily accessible formulation of SP algorithms in its most general form for arbitrary CSPs, without resorting to statistical physics concepts.

II. Main Results and Paper Organization

The main results of this paper are summarized as follows.

In the first part, we formulate SP and weighted SP for general CSPs as what we call "probabilistic token passing" (PTP) and "weighted probabilistic token passing" (weighted PTP) respectively, where a message is a distribution (or non-negative function) on the set of "tokens" associated with a variable. Here a "token" is a non-empty *subset* of the variable's alphabet[1]. It has been previously observed in SP applied to various problems that a "joker" symbol is added to the original variable alphabet. Here we point out that extending the alphabet by simply adding a joker symbol is not sufficient for general CSPs, particularly for those involving non-binary variables. We stress that the *right* extension of the variable alphabet is to replace it with the set of all non-empty subsets of the original alphabet. Although an equivalent treatment has been described in some previous literature for non-weighted SP [19], this perspective is for the first time made explicit beyond statistical physics context and for both non-weighted and weighted

[1] In fact more rigorously, a token is a non-empty *subset* of all possible *assignments* of a variable – In this paper, for more mathematical rigor and clarity, we make a distinction between the alphabet of a variable and the set of all assignments to the variable, where an assignment to variable x_v is treated as a function mapping the singleton set $\{v\}$ to the alphabet of x_v. Nevertheless, one may always identify the set of all assignments to x_v with the alphabet of x_v via a one-to-one correspondence and loosely refer to the set of all assignments of a variable as the alphabet of the variable.

SP. Based on this notion of alphabet extension, we generalize weighted SP for arbitrary CSPs in the form of weighted PTP. In other words, the weighted PTP formulation presented in this paper serves as a recipe for designing weighted SP algorithm for arbitrary CSPs.

In the second part, we present an MRF formalism — which we refer to as “normally realized MRF” — for arbitrary CSPs using Forney graphs, generalizing the MRF construction in the style of [15] and [17] presented for k-SAT problems. States, each consisting of a left state and a right state, are introduced in the MRF, where the left state corresponds to the token passed from the variable and the right state corresponds to the token passed from the constraint. For any given CSP, the MRF is parametrized by a collection of weighting functions, each corresponding to a variable in the CSP; in the k-SAT special case, these weighting functions may reduce to a single parameter, γ. Noting the combinatorial importance of such MRF in the context of k-SAT problems [15], one expects that this general formulation of MRF for arbitrary CSP may serve a similar role, namely providing a combinatorial framework describing the topology of the solution space [15]. This direction, clearly deserving further investigation, is however out of the scope of this paper.

On the normally realized MRF formalism, we then proceed to derive the BP update equations and investigate the reduction of BP to weighted PTP (noting that weighted PTP *is* weighted SP and that non-weighted SP is a special case of weighted SP). Primarily re-developing the results of [15] and [17] on BP-to-SP reduction, we show that for k-SAT problems, BP is readily reducible to weighted PTP as long as a condition — which we refer to as the *state-decoupling condition* — is imposed on the BP messages in initialization. An interesting fact about this condition in the context of k-SAT problems is that as long as the condition is satisfied in the first BP iteration, it will continue to be satisfied in all iterations after. This forms the basis on which BP messages may be simplified to the form of weighted PTP messages. This condition, also arising in [15] and [17] as a peculiar and curious construction, had not been explained prior to this work. In this paper, we argue that the state-decoupling condition serves a critical role in the reduction of the weighted PTP messages from the BP messages derived from the MRF formalism in the style of [15] and [17], or from the normally realized MRF presented in this paper. Using the example of 3-COL problems, we show that such a condition is also needed in all BP iterations so as for BP to reduce to PTP. However, in that case, we show that this condition can not be made satisfied in every BP iteration (except for the trivial cases in which the BP messages contain no useful information) and one must manually impose this condition by manipulating the BP messages in each iteration. This result on one hand justifies the important role of the state-decoupling condition in the reduction of BP to PTP and on the other hand asserts that BP is *not* PTP and hence *not* SP for 3-COL problems!

At that point, one is ready to conclude that weighted PTP or weighted SP is not a special case of BP for general CSPs. The manual manipulation of BP messages in 3-COL problems, which results in what we call *state-decoupled BP* brings up a further question, namely, for general CSPs, whether PTP and weighted PTP are readily expressed as state-decoupled BP. We proceed to show that for general CSPs, the reduction of weighted PTP from BP requires yet another condition pertaining to the structure of the CSP. Briefly, this additional condition demands that the constraints in the CSP be "locally compatible" with each other in some sense. We show that the local compatibility condition of the CSP is the necessary and sufficient condition for state-decoupled BP to reduce to weighted PTP or weighted SP. At that end, we complete the answer to the titular question "is SP BP?".

As mentioned earlier, in addition to answering whether SP is BP, another objective of this paper is to explain SP as simply as possible. For this purpose, we have made an effort in presenting this paper in a pedagogical manner and carrying along the examples of k-SAT and 3-COL problems throughout the paper.

The remainder of this paper is organized as follows. In Section III, we present a generic formulation of CSPs while also introducing various notations that will be used in later parts of the paper. In Section IV, we introduce the existing SP algorithms using the examples of k-SAT problems and 3-COL problems, where we purposefully avoid SP formulations in statistical physics languages. We then proceed in Section V to present a general formulation of SP algorithms in terms of PTP and weighted PTP. In Section VI, we present the normally realized MRF formalism and present results concerning the reduction of BP messages to SP messages. At this time, how SP algorithms behave over iterations and how they solve a CSP are important open problems. Although such questions are not of particular importance for the purpose of this paper, completely ignoring them appears not satisfactory to us and perhaps also to some readers. For this reason, we present some preliminary results along those lines for understanding the dynamics of PTP. — These results are included in the Appendix so as to maintain the focus of this paper. The paper is briefly concluded in Section VII.

III. A Generic Formulation of Constraint Satisfaction Problems

Let V be a finite set, in which each element will be referred to as a *coordinate*. Associated with each $v \in V$, there is a finite *alphabet* χ_v. For each $v \in V$, we will assume throughout of this paper that every χ_v is identical to each other, and is therefore denoted by χ. We note that this slight loss of generality is made only for lightening the upcoming notations, and that there is no difficulty to extend the results of this paper to more general cases where χ_v's are different from each other. For any subset $U \subseteq V$, a *χ-assignment* x_U on U is a function mapping U into the set χ. That is, a χ-assignment x_U specifies a way to assign each coordinate $u \in U$ a value in χ. The set of all χ-assignments on U will be denoted by χ^U. When U is a singleton set $\{u\}$, which contains a single coordinate u, we will call χ-assignment $x_{\{u\}}$ on $\{u\}$ an *elementary (χ-)assignment* and write it as x_u for simplicity. Clearly, any given elementary χ-assignment x_u is uniquely specified by a value $r \in \chi$, which is the assigned value in χ to coordinate u. In this case, this assignment is denoted by r_u, for example, if $\chi := \{0, 1\}$, then the only possible χ-assignments on $\{u\}$ are 0_u and 1_u, which are the elementary assignments assigning 0 and 1 to coordinate u, respectively.

Suppose that $U \subset W \subseteq V$ and that x_W is a χ-assignment on W. We will use $x_{W:U}$ to denote the (function) restriction of x_W on U. For any subset of χ-assignments $\Omega \subseteq \chi^W$ on W, we denote the projection of Ω on U by $\Omega_{:U}$. That is,

$$\Omega_{:U} := \{x_{W:U} : x_W \in \Omega\}.$$

If coordinate set U can be partitioned into disjoint subsets A and B, then it is obvious that assignment x_U decomposes into assignments $x_{U:A}$ and $x_{U:B}$, and x_U may be written as $(x_{U:A}, x_{U:B})$ (in any order). Evidently, x_U may be decomposed according to any partition of U, not necessarily two-fold partitions. In particular, if a collection of sets $\{U_i : i \in \mathcal{I}\}$, for some $\mathcal{I}$, form a partition of U, then we may assignment x_U as $\langle x_{U:U_i}\rangle_{i\in\mathcal{I}}$.

For simplicity, we will write (x_A, x_B) and $\langle x_{U_i}\rangle_{i\in\mathcal{I}}$ in place of $(x_{U:A}, x_{U:B})$ and $\langle x_{U:U_i}\rangle_{i\in\mathcal{I}}$ respectively. In fact, unless some particular clarity is needed, we will always write $x_{W:U}$ simply as x_U, making the underlying x_W implicit. Furthermore, when U is a singleton set $\{u\}$, as mentioned earlier, we will simply denote it by x_u, which reduces to the conventional "variable" notation standard literatures of graphical models.

Given χ and V, the objective of a constraint satisfaction problem (CSP) is to find a global χ-assignment x_V that satisfies a given set of constraints or to conclude that no such assignment exists. Formally, we will use set C to index the set of constraints $\{\Gamma_c : c \in C\}$. Each constraint Γ_c, $c \in C$, applies to a subset of the coordinates V, which will be denoted by $V(c)$. Specifically, each constraint Γ_c is identified with a subset of $\chi^{V(c)}$, and the constraint is satisfied by global χ-assignment x_V if $x_{V:V(c)} \in \Gamma_c$. Then any CSP may be formulated via specifying V, C, χ, $\{V(c) : c \in C\}$ and $\{\Gamma_c : c \in C\}$, where the objective of the CSP is to find a χ-assignment x_V such that

$$\prod_{c \in C} [x_{V:V(c)} \in \Gamma_c] = 1, \tag{1}$$

or to conclude that no such assignment exists. Here the notation $[P]$, for any Boolean proposition P, is the Iverson's convention [8], namely, evaluating to 1 if P, and to 0 otherwise.

Now it is easy to verify that the factorization structure of (1) can be represented by a factor graph [8]: in the factor graph, "variable vertices" are indexed by V, where the "variable" indexed by $v \in V$ represents an elementary assignment $x_{V:\{v\}}$ on $\{v\}$, or simply x_v; "function vertices" are indexed by C, where the function indexed by $c \in C$ is $[x_{V:V(c)} \in \Gamma_c]$, which, with a slight overloading of notation, will also be denoted by $\Gamma_c(x_{V(c)})$; there is an edge connecting variable vertex x_v with function vertex Γ_c if and only if $v \in V(c)$. Inspired by its correspondence (to an edge) in the factor graph, we will use $(v - c)$ to denote a coordinate-constraint pair (v, c) where coordinate v is involved in constraint Γ_c in the CSP.

For notational symmetry, we denote the set $\{c : v \in V(c)\}$ by $C(v)$, namely, $C(v)$ indexes the set of all constraints involving coordinate v, or the set of all function vertices connecting to variable vertex x_v. We will assume that $|C(v)| \geq 2$ for all $v \in V$. It is clear that such an assumption is without loss of generality, since if a variable x_v is involved in only one constraint, one may always modify the constraint and remove the variable from the problem. Similarly, we will assume that $|V(c)| \geq 2$ for every $c \in C$. This is also without loss of generality since if a constraint Γ_c only involves a single variable x_v, it is always possible to "absorb" this constraint in other constraints involving x_v (noting that x_v must have another constraint since $|C(v)| \geq 2|$).

A. k-SAT

The k-SAT problems are a classic family of CSPs, known to be NP-complete for $k \geq 3$ [2]. An instance of k-SAT problems consists of a set of variables $\{x_v : v \in V\}$, each of which takes on values from the set $\chi := \{0, 1\}$, and a set of constraints $\{\Gamma_c : c \in C\}$, each of which involves exactly k variables. For each constraint Γ_c and every $v \in V(c)$, there is a value $L_{v,c} \in \{0, 1\}$ which we will refer to as the *preferred value* on v in constraint Γ_c. The k-SAT problem is then to decide on an assignment x_V such that for each constraint Γ_c, at least one of its involved coordinate is assigned its preferred value in Γ_c. To map back to the afore-mentioned set-theoretic formulation of constraints, in a k-SAT problem, for each $c \in C$, let l^c denote the χ-assignment on $V(c)$ in which every coordinate $v \in V(c)$ is assigned the negated value $\bar{L}_{v,c}$ of its preferred value $L_{v,c}$ in Γ_c, namely that $l^c_{:\{v\}} = \bar{L}_{v,c}$ for every $(v - c)$, then constraint Γ_c is defined as $\Gamma_c := \chi^{V(c)} \setminus \{l^c\}$.

The factor-graph representation of a toy 3-SAT problem is shown in Fig. 1. For k-SAT problems, it is convenient to treat each preferred value $L_{v,c}$ as the label for edge (x_v, Γ_c) on the factor graph, and use dashed edge to represent label 0 and solid edge to represent label 1.

We note that it is customary in this paper that variable vertices in a factor graph are listed on the left side and function (constraint) vertices listed on the right side.

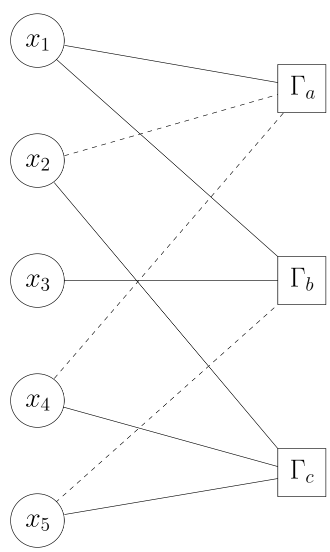


Fig. 1. A factor graph for 3-SAT problem specified by formula $(x_1 \vee \overline{x}_2 \vee \overline{x}_4) \wedge (x_1 \vee x_3 \vee \overline{x}_5) \wedge (x_2 \vee x_4 \vee x_5)$. Logic operation notations are used here to define the problem, where $\vee$ denotes logic `OR`, $\wedge$ denotes logic `AND`, and the horizontal bar on a variable denotes the negation of the variable. The function represented by the factor graph is $[(x_1, x_2, x_4) \in \Gamma_a] \cdot [(x_1, x_3, x_5) \in \Gamma_b] \cdot [(x_2, x_4, x_5) \in \Gamma_c]$, where $\Gamma_a = \chi^{\{1,2,4\}} \setminus \{(0_1, 1_2, 1_4)\}$, $\Gamma_b = \chi^{\{1,3,5\}} \setminus \{(0_1, 0_3, 1_5)\}$, and $\Gamma_c = \chi^{\{2,4,5\}} \setminus \{(0_2, 0_4, 0_5)\}$.

B. Graph Coloring

Graph coloring or q-COL problems are another family of NP-complete problems. Given an undirected graph (Δ, Ξ) with vertex set Δ and edge set Ξ, the objective of the q-COL problem on (Δ, Ξ) is to assign each vertex in Δ a color from q different colors such that every pair of adjacent vertices have different colors. To use the above generic formulation of CSPs, we will denote the set of all q colors by set $\chi := \{1, 2, \ldots, q\}$. We will denote every undirected edge in Ξ, say the edge connecting vertices u and v, by set $\{u, v\}$. The set V of all coordinates is then identified with set Δ, and the set C indexing all constraints is identified with Ξ. Specifically note that every $c \in C$ is then identified with some $\{u, v\} \in \Xi$, and $V(c)$ is identified with c, or the corresponding set $\{u, v\}$. Suppose that $c = \{u, v\} \in \Xi$, then constraint Γ_c is identified with $\chi^{\{u,v\}} \setminus \{(1_u, 1_v), (2_u, 2_v), \ldots, (q_u, q_v)\}$. Fig. 2(b) shows the factor-graph representation of a q-COL problem on the undirected graph shown in Fig. 2(a).

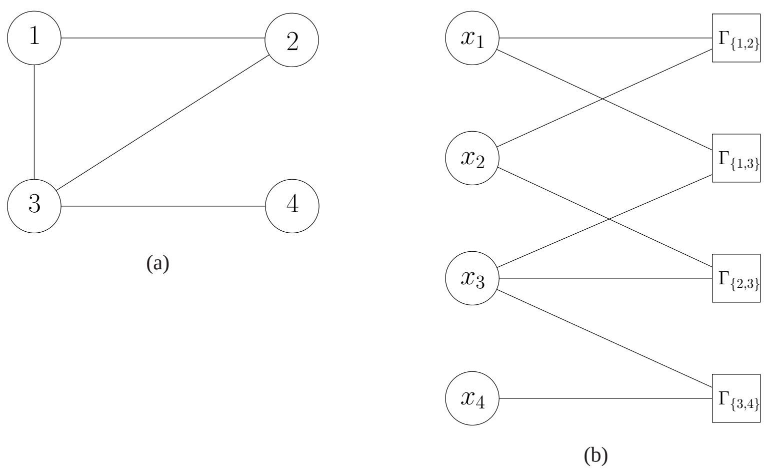


Fig. 2. (a) An undirected graph. (b) The factor graph for a q-COL problem on graph (a). The global function represented by the factor graph is $[(x_1, x_2) \in \Gamma_{\{1,2\}}] \cdot [(x_1, x_3) \in \Gamma_{\{1,3\}}] \cdot [(x_2, x_3) \in \Gamma_{\{2,3\}}] \cdot [(x_3, x_4) \in \Gamma_{\{3,4\}}]$, where $\Gamma_{\{u,v\}} := \chi^{\{u,v\}} \setminus \{(1_u, 1_v), (2_u, 2_v), \ldots, (q_u, q_v)\}$.

IV. Survey Propagation Algorithms

A. Survey Propagation for k-SAT Problems

Extensive study has been carried out to understand the hardness of k-SAT problems (for $k \geq 3$) and to develop efficient solvers. A parameter $\alpha := |C|/|V|$ is observed to be critically related to the hardness of random k-SAT problems. There appear two thresholds of α, denoted by α_d and α_c, ($\alpha_d < \alpha_c$), marking two "phase transitions" [1]. When $\alpha > \alpha_c$, random k-SAT problems are unsatisfiable (i.e., having no satisfying assignment) with high probability; when $\alpha_d < \alpha < \alpha_c$, the satisfying assignments form exponentially many disjoint "clusters", making the problem extremely difficult; when $\alpha < \alpha_d$, the satisfying assignments merge into one huge cluster and problems are easier. In the regime of $\alpha < \alpha_d$, local search algorithms, such as BP, may find a satisfying assignment. In the regime of $\alpha_d < \alpha < \alpha_c$, local search algorithms usually fail.

The discovery and first application of survey propagation (SP) are in solving the k-SAT problems in the hard regime, where messages are passed on the above-defined factor graphs [1]. In SP, a "joker" symbol "$*$" is introduced to variable alphabet χ of the k-SAT problem, where x_v equal to the "joker" indicates that it is free to take any value from its original alphabet, and that x_v equals a non-joker symbol indicates that it is constrained to taking the designated value. Briefly, SP on k-SAT problems may be viewed as an iterative method for estimating the "biases" of each variable x_v on $0, 1$ and $*$ respectively and a variable that is highly biased on 0 or 1 can be fixed to that value whereby simplifying the problem. It is shown that in the hard regime of random k-SAT problems, the "joker" symbol connects the disconnected clusters, making SP remain very effective even for α very close to α_c [15]. For k-SAT problems, the original version of SP [1] is generalized in [15] to what we call the *weighted SP* [2] or SP(γ) in this paper. SP(γ) is a family of algorithms parametrized by a real number $\gamma \in [0, 1]$, where SP(1) is the original SP and for some judicious choice of $\gamma \in (0, 1)$, SP(γ) may have further improved performance.

We note that generalizing SP to the family of weighted SP algorithms has only been reported for k-SAT problems to date, and one of the objectives of this paper is to extend such a generalization to arbitrary CSPs.

Similar to BP, in the SP algorithms, messages are passed between variable vertices and function vertices. For the purpose of describing the SP message-update rule for k-SAT problems, we introduce the following notations. For any $(v - c)$, $C_c^{\mathtt{u}}(v)$ 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 $\{b \in C(v) \setminus \{c\} : L_{v,b} = L_{v,c}\}$.

[2] In [15], weighted SP is referred to as generalized SP. In this paper, we would like to reserve the term "generalized SP" to refer to SP algorithms generalized for arbitrary CSPs beyond k-SAT problems.

Following [15], the message-update rule of SP(γ) is described as follows.

The message passed from variable vertex x_v to function vertex Γ_c — also referred as a *left message* — is a triplet of real numbers $(\Pi^{\mathtt{u}}_{v\to c}, \Pi^{\mathtt{s}}_{v\to c}, \Pi^{*}_{v\to c})$, and the message passed from function vertex Γ_c to variable vertex x_v — also referred to as a *right message* — is a real number $\eta_{c\to v} \in [0,1]$. These messages are updated respectively according to the following equations.

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

$$\Pi^{\mathtt{s}}_{v\to c} := \left(1- \prod_{b\in C^{\mathtt{s}}_c(v)} (1-\eta_{b\to v})\right) \prod_{b\in C^{\mathtt{u}}_c(v)} (1-\eta_{b\to v}) \tag{3}$$

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

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

The initialization of SP messages is usually random, and message-passing schedule is typically similar to the *flooding schedule* [8] in BP message passing, namely, that each iteration may be defined by all variable vertices passing messages followed by all function vertices passing messages. We note that throughout this paper all message-passing schedules are restricted to the flooding schedule for convenience, where each iteration is defined as first updating all "left messages" and then updating all "right messages" [3]

Similar to BP, at the end of an iteration, SP may compute a "summary message" at each variable vertex. For any $v \in V$, define $C^1(v) := \{b \in C(v) : L_{v,b} = 1\}$ and $C^0(v) := \{b \in C(v) : L_{v,b} = 0\}$, then the "summary message" at x_v is a triplet $(\zeta^1_v, \zeta^0_v, \zeta^*_v)$ of real numbers, computed by

[3]An iteration may also include updating all summary messages after updating the right messages; see the description of summary messages.

$$\zeta_v^1 := \left(1 - \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v}) \tag{6}$$

$$\zeta_v^0 := \left(1 - \gamma \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \tag{7}$$

$$\zeta_v^* := \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v}) \tag{8}$$

where summary message $(\zeta^1, \zeta^0, \zeta^*)$ is typically normalized to a scaled version $(\zeta^{1\text{norm}}, \zeta^{0\text{norm}}, \zeta^{*\text{norm}})$ such that

$$\zeta^{1\text{norm}} + \zeta^{0\text{norm}} + \zeta^{*\text{norm}} = 1.$$

Equations (2) to (8) and the normalization procedure after completely specify the message-update rule of SP(γ).

Usually, SP is applied in conjunction with a heuristic "decimation" procedure, which is carried out after SP converges or after a certain number of SP iterations. In the decimation procedure, the "polarity" $B(v) := \zeta_v^{0\text{norm}} - \zeta_v^{1\text{norm}}$ at each $v \in V$ is calculated, and the most polarized variable (namely, one having the highest $|B(v)|$) is fixed to 0 or 1 according to the sign of $B(v)$: x_v is set to 0 if $B(v) > 0$, and to 1 otherwise. The k-SAT problem is then simplified and SP is applied again. This process iterates until the reduced problem is simple enough for a local search algorithm.

When $\gamma = 1$, it is shown in [19] and [15] that the passed messages as in (2) through (5) can be interpreted probabilistically, namely, $\eta_{c \to v}$ may be interpreted as the probability that a "warning" symbol is sent from Γ_c to x_v, and $\Pi^{\mathtt{u}}_{v \to c}$, $\Pi^{\mathtt{s}}_{v \to c}$ and $\Pi^{*}_{v \to c}$ are respectively the probabilities that x_v sends to Γ_c symbol $\bar{L}_{v,c}$, symbol $L_{v,c}$ and symbol $*$.

When $\gamma < 1$, SP(γ) however can no longer be interpreted probabilistically. We now present a slightly modified formulation of SP(γ), referred to as SP$^*(\gamma)$, which is completely equivalent to SP(γ) defined in [15], and which will be shown in a later section to have a natural probabilistic interpretation.

In SP$^*(\gamma)$, the left message $(\Pi^{\mathtt{u}}_{v \to c}, \Pi^{\mathtt{s}}_{v \to c}, \Pi^{*}_{v \to c})$ passed from variable vertex x_v to function vertex Γ_c is modified to the equations given in (9) to (11), and the right message $\eta_{c \to v}$ passed from function vertex Γ_c to variable vertex x_v and the summary message $(\zeta_v^1, \zeta_v^0, \zeta_v^*)$ at variable x_v stay unchanged.

$$\Pi^{\mathtt{u}}_{v\to c} := \left(1 - \gamma \prod_{b\in C_c^{\mathtt{u}}(v)} (1-\eta_{b\to v})\right) \prod_{b\in C_c^{\mathtt{s}}(v)} (1-\eta_{b\to v}) \tag{9}$$

$$\Pi^{\mathtt{s}}_{v\to c} := \left(1 - \gamma \prod_{b\in C_c^{\mathtt{s}}(v)} (1-\eta_{b\to v})\right) \prod_{b\in C_c^{\mathtt{u}}(v)} (1-\eta_{b\to v}) \tag{10}$$

$$\Pi^{*}_{v\to c} := \gamma \prod_{b\in C_c^{\mathtt{s}}(v)} (1-\eta_{b\to v}) \prod_{b\in C_c^{\mathtt{u}}(v)} (1-\eta_{b\to v}) \tag{11}$$

The following lemma shows that SP(γ) and SP$^*(\gamma)$ are equivalent.

Lemma 1: For the same initialization of $\{\eta_{c\to v} : \forall (v-c)\}$, at any given iteration, SP$^*(\gamma)$ and SP(γ) give rise to identical results in $\eta_{c\to v}$ for every $(v-c)$, and in $(\zeta_v^1, \zeta_v^0, \zeta_v^*)$ for every $v \in V$.

Proof: The lemma follows from that in the computation of $\eta_{c\to v}$ and hence of $(\zeta_v^1, \zeta_v^0, \zeta_v^*)$, $\Pi^{\mathtt{s}}_{v\to c}$ and $\Pi^{*}_{v\to c}$ always appear together in the form of $\Pi^{\mathtt{s}}_{v\to c} + \Pi^{*}_{v\to c}$. But it is easy to see that in SP(γ) and in SP$^*(\gamma)$, $\Pi^{\mathtt{s}}_{v\to c} + \Pi^{*}_{v\to c}$ has the same parametric form, both equal to $\prod_{b\in C_c^{\mathtt{u}}(v)}(1 - \eta_{b\to v})$. ■

We conclude this subsection by remarking that it is possible to verify that all results concerning SP(γ) in [15] hold for SP$^*(\gamma)$ [4]. As such, in the rest of this paper, SP$^*(\gamma)$ rather than SP(γ) will be taken as the weighted SP for k-SAT problems.

B. Survey Propagation for q-COL Problems

Similar to SP developed for k-SAT problems, in q-COL problems, SP passes messages between the variable vertices and the function (constraint) vertices in the factor-graph representation of the problem. Some notable differences however exist.

First, weighted SP has not been developed for q-COL problems to date, and it is not even clear whether such algorithm family, if existing, can be developed in a similar manner as that for k-SAT in [15], namely, via reducing the BP algorithm derived from a properly defined MRF. Answering this question in a later section, we here therefore only review the original version of

[4]Specifically, we note that BP on the MRF formulated in [15] will also reduce to SP$^*(\gamma)$. We leave this for the interested readers to verify.

SP applied to 3-COL problems following the formulation in [3], which is analogous to SP(1), or the non-weighted SP, in the context of k-SAT.

Second, the SP messages for q-COL problems can be expressed more compactly, due to a specific nature of the problem, on which we now elaborate.

For q-COL problems, each constraint vertex has degree 2. This allows the combination of the message passed from variable x_u to a neighboring constraint, say Γ_c, with the message passed from constraint Γ_c to the other neighbor, say x_v, of Γ_c. As a consequence, Γ_c may be suppressed in the factor graph, and messages are directly passed between variable vertices that are distance 2 apart [5] (or equivalently, messages are passed on graph (Δ, Ξ)). Following [3], a compact version of SP message-passing rule for 3-COL problems is given as follows, where the message passed from variable x_u to variable x_v is a quadruplet of real numbers $(\eta^1_{u\to v}, \eta^2_{u\to v}, \eta^3_{u\to v}, \eta^*_{u\to v})$. For $i = 1, 2, 3$,

$$\eta^i_{u\to v} := \frac{\prod\limits_{w\in N(u)\setminus\{v\}} (1-\eta^i_{w\to u}) - \sum\limits_{j\neq i} \prod\limits_{w\in N(u)\setminus\{v\}} (\eta^*_{w\to u} + \eta^j_{w\to u}) + \prod\limits_{w\in N(u)\setminus\{v\}} \eta^*_{w\to u}}{\sum\limits_{j=1,2,3} \prod\limits_{w\in N(u)\setminus\{v\}} (1-\eta^j_{w\to u}) - \sum\limits_{j=1,2,3} \prod\limits_{w\in N(u)\setminus\{v\}} (\eta^*_{w\to u} + \eta^j_{w\to u}) + \prod\limits_{w\in N(u)\setminus\{v\}} \eta^*_{w\to u}} \tag{12}$$

where $N(u)$ is the set $\{v : v \in V, \{u, v\} \in \Xi\}$, namely, the set of neighboring vertices of vertex u on graph $\{\Delta, \Xi\}$; and

$$\eta^*_{u\to v} := 1 - \sum_{j=1,2,3} \eta^j_{u\to v}. \tag{13}$$

For 3-COL problems, the "summary message" computed at each variable vertex x_v is a quadruplet of real numbers, denoted by $(\zeta^1_v, \zeta^2_v, \zeta^3_v, \zeta^*_v)$, where for $i = 1, 2, 3$,

$$\zeta^i_v := \frac{\prod\limits_{u\in N(v)} (1-\eta^i_{u\to v}) - \sum\limits_{j\neq i} \prod\limits_{u\in N(v)} (\eta^*_{u\to v} + \eta^j_{u\to v}) + \prod\limits_{u\in N(v)} \eta^*_{u\to v}}{\sum\limits_{j=1,2,3} \prod\limits_{u\in N(v)} (1-\eta^j_{u\to v}) - \sum\limits_{j=1,2,3} \prod\limits_{u\in N(v)} (\eta^*_{u\to v} + \eta^j_{u\to v}) + \prod\limits_{u\in N(v)} \eta^*_{u\to v}}$$

and

$$\zeta^*_v := 1 - \sum_{j=1,2,3} \zeta^j_v.$$

Similar to that for k-SAT problems, the summary message for a 3-COL problem at variable x_v may indicate the "bias" of variable x_v to each letter in $\{1, 2, 3, *\}$. In the decimation procedure

[5] Still implementing the flooding schedule, the SP message-update rule for 3-COL problems however suppresses the passing of one set of messages (say, for example, the right messages) by including the computation of these messages in updating the other set of messages.

for 3-COL problems – carried out in a similar way to that for k-SAT problems, a variable is fixed to a color $i \in \{1, 2, 3\}$ if it is highly biased to that color. The reader is referred to [3] for a detailed account of a heuristic decimation rule used in solving 3-COL problems using SP.

We note that this paper primarily focuses on SP update equations, where the decimation aspect of SP is largely ignored.

V. SP AS PROBABILISTIC TOKEN PASSING

To date, SP algorithms have been applied to various other CSPs, for example, in coding for Blackwell channels [4], in quantization of Bernoulli sources [5], and in solving graph coloring problems [3], etc.. However, a general formulation of SP, particularly that of weighted SP, for solving arbitrary non-binary CSPs, has been largely missing. Specifically, we note the following milestones in the formulation of SP algorithms.

- The work of [19] presents non-weighted version of SP formulas for general CSPs beyond those involving only binary variables. However, the exposition of [19] uses the language of statistical physics, rather remote to the engineering community, and a cleaner and more friendly formulation of SP, and particularly of weighted SP, is desirable for general problems.
- The work of [15] presents weighted SP for k-SAT problems, in which weighted SP is treated as a special case of BP in a properly defined MRF. This treatment of SP and the corresponding principle for developing weighted SP are conceivably applicable to all binary CSPs. However, it has remained open, prior to this work, whether such an approach to understanding and developing weighted SP is applicable to arbitrary non-binary CSPs.

The line of development in this section is summarized below.

We will first present an understanding of non-weighted SP for arbitrary CSPs (namely, that formulated in [19]) in terms of "probabilistic token passing (PTP)". Although similar understanding has been previously reported in various contexts, we here stress the role of extending the variable alphabet in SP algorithms, and explicitly point out that the alphabet extension is *not* to simply include an extra joker symbol, but to *replace* the variable alphabet with its *power set* (excluding the empty-set element). To make the PTP procedure more intuitively sensible, prior to defining PTP, we will introduce a precursor of PTP, which we call "deterministic token passing" (DTP).

After introducing PTP, we then show that the probabilistic interpretation of non-weighted SP in terms of PTP makes it naturally generalizable to a weighted version, which we call weighted PTP. For a brief preview, the generalization of PTP to weighted PTP essentially involves generalizing a *functional dependency* in PTP message-update rule to a *probabilistic dependency*. Interestingly as we will show, it turns out that for k-SAT problems, weighted PTP precisely coincides with weighted SP of [15]. This should convincingly demonstrate that weighted PTP is a generalization of weighted SP for arbitrary CSPs.

The outline of this section is given as follows. Subsection V-A introduces the notion of alphabet extension and related concepts. Subsection V-B defines DTP as a precursor of PTP. In Subsection V-C, we introduce PTP. In Subsection V-D, we show that PTP is equivalent to SP, using 3-COL problem as an example. In Subsection V-E, we introduce weighted PTP. In Subsection V-F, we show that weighted PTP generalize weighted SP using k-SAT problems as an example.

A. Alphabet Extension

For a given CSP with variable alphabet χ, we define the *extended alphabet* χ^* as the power set of χ excluding the empty set $\emptyset$. That is, $\chi^* = \{t : t \subseteq \chi, t \neq \emptyset\}$). The extended alphabet χ^* of k-SAT problems is then the set $\{\{0\}, \{1\}, \{0, 1\}\}$. For 3-COL problems, χ^* is the set $\{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. Each element t of χ^* will be written as a string – in bold font – containing the elements of t. For example, we may write $\{1, 2\}$ as $\mathbf{12}$, $\{1, 2, 3\}$ as $\mathbf{123}$ and $\{1\}$ simply as $\mathbf{1}$.

Given any subset $U \subseteq V$, a χ^*-assignment y_U on U is referred to as a *rectangle* on U. The set of all rectangles on U is denoted by $(\chi^*)^U$. Given rectangle $y_U \in (\chi^*)^U$, for every $v \in U$, $y_{U:\{v\}}$, or simply written as y_v — following an earlier convention of this paper — is referred to as the v-*side* of y_U. Apparently, rectangle y_U has $|U|$ sides, and may also be written as the concatenation of all its sides, namely, as $\langle y_v \rangle_{v \in U}$.

For any $v \in V$, an elementary χ^*-assignment $t_v \in (\chi^*)^{\{v\}}$ will be referred to as a *token* on v. Using this nomenclature, the v-side of any rectangle is a token on v. We note that a token t_v may be interpreted as a set of elementary χ-assignments on $\{v\}$, which is in fact the set of all elementary χ-assignments on $\{v\}$ that assign v a value in set $t_v(v) \subseteq \chi$. For example, suppose that $\chi := \{1, 2, 3\}$, then token $\mathbf{12}_v$ may be identified with the set $\{1_v, 2_v\}$ of elementary χ-assignments on $\{v\}$.

It is worth noting that when a token t_v is identified with a set of elementary χ-assignments on v, a rectangle $\langle t_v \rangle_{v \in U}$ may be identified with the *Cartesian product* of all its sides. For example, rectangle $(\mathbf{12}_v, \mathbf{23}_u)$ may be interpreted as the following set of χ-assignments on $\{v, u\}$: $\{(1_v, 2_u), (1_v, 3_u), (2_v, 2_u), (2_v, 3_u)\}$. Under this interpretation, we will also make frequent uses of the Cartesian product notation, writing rectangle $(\mathbf{12}_v, \mathbf{23}_u)$ as $\mathbf{12}_v \times \mathbf{23}_u$, and rectangle $\langle t_v \rangle_{v \in U}$ as $\prod_{v \in U} t_v$. We note that this interpretation is in fact the reason for which we choose the terminologies "rectangle" and "side".

For simplicity, from here on, we shall reserve the term "assignment" to referring to a χ-assignment only, and a χ^*-assignment will be referred to as a "rectangle", "side" or "token".

We say that an assignment x_U on U is *contained* in rectangle y_U if $x_{U:\{v\}}(v) \in y_{U:\{v\}}(v)$ for every $v \in U$. For example, assignment $(1_v, 2_u)$ is contained in rectangle $(\mathbf{13}_v, \mathbf{23}_u)$ We will use $x_U \in y_U$ to denote this containedness relationship, since this notation is precise when the rectangle y_U is interpreted as a *set* of assignments on U.

Given a CSP and a $(v - c)$ pair, we define function $\mathtt{F}_c^v : (\chi^*)^{V(c)\setminus\{v\}} \to (\chi^*)^{\{v\}}$ as follows: for every rectangle $\prod_{u \in V(c)\setminus\{v\}} t_u$ on $V(c) \setminus \{v\}$,

$$\mathtt{F}_c^v \left(\prod_{u \in V(c)\setminus\{v\}} t_u \right) := \left(\left(\chi^{\{v\}} \times \prod_{u \in V(c)\setminus\{v\}} t_u \right) \cap \Gamma_c \right)_{:\{v\}}.$$

We often write $\mathtt{F}_c^v$ in short as $\mathtt{F}_c$ since the domain and co-domain of the function may be recovered from the form of its argument. Given rectangle $\prod_{u \in V(c)\setminus\{v\}} t_u$ on $V(c) \setminus \{v\}$, we call $\mathtt{F}_c \left(\prod_{u \in V(c)\setminus\{v\}} t_u \right)$ the *forced token* by rectangle $\prod_{u \in V(c)\setminus\{v\}} t_u$ via constraint Γ_c. It is easy to verify that the forced token $\mathtt{F}_c \left(\prod_{u \in V(c)\setminus\{v\}} t_u \right)$ is simply the set of all (elementary) assignments on $\{v\}$ which, when concatenated with an assignment on $V(c) \setminus \{v\}$ contained in rectangle $\prod_{u \in V(c)\setminus\{v\}} t_u$, make local constraint Γ_c satisfied. We now give some examples using the toy 3-SAT problem shown in Fig. 1 to illustrate this definition. Consider constraint Γ_a, if rectangle $t_{\{1,2\}}$ on $\{1, 2\}$ is defined as $(\mathbf{1}_1, \mathbf{01}_2)$, then forced token $\mathtt{F}_a(t_{\{1,2\}}) = \mathbf{01}_4$, since when assigning variable x_4 either value 0 or 1, it is possible to find an assignment of variables x_1 and x_2 in rectangle $t_{\{1,2\}}$ that makes Γ_a satisfied; on the other hand, if $t_{\{1,2\}} = (\mathbf{0}_1, \mathbf{1}_2)$, then forced token $\mathtt{F}_a(t_{\{1,2\}}) = \mathbf{0}_4$, since rectangle $t_{\{1,2\}}$ contains a single assignment of x_1 and x_2 (namely $(0_1, 1_2)$), and the only assignment of x_4 that will make constraint Γ_a satisfied is the one assigning 0 to x_4, namely 0_4.

A "monotonicity property" of function $\mathtt{F}_c$, stated in the following lemma, follows immediately from the definition of the function.

Lemma 2: Suppose that x_v and Γ_c are a pair of neighboring variable and constraint vertices in the factor graph, and that $y_{V(c)\setminus\{v\}}$ and $y'_{V(c)\setminus\{v\}}$ are two rectangles on $V(c)\setminus\{v\}$. Then $y_{V(c)\setminus\{v\}} \subset y'_{V(c)\setminus\{v\}}$ impliese that $\mathtt{F}_c\left(y_{V(c)\setminus\{v\}}\right) \subseteq \mathtt{F}_c\left(y'_{V(c)\setminus\{v\}}\right)$.

B. Deterministic Token Passing (DTP)

As we will introduce — for arbitrary CSPs — a probabilistic interpretation of non-weighted SP (namely, PTP) and generalize it to a weighted version (namely, weighted PTP), in this subsection, we first introduce an algorithmic procedure, which we call *deterministic token passing* or DTP. We note that the purpose of introducing DTP is to provide an easier access to PTP, a procedure to be introduced in the next subsection.

In DTP, messages are tokens passed along the edges of the factor graph representing the CSP of interest. Specifically, the token passed from and to each variable x_v is a token on v, or equivalently, a set of (elementary) assignments on $\{v\}$. For any pair of neighboring vertices x_v and Γ_c on the factor graph, the token, or left message, $t_{v\to c}$ passed from variable x_v to constraint Γ_c depends on all incoming tokens (right messages) passed to x_v except that passed from Γ_c. Similarly, the token, or right message, $t_{c\to v}$ passed from constraint Γ_c to variable x_v depends on all incoming tokens (left messages) passed to Γ_c except that passed from x_v. Each iteration of token passing in DTP is defined by every variable passing a token on each of its edges followed by every constraint passing a token on each of its edges. Within any iteration, the token-passing rule of DTP is given as follows.

$$t_{v\to c} := \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \tag{14}$$

$$t_{c\to v} := \mathtt{F}_c\left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c}\right). \tag{15}$$

That is, the token passed from a variable is the *intersection* of its incoming tokens from the upstream, whereas the token passed from a constraint is the forced token via the constraint by the rectangle formed by the upstream incoming tokens as sides.

It is intuitive to illuminate this message-passing rule using the following analogy. We may view the token sent from a variable as the "intention" of the variable, indicating the possible values that the variable intends to take. On the other hand, we may view the token sent from a constraint as the "command" from the constraint, indicating the possible values that the constraint allows the destination variable to take. If a is an intention and b is a command, where both are tokens on the same coordinate, then the relationship $a \subseteq b$ may be viewed as that "intention a obeys command b". Under this perspective, the token sent from a variable is the "maximal" intention of the variable that obeys all incoming commands from the upstream constraints; on the other hand, the token sent from a constraint is the "maximal" command that is "compatible" with all incoming intentions from the upstream variables. Here "maximality" is in the sense of maximizing the cardinality of the subset of assignments, and "compatibility" is in the sense of satisfying the local constraint.

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

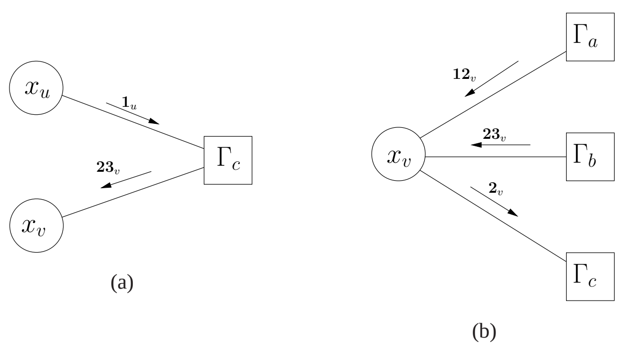


Fig. 3. Examples of deterministic token passing for a 3-COL problem. (a) Token $t_{c\to v}$ passed from constraint Γ_c to variable x_v. (b) Token $t_{v\to c}$ passed from variable x_v to constraint Γ_c.

A summary message or "summary token" at variable vertex x_v may be computed, according to the rule in (16) for each $v \in V$ at any iteration after the all constraint vertices have passed tokens.

$$t_v := \bigcap_{b \in C(v)} t_{b\to v}. \tag{16}$$

Using the "intention/command" analogy, the summary token at a variable is the "maximal" intention of the variable that obeys the incoming commands from *all* directions.

Some caution is needed on the well-definedness of the updating rule of passed tokens and summary tokens. That is, in (14), (15) and (16) the right-hand side can be equal to the empty set $\emptyset$, which is not a well-defined token. Whenever in an iteration a not-well-defined token (i.e., the empty set) arises from the updating rule, we may force DTP to terminate. — As we will see later in the "random" version of DTP (i.e., PTP and weighted PTP), we will eventually condition on the case in which these events do not happen.

At any iteration, one may read out the summary tokens at all variable vertices and form a rectangle on V using these tokens as its sides. It is clear that at any given iteration, the resulting rectangle formed by the summary tokens depends on the initialization of DTP.

Although our primary purpose of introducing DTP is to make smoother the transition to understanding PTP, in Appendix A, we present some elementary results concerning the dynamics of DTP. We note that those results will also be used to derive some insights on the dynamics of PTP — an algorithmic procedure that we introduce next as a simple formulation of SP.

C. Probabilistic Token Passing (PTP)

We now introduce the "probabilistic token passing" (or PTP) procedure. The key distinction between PTP and DTP is that on each edge and along each direction, PTP passes a *random* token and the messages being updated in PTP are the *distributions* of the random tokens.

Specifically, PTP message-update rule can be constructed by considering the following mechanism of passing random tokens.

1) On each edge connecting variable x_v and constraint Γ_c in the factor graph, the token $t_{v\to c}$ passed to constraint Γ_c and the token $t_{c\to v}$ passed to variable x_v are both *random variables*, distributed over $(\chi^*)^{\{v\}}$.
2) For any given vertex in the factor graph, all of its incoming random tokens are assumed to be independent.
3) For any given vertex in the factor graph, the outgoing random token sent along any edge is a function of all the incoming random tokens from the upstream, where the functional dependency is precisely that specified in DTP, namely, (14) or (15), depending on whether the vertex is a variable vertex or a function (constraint) vertex.
4) The summary (random) token t_v at each variable vertex x_v is a function of all incoming random tokens, where the functional dependency is precisely that specified in DTP, namely, (16).

Building on this mechanism, we will then define each PTP (passed or summary) message as the distribution of the corresponding random token *conditioned* on that the token is well defined (namely, not equal to the empty set). We note that such a “conditioning” merely involves a normalization (namely, scaling) of each message so that it sums to 1 over all valid tokens. We will use $\lambda_{v\to c}$ to denote the message sent from x_v to Γ_c — also referred to as a left message, $\rho_{c\to v}$ to denote the message sent from Γ_c to x_v — also referred to as a right message, and μ_v to denote the summary message at variable vertex x_v. It is then straight-forward to derive the message-update rule of PTP as follows, where the superscript “`norm`” on a message indicates that the message has been normalized.

PTP Message-Update Rule

$$\lambda_{v\to c}(t_{v\to c}) := \sum_{\langle t_{b\to v}\rangle_{b\in C(v)\setminus\{c\}}} \left[t_{v\to c} = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \right] \prod_{b\in C(v)\setminus\{c\}} \rho^{\rm norm}_{b\to v}(t_{b\to v}) \quad (17)$$

$$\rho_{c\to v}(t_{c\to v}) := \sum_{\langle t_{u\to c}\rangle_{u\in V(c)\setminus\{v\}}} \left[t_{c\to v} = \mathtt{F}_c \left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c} \right) \right] \prod_{u\in V(c)\setminus\{v\}} \lambda^{\rm norm}_{u\to c}(t_{u\to c}) \quad (18)$$

$$\mu_v(t_v) := \sum_{\langle t_{c\to v}\rangle_{c\in C(v)}} \left[t_v = \bigcap_{c\in C(v)} t_{c\to v} \right] \prod_{c\in C(v)} \rho^{\rm norm}_{c\to v}(t_{c\to v}), \quad (19)$$

and the normalized messages are defined as

$$\lambda^{\rm norm}_{v\to c}(t_{v\to c}) := \lambda_{v\to c}(t_{v\to c}) / \sum_{t\in(\chi^*)^{\{v\}}} \lambda_{v\to c}(t) \quad (20)$$

$$\rho^{\rm norm}_{c\to v}(t_{c\to v}) := \rho_{c\to v}(t_{c\to v}) / \sum_{t\in(\chi^*)^{\{v\}}} \rho_{c\to v}(t) \quad (21)$$

$$\mu^{\rm norm}_{v}(t_{v}) := \mu_{v}(t_{v}) / \sum_{t\in(\chi^*)^{\{v\}}} \mu_{v}(t). \quad (22)$$

We note that the update of messages in each PTP iteration is proceeded by first computing the un-normalized messages and then computing their normalized version.

D. SP as PTP

We now show that SP is precisely PTP using the example of 3-COL problems. Here we note that it is possible (and entails little additional difficulty) to show the equivalence between PTP and the *general* formulation of non-weighted SP [19] for arbitrary CSPs. However, as we feel it unnecessary to distract the readers with the additional statistical physics terminologies presented in [19], we choose not to repeat the exposition of SP in [19] and only show that SP is PTP for the special case of 3-COL problems.

In the factor graph representing a 3-COL problem, noting that each constraint vertex has degree 2, we will make a slight abuse of notation: for any $(v-c)$ pair, we will use $V(c)\setminus\{v\}$ to also denote the *index* of the unique other variable vertex (besides x_v) connecting to Γ_c, although $V(c)\setminus\{v\}$ originally refers to the singleton set containing that index. Whether $V(c)\setminus\{v\}$ should be treated as the index of a variable or as the singleton set containing the index should be clear from the context.

For notational simplicity, from here on, for every element in the token set $(\chi^*)^{\{v\}}$, when no ambiguity is resulted, we will suppress the subscript indicating the coordinate of the element. For example, we will write $\mathbf{12}_v$ as $\mathbf{12}$, when the subscript can be recovered from the context. Additionally, we will use $i, j,$ and k to denote the three distinct colors $1, 2,$ and 3 in the 3-COL problem, so that token $\mathbf{i}$ can refer to any token that is a singleton set, token $\mathbf{ij}$ can refer to any token that contains a pair of assignments, and token $\mathbf{ijk}$ refers to the token containing all three assignments.

Using these notations, the PTP message-update rule for 3-COL problems can be easily derived, which is presented in the following lemma.

Lemma 3: For 3-COL problems, the PTP message-update rule is:

$$\lambda_{v\to c}(\mathbf{i}) := \prod_{b\in C(v)\setminus\{c\}}\left(\rho^{\text{norm}}_{b\to v}(\mathbf{ij})+\rho^{\text{norm}}_{b\to v}(\mathbf{ik})+\rho^{\text{norm}}_{b\to v}(\mathbf{ijk})\right)-\prod_{b\in C(v)\setminus\{c\}}\left(\rho^{\text{norm}}_{b\to v}(\mathbf{ij})+\rho^{\text{norm}}_{b\to v}(\mathbf{ijk})\right)$$
$$-\prod_{b\in C(v)\setminus\{c\}}\left(\rho^{\text{norm}}_{b\to v}(\mathbf{ik})+\rho^{\text{norm}}_{b\to v}(\mathbf{ijk})\right)+\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{ijk}) \tag{23}$$
$$\lambda_{v\to c}(\mathbf{ij}) := \prod_{b\in C(v)\setminus\{c\}}\left(\rho^{\text{norm}}_{b\to v}(\mathbf{ij})+\rho^{\text{norm}}_{b\to v}(\mathbf{ijk})\right)-\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{ijk}) \tag{24}$$
$$\lambda_{v\to c}(\mathbf{ijk}) := \prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{ijk}) \tag{25}$$
$$\rho_{c\to v}(\mathbf{ij}) := \lambda^{\text{norm}}_{V(c)\setminus\{v\}\to c}(\mathbf{k}) \tag{26}$$
$$\rho_{c\to v}(\mathbf{ijk}) := \lambda^{\text{norm}}_{V(c)\setminus\{v\}\to c}(\mathbf{ij})+\lambda^{\text{norm}}_{V(c)\setminus\{v\}\to c}(\mathbf{ik})+\lambda^{\text{norm}}_{V(c)\setminus\{v\}\to c}(\mathbf{jk})+\lambda^{\text{norm}}_{V(c)\setminus\{v\}\to c}(\mathbf{ijk}) \tag{27}$$
$$\mu_v(\mathbf{i}) := \prod_{c\in C(v)}\left(\rho^{\text{norm}}_{c\to v}(\mathbf{ij})+\rho^{\text{norm}}_{c\to v}(\mathbf{ik})+\rho^{\text{norm}}_{c\to v}(\mathbf{ijk})\right)-\prod_{c\in C(v)}\left(\rho^{\text{norm}}_{c\to v}(\mathbf{ij})+\rho^{\text{norm}}_{c\to v}(\mathbf{ijk})\right)$$
$$-\prod_{c\in C(v)}\left(\rho^{\text{norm}}_{c\to v}(\mathbf{ik})+\rho^{\text{norm}}_{c\to v}(\mathbf{ijk})\right)+\prod_{c\in C(v)}\rho^{\text{norm}}_{c\to v}(\mathbf{ijk}) \tag{28}$$
$$\mu_v(\mathbf{ij}) := \prod_{c\in C(v)}\left(\rho^{\text{norm}}_{c\to v}(\mathbf{ij})+\rho^{\text{norm}}_{c\to v}(\mathbf{ijk})\right)-\prod_{c\in C(v)}\rho^{\text{norm}}_{c\to v}(\mathbf{ijk}) \tag{29}$$
$$\mu_v(\mathbf{ijk}) := \prod_{c\in C(v)}\rho^{\text{norm}}_{c\to v}(\mathbf{ijk}). \tag{30}$$

It is then possible to relate the PTP messages and the (non-weighted) SP messages for 3-COL problems, and show their equivalence.

Theorem 1: For 3-COL problems, the correspondence between SP and PTP message-update rules is

$$\begin{aligned}
\eta^{i}_{u\to v} &\leftrightarrow \lambda^{\text{norm}}_{u\to\{u,v\}}(\mathbf{i})\\
\eta^{*}_{u\to v} &\leftrightarrow 1-\sum_{\mathbf{i}=\mathbf{1},\mathbf{2},\mathbf{3}}\lambda^{\text{norm}}_{u\to\{u,v\}}(\mathbf{i})\\
&= \lambda^{\text{norm}}_{u\to\{u,v\}}(\mathbf{ij})+\lambda^{\text{norm}}_{u\to\{u,v\}}(\mathbf{ik})+\lambda^{\text{norm}}_{u\to\{u,v\}}(\mathbf{jk})+\lambda^{\text{norm}}_{u\to\{u,v\}}(\mathbf{ijk})\\
\eta^{i}_{u} &\leftrightarrow \mu^{\text{norm}}_{u}(\mathbf{i})\\
\eta^{*}_{u} &\leftrightarrow 1-\sum_{\mathbf{i}=\mathbf{1},\mathbf{2},\mathbf{3}}\mu^{\text{norm}}_{u}(\mathbf{i})
\end{aligned} \tag{31}$$

Proof: First we will identify c in the subscript of $\lambda^{\text{norm}}_{u\to c}$ with $\{u,v\}$ in which v indexes the destination vertex in the subscript of $\eta_{u\to v}$.

For any $c = \{u, v\}$, let $\alpha_{u,v} = \lambda^{\text{norm}}_{u\to c}(\mathbf{ij}) + \lambda^{\text{norm}}_{u\to c}(\mathbf{ik}) + \lambda^{\text{norm}}_{u\to c}(\mathbf{jk}) + \lambda^{\text{norm}}_{u\to c}(\mathbf{ijk})$. When applying PTP update equations (26) and (27) to equations (23) to (25) and re-writing the update rule in terms of left messages only, the un-normalized left messages are updated as follows.

$$\begin{aligned}
\lambda_{u\to c}(\mathbf{i}) &= \prod_{b\in C(u)\setminus\{c\}} \left(1 - \lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{i})\right) - \prod_{b\in C(u)\setminus\{c\}} \left(\lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{j}) + \alpha_{V(b)\setminus\{u\},u}\right) \quad (32)\\
&\quad - \prod_{b\in C(u)\setminus\{c\}} \left(\lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{k}) + \alpha_{V(b)\setminus\{u\},u}\right) + \prod_{b\in C(u)\setminus\{c\}} \alpha_{V(b)\setminus\{u\},u}\\
\lambda_{u\to c}(\mathbf{ij}) &= \prod_{b\in C(u)\setminus\{c\}} \left(\lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{k}) + \alpha_{V(b)\setminus\{u\},u}\right) - \prod_{b\in C(u)\setminus\{c\}} \alpha_{V(b)\setminus\{u\},u} \quad (33)\\
\lambda_{u\to c}(\mathbf{ijk}) &= \prod_{b\in C(u)\setminus\{c\}} \alpha_{V(b)\setminus\{u\},u}. \quad (34)
\end{aligned}$$

After normalization, we have

$$\begin{aligned}
\lambda^{\text{norm}}_{u\to c}(\mathbf{i}) &= \frac{1}{\beta}\cdot\left(\prod_{b\in C(u)\setminus\{c\}} \left(1 - \lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{i})\right) - \prod_{b\in C(u)\setminus\{c\}} \left(\lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{j}) + \alpha_{V(b)\setminus\{u\},u}\right)\right.\\
&\quad \left. - \prod_{b\in C(u)\setminus\{c\}} \left(\lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{k}) + \alpha_{V(b)\setminus\{u\},u}\right) + \prod_{b\in C(u)\setminus\{c\}} \alpha_{V(b)\setminus\{u\},u}\right) \quad (35)\\
\lambda^{\text{norm}}_{u\to c}(\mathbf{ij}) &= \frac{1}{\beta}\cdot\left(\prod_{b\in C(u)\setminus\{c\}} \left(\lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{k}) + \alpha_{V(b)\setminus\{u\},u}\right) - \prod_{b\in C(u)\setminus\{c\}} \alpha_{V(b)\setminus\{u\},u}\right) \quad (36)\\
\lambda^{\text{norm}}_{u\to c}(\mathbf{ijk}) &= \frac{1}{\beta}\cdot \prod_{b\in C(u)\setminus\{c\}} \alpha_{V(b)\setminus\{u\},u}, \quad (37)
\end{aligned}$$

where $\beta := \sum_{t\in(\chi^*)^{\{u\}}} \lambda_{u\to c}(t)$.

It is easy to see that

$$\begin{aligned}
\beta = &\sum_{\mathbf{i=1,2,3}} \prod_{b\in C(u)\setminus\{c\}} \left(1 - \lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{i})\right) - \sum_{\mathbf{i=1,2,3}} \prod_{b\in C(u)\setminus\{c\}} \left(\lambda^{\text{norm}}_{V(b)\setminus\{u\}\to b}(\mathbf{i}) + \alpha_{V(b)\setminus\{u\},u}\right)\\
&+ \prod_{b\in C(u)\setminus\{c\}} \alpha_{V(b)\setminus\{u\},u}.
\end{aligned}$$

For any $c = \{u, v\}$, it is clear that when identifying $\lambda^{\text{norm}}_{u\to c}(\mathbf{i})$ with $\eta^i_{u\to v}$ and identifying $\alpha_{\{u,v\}} = 1 - \sum_{\mathbf{i=1,2,3}} \lambda^{\text{norm}}_{u\to\{u,v\}}(\mathbf{i})$ with $\eta^*_{u\to v}$, the update rule for passed message $(\eta^1_{u\to v}, \eta^2_{u\to v}, \eta^3_{u\to v}, \eta^*_{u\to v})$ in SP is resulted.

To prove the equivalence of PTP and SP summary messages, we can follow the same procedure as we did for proving the equivalence of PTP left messages and SP left messages. When applying message update equations (26) and (27) to equations (28) to (30) and re-write summary messages in terms of left messages, the PTP summary messages are updated as follows.

$$
\begin{aligned}
\mu_u(\mathbf{i}) &= \prod_{c\in C(u)} \left(1-\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{i})\right) - \prod_{c\in C(u)} \left(\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{j}) + \alpha_{V(c)\setminus\{u\},u}\right) \qquad (38)\\
&\quad - \prod_{c\in C(u)} \left(\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{k}) + \alpha_{V(c)\setminus\{u\},u}\right) + \prod_{c\in C(u)} \alpha_{V(c)\setminus\{u\},u}\\
\mu_u(\mathbf{ij}) &= \prod_{c\in C(u)} \left(\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{k}) + \alpha_{V(c)\setminus\{u\},u}\right) - \prod_{c\in C(u)} \alpha_{V(c)\setminus\{u\},u} \qquad (39)\\
\mu_u(\mathbf{ijk}) &= \prod_{c\in C(u)} \alpha_{V(c)\setminus\{u\},u}. \qquad (40)
\end{aligned}
$$

After normalization, we have

$$
\begin{aligned}
\mu^{\text{norm}}_u(\mathbf{i}) &= \frac{1}{\beta'}\cdot\Bigg(\prod_{c\in C(u)} \left(1-\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{i})\right) - \prod_{c\in C(u)} \left(\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{j}) + \alpha_{V(c)\setminus\{u\},u}\right)\\
&\quad - \prod_{c\in C(u)} \left(\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{k}) + \alpha_{V(c)\setminus\{u\},u}\right) + \prod_{c\in C(u)} \alpha_{V(c)\setminus\{u\},u}\Bigg) \qquad (41)\\
\mu^{\text{norm}}_u(\mathbf{ij}) &= \frac{1}{\beta'}\cdot\Bigg(\prod_{c\in C(u)} \left(\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{k}) + \alpha_{V(c)\setminus\{u\},u}\right) - \prod_{c\in C(u)} \alpha_{V(c)\setminus\{u\},u}\Bigg) \qquad (42)\\
\mu^{\text{norm}}_u(\mathbf{ijk}) &= \frac{1}{\beta'}\cdot\prod_{c\in C(u)} \alpha_{V(c)\setminus\{u\},u}, \qquad (43)
\end{aligned}
$$

where $\beta' := \sum_{t\in(\chi^*)^{\{u\}}} \mu_u(t)$.

It is easy to show that

$$
\begin{aligned}
\beta' = &\sum_{\mathbf{i}=\mathbf{1},\mathbf{2},\mathbf{3}} \prod_{c\in C(u)} \left(1-\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{i})\right) - \sum_{\mathbf{i}=\mathbf{1},\mathbf{2},\mathbf{3}} \prod_{c\in C(u)} \left(\lambda^{\text{norm}}_{V(c)\setminus\{u\}\to c}(\mathbf{i}) + \alpha_{V(c)\setminus\{u\},u}\right)\\
&+ \prod_{c\in C(u)} \alpha_{V(c)\setminus\{u\},u}.
\end{aligned}
$$

For any $u \in V$, it is clear that when identifying $\mu^{\text{norm}}_u(\mathbf{i})$ with η^i_u and identifying $1 - \sum_{\mathbf{i}=\mathbf{1},\mathbf{2},\mathbf{3}} \mu^{\text{norm}}_u(\mathbf{i})$ with η^*_u, the update rule for summary message $(\eta^1_u, \eta^2_u, \eta^3_u, \eta^*_u)$ in SP is resulted.

■

This theorem suggests that for 3-COL problems, SP *is* PTP. Similar results can be shown for k-SAT problems — instead of showing this result, we will in a later section, show a more general result, namely that weighted SP is weighted PTP for k-SAT problems. It should be convincing then that the general principle of designing SP algorithm for arbitrary CSPs is the recipe specified in the PTP message-update rule.

In the correspondence between SP and PTP for 3-COL problems established in this theorem, it is worth noting that symbol i in the SP messages corresponds to the singleton token $\mathbf{i}$ that contains the single element i, and symbol $*$ in the SP messages corresponds to the group of all non-singleton tokens. We note that the fact that all non-singleton tokens can be represented by a single symbol $*$ is rather a coincidence, intrinsically related to the structure of 3-COL problems, and should not be understood as a general principle. Specifically, for 3-COL problems, each constraint vertex has degree 2, and as long as a non-singleton token is passed to a constraint vertex, the outgoing token from the constraint vertex will be token $\mathbf{123}$. It is precisely due to this fact that all non-singleton tokens can be represented by the same symbol — the joker symbol $*$, as is conventionally termed. This observation then implies that for general CSPs with non-binary alphabet, SP, or equivalently PTP, may be expected to contain more than one "joker" symbols, each corresponding to one or several non-singleton tokens. In other words, this suggests that the notion of "joker" symbol in SP messages is *not* a fundamental one, and that the rather fundamental perspective of SP is the extension of the variable alphabet to its power set with empty set excluded — or equivalently via a one-to-one correspondence, the set of all tokens associated with the variable.

Finally, we remark that there can be a caveat on whether SP and PTP are exactly equivalent, when taking into account the decimation procedure associated with the SP algorithms. Specifically, we note that decimation is performed based on summary messages in SP. For 3-COL problems, each SP summary message contains "biases" on four different symbols, but each PTP summary message contains "biases" on seven different tokens. The natural decimation procedure for PTP is then to fix one "highly biased" variable to one of the seven tokens, rather than to one of the four symbols. Although it is not clear at this point whether this finer procedure may provide gains in algorithm performance, it nevertheless suggests that PTP is slightly more general than SP. Investigation on possible benefit of this slight generality can be an interesting direction of research.

E. Weighted PTP

In the mechanism of passing random tokens that underlies the PTP message passing rule, the outgoing token sent from a variable vertex is a *function* of all incoming tokens from its upstream. A natural angle to generalize the dependency of these outgoing tokens on the incoming tokens is to generalize this *functional dependency* to a *probabilistic dependency*. Specifically, using the "intention-command" analogy, this probabilistic dependency will allow the intention of a variable, conditioned on all incoming commands from the upstream, to take any set of the values — not necessarily the maximal set — that obeys by the commands, and this probabilistic dependency is specified via the probability of each allowed intention. This result in what we call *weighted PTP*.

In weighted PTP, we assume that the token $t_{v\to c}$ passed from variable vertex x_v to constraint vertex Γ_c may be any subset of the intersection of all incoming tokens passed to x_v except that passed from Γ_c, and the probability that token $t_{v\to c}$ equals to each subset is specified via a non-negative function $\omega_v(a|b)$ defined on $(\chi^*)^{\{v\}} \times \left((\chi^*)^{\{v\}} \cup \{\emptyset_v\}\right)$ for each $v \in V$. We will restrict $\omega_v(a|b)$ to an *obedience conditional* on $(\chi^*)^{\{v\}}$, the definition of which is given as follows.

Definition 1 (Obedience Conditional): A non-negative function $h(a|b)$ on $(\chi^*)^{\{v\}} \times \left((\chi^*)^{\{v\}} \cup \{\emptyset_v\}\right)$ is said to be an obedience conditional on $(\chi^*)^{\{v\}}$ if $h(a|\emptyset_v) = 0$ for all $a \in (\chi^*)^{\{v\}}$ and $h(a|b) = 0$ for any $a, b \in (\chi^*)^{\{v\}}$ with $a \not\subseteq b$.

First we note that in the definition, variable a in $h(\cdot)$ is intended to refer to an "intention", variable b is intended to refer to a "command", and function h is evaluated to zero if the command is null or if the intention does not obey the command. This is the reason for which we name such a function an "obedience" conditional. Second, it is also worth noting that an obedience conditional h as defined above is not a true conditional distribution, since it is not the case that $\sum\limits_a h(a|b) = 1$ for all b. However, it is a minor technicality to modify the definition of h (without impacting the development of any result in this paper) so that it is indeed a conditional distribution [6]. Thus for the purpose of this paper, one may always regard an obedience conditional as a conditional distribution of an intention given a command.

Apparently, function $[a = b]$ is a special case of obedience conditional, characterizing a special *functional* dependency of intention a on command b, namely that the intention set a is exactly the command set b.

We now give the precise message-update rule of weighted PTP where the only difference with PTP is in left message and summary message.

Weighted PTP Message-Update Rule

$$\lambda_{v\to c}(t_{v\to c}) := \sum_{\langle t_{b\to v}\rangle_{b\in C(v)\setminus\{c\}}} \omega_v\left(t_{v\to c}\,\middle|\,\bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}\right)\prod_{b\in C(v)\setminus\{c\}}\rho^{\mathrm{norm}}_{b\to v}(t_{b\to v}) \quad (44)$$

$$\rho_{c\to v}(t_{c\to v}) := \sum_{\langle t_{u\to c}\rangle_{u\in V(c)\setminus\{v\}}} \left[t_{c\to v} = \mathtt{F}_c\left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c}\right)\right]\prod_{u\in V(c)\setminus\{v\}}\lambda^{\mathrm{norm}}_{u\to c}(t_{u\to c}) \quad (45)$$

$$\mu_v(t_v) := \sum_{\langle t_{c\to v}\rangle_{c\in C(v)}} \omega_v\left(t_v\,\middle|\,\bigcap_{c\in C(v)} t_{c\to v}\right)\prod_{c\in C(v)}\rho^{\mathrm{norm}}_{c\to v}(t_{c\to v}), \quad (46)$$

and the normalized messages are defined as

$$\lambda^{\mathrm{norm}}_{v\to c}(t_{v\to c}) := \lambda_{v\to c}(t_{v\to c})/\sum_{t\in(\chi^*)^{\{v\}}}\lambda_{v\to c}(t) \quad (47)$$

$$\rho^{\mathrm{norm}}_{c\to v}(t_{c\to v}) := \rho_{c\to v}(t_{c\to v})/\sum_{t\in(\chi^*)^{\{v\}}}\rho_{c\to v}(t) \quad (48)$$

$$\mu^{\mathrm{norm}}_{v}(t_{v}) := \mu_{v}(t_{v})/\sum_{t\in(\chi^*)^{\{v\}}}\mu_{v}(t). \quad (49)$$

[6] Given an obedience conditional h, we may define a conditional distribution $\tilde{h}(a|b)$. Let Z be $\max\limits_{b\in(\chi^*)^{\{v\}}}\sum\limits_{a\in(\chi^*)^{\{v\}}} h(a|b)$. Let non-negative function $\tilde{h}(a|b)$ on $\left((\chi^*)^{\{v\}}\cup\{\emptyset_v\}\right)\times\left((\chi^*)^{\{v\}}\cup\{\emptyset_v\}\right)$ be defined as follows: $\tilde{h}(a|\emptyset_v) := [a = \emptyset_v]$; $\tilde{h}(\emptyset_v|b) := 1 - \sum\limits_{a\in(\chi^*)^{\{v\}}} h(a|b)/Z$ for all $b \neq \emptyset_v$; and for all other (a,b), $\tilde{h}(a|b) := h(a|b)/Z$. It is easy to see that $\tilde{h}(a|b)$ is a conditional distribution. Since eventually we will condition on that $a \neq \emptyset$, it is straight-forward to verify that the role of h is equivalent to $\tilde{h}$.

It is easily seen that weighted PTP is a family of algorithms, parametrized by a collection of obedience conditionals, $\{\omega_v : v \in V\}$, each for a coordinate. The fact that conditional distribution $\omega_v(a|b)$ generalizes indicator function $[a = b]$ immediately implies that weighted PTP generalizes PTP, as stated in the following lemma.

Lemma 4: If $\omega_v(a|b) := [a = b]$ for all $v \in V$, then weighted PTP is PTP.

F. Weighted PTP Generalizes Weighted SP

Now we will show that the weighted SP developed for k-SAT problems [15] is a special case of weighted PTP. That is, for k-SAT problems, when setting functions $\{\omega_v : v \in V\}$ in weighted PTP to a particular form, weighted SP, or SP$^*(\gamma)$ is resulted.

For a k-SAT problem, let function $\omega_v(a|b)$ for every $v \in V$ in weighted PTP be defined via a single real number $\gamma \in [0, 1]$ as follows.

$$\omega_v(a|b) := \begin{cases} \gamma, & \text{if } a = b = \mathbf{01} \\ 1-\gamma, & \text{if } a \subset b = \mathbf{01} \\ 1, & \text{if } a = b \neq \mathbf{01} \\ 0, & \text{otherwise} \end{cases} \tag{50}$$

Lemma 5: Let $\{\omega_v : v \in V\}$ in k-SAT be defined as in (50). The message-update rule of weighted PTP is then:

$$\lambda_{v\to c}(\mathbf{0}) := \prod_{b\in C(v)\setminus\{c\}} (\rho^{\text{norm}}_{b\to v}(\mathbf{0}) + \rho^{\text{norm}}_{b\to v}(\mathbf{01})) - \gamma \prod_{b\in C(v)\setminus\{c\}} \rho^{\text{norm}}_{b\to v}(\mathbf{01}) \tag{51}$$

$$\lambda_{v\to c}(\mathbf{1}) := \prod_{b\in C(v)\setminus\{c\}} (\rho^{\text{norm}}_{b\to v}(\mathbf{1}) + \rho^{\text{norm}}_{b\to v}(\mathbf{01})) - \gamma \prod_{b\in C(v)\setminus\{c\}} \rho^{\text{norm}}_{b\to v}(\mathbf{01}) \tag{52}$$

$$\lambda_{v\to c}(\mathbf{01}) := \gamma \prod_{b\in C(v)\setminus\{c\}} \rho^{\text{norm}}_{b\to v}(\mathbf{01}) \tag{53}$$

$$\rho_{c\to v}(\mathbf{0}) := [L_{v,c}=0] \cdot \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1} \lambda^{\text{norm}}_{u\to c}(\mathbf{0}) \cdot \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0} \lambda^{\text{norm}}_{u\to c}(\mathbf{1}) \tag{54}$$

$$\rho_{c\to v}(\mathbf{1}) := [L_{v,c}=1] \cdot \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1} \lambda^{\text{norm}}_{u\to c}(\mathbf{0}) \cdot \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0} \lambda^{\text{norm}}_{u\to c}(\mathbf{1}) \tag{55}$$

$$\rho_{c\to v}(\mathbf{01}) := 1 - \prod_{\substack{u\in V(c)\setminus\{v\}:\\ L_{u,c}=1}} \lambda^{\text{norm}}_{u\to c}(\mathbf{0}) \cdot \prod_{\substack{u\in V(c)\setminus\{v\}:\\ L_{u,c}=0}} \lambda^{\text{norm}}_{u\to c}(\mathbf{1}) \tag{56}$$

$$\mu_v(\mathbf{0}) := \prod_{c\in C(v)} (\rho^{\text{norm}}_{c\to v}(\mathbf{0}) + \rho^{\text{norm}}_{c\to v}(\mathbf{01})) - \gamma \prod_{c\in C(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}) \tag{57}$$

$$\mu_v(\mathbf{1}) := \prod_{c\in C(v)} (\rho^{\text{norm}}_{c\to v}(\mathbf{1}) + \rho^{\text{norm}}_{c\to v}(\mathbf{01})) - \gamma \prod_{c\in C(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}) \tag{58}$$

$$\mu_v(\mathbf{01}) := \gamma \prod_{c\in C(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}). \tag{59}$$

Proof: These update equations can be immediately obtained from weighted PTP message update equations (44) to (46), where (56) follows from

$$\begin{aligned}\rho_{c\to v}(\mathbf{01}) &= \prod_{u\in V(c)\setminus\{v\}} (\lambda^{\text{norm}}_{u\to c}(\mathbf{0}) + \lambda^{\text{norm}}_{u\to c}(\mathbf{1}) + \lambda^{\text{norm}}_{u\to c}(\mathbf{01})) - \prod_{\substack{u\in V(c)\setminus\{v\}:\\ L_{u,c}=1}} \lambda^{\text{norm}}_{u\to c}(\mathbf{0}) \cdot \prod_{\substack{u\in V(c)\setminus\{v\}:\\ L_{u,c}=0}} \lambda^{\text{norm}}_{u\to c}(\mathbf{1}) \\ &= 1 - \prod_{\substack{u\in V(c)\setminus\{v\}:\\ L_{u,c}=1}} \lambda^{\text{norm}}_{u\to c}(\mathbf{0}) \cdot \prod_{\substack{u\in V(c)\setminus\{v\}:\\ L_{u,c}=0}} \lambda^{\text{norm}}_{u\to c}(\mathbf{1})\end{aligned}$$

. ∎

Theorem 2: Let $\{\omega_v : v \in V\}$ in a k-SAT problem be defined as in (50). Denote by $(\Pi^{\mathtt{s}\ \text{norm}}_{v\to c}, \Pi^{\mathtt{u}\ \text{norm}}_{v\to c}, \Pi^{*\ \text{norm}}_{v\to c})$ the normalized version of SP message $(\Pi^{\mathtt{s}}_{v\to c}, \Pi^{\mathtt{u}}_{v\to c}, \Pi^{*}_{v\to c})$, namely that $\Pi^{\mathtt{s}\ \text{norm}}_{v\to c} = \Pi^{\mathtt{s}}_{v\to c}/\left(\Pi^{\mathtt{s}}_{v\to c} + \Pi^{\mathtt{u}}_{v\to c} + \Pi^{*}_{v\to c}\right)$, $\Pi^{\mathtt{u}\ \text{norm}}_{v\to c} = \Pi^{\mathtt{u}}_{v\to c}/\left(\Pi^{\mathtt{s}}_{v\to c} + \Pi^{\mathtt{u}}_{v\to c} + \Pi^{*}_{v\to c}\right)$, and $\Pi^{*\ \text{norm}}_{v\to c} = \Pi^{\mathtt{s}}_{v\to c}/\left(\Pi^{\mathtt{s}}_{v\to c} + \Pi^{\mathtt{u}}_{v\to c} + \Pi^{*}_{v\to c}\right)$. Then the correspondence between SP$^*(\gamma)$ message-update rule and weighted PTP message-update rule is

$$\Pi^{\mathbf{s}\ \mathrm{norm}}_{v\to c} \leftrightarrow [L_{v,c}=0]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{0})+[L_{v,c}=1]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{1}) \tag{60}$$

$$\Pi^{\mathbf{u}\ \mathrm{norm}}_{v\to c} \leftrightarrow [L_{v,c}=0]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{1})+[L_{v,c}=1]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{0}) \tag{61}$$

$$\Pi^{*\ \mathrm{norm}}_{v\to c} \leftrightarrow \lambda^{\mathrm{norm}}_{v\to c}(\mathbf{01}) \tag{62}$$

$$\eta_{c\to v} \leftrightarrow \rho^{\mathrm{norm}}_{c\to v}(\mathbf{0})+\rho^{\mathrm{norm}}_{c\to v}(\mathbf{1}) \tag{63}$$

$$\zeta^0_v \leftrightarrow \mu_v(\mathbf{0}) \tag{64}$$

$$\zeta^1_v \leftrightarrow \mu_v(\mathbf{1}) \tag{65}$$

$$\zeta^*_v \leftrightarrow \mu_v(\mathbf{01}). \tag{66}$$

Prior to proving the theorem, we will introduce some notations and a simple lemma which will be useful in the proof. For any neighboring variable vertex x_v and constraint vertex Γ_c, we will denote by $\mathbf{L_{v,c}}$ the singleton token containing the single elementary assignment that assigns coordinate v the edge label $L_{v,c}$. Similarly, we will denote by $\mathbf{\bar{L}_{v,c}}$ the singleton token containing the single elementary assignment that assigns coordinate v the negated edge label $\bar{L}_{v,c}$. With these notations, the following lemma immediately follows from Lemma 5.

Lemma 6: For any $(v-c)$ pair in a k-SAT problem, the right message $\rho^{\mathrm{norm}}_{c\to v}$ satisfies:

$$\rho^{\mathrm{norm}}_{c\to v}(\mathbf{L_{v,c}})+\rho^{\mathrm{norm}}_{c\to v}(\mathbf{01}) = 1 \tag{67}$$

$$\rho^{\mathrm{norm}}_{c\to v}(\mathbf{\bar{L}_{v,c}})+\rho^{\mathrm{norm}}_{c\to v}(\mathbf{01}) = \rho^{\mathrm{norm}}_{c\to v}(\mathbf{01}). \tag{68}$$

Now we are ready to prove Theorem 2.

Proof: We will refer to the message correspondence in Equations (60) to (62) as the “left correspondence”, the correspondence in (63) as the “right correspondence”, and the correspondence in Equations (64) to (66) as the “summary correspondence”.

We will prove the theorem by first showing that if the left correspondence holds, then the right correspondence holds, and conversely that if the right correspondence holds, then the left correspondence holds. This should prove that correspondence between SP$^*(\gamma)$ and weighted PTP in their passed messages. We will then complete the proof by showing the summary correspondence.

First suppose that the left correspondence holds, namely that $\Pi^{\mathbf{s}\ \mathrm{norm}}_{v\to c}=[L_{v,c}=0]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{0})+[L_{v,c}=1]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{1})$, $\Pi^{\mathbf{u}\ \mathrm{norm}}_{v\to c}=[L_{v,c}=0]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{1})+[L_{v,c}=1]\cdot\lambda^{\mathrm{norm}}_{v\to c}(\mathbf{0})$, and $\Pi^{*\ \mathrm{norm}}_{v\to c}=$

$\lambda_{v\to c}^{\text{norm}}(\mathbf{01})$.

In each iteration, by Lemma 5 and the fact $[L_{v,c}=1]+[L_{v,c}=0]=1$ for every $(v-c)$ pair, the right messages satisfy

$$\begin{aligned}\rho_{c\to v}(\mathbf{0})+\rho_{c\to v}(\mathbf{1})+\rho_{c\to v}(\mathbf{01}) &= [L_{v,c}=0]\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1}\lambda_{u\to c}^{\text{norm}}(\mathbf{0})\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0}\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\\ &+[L_{v,c}=1]\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1}\lambda_{u\to c}^{\text{norm}}(\mathbf{0})\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0}\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\\ &+1-\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1}\lambda_{u\to c}^{\text{norm}}(\mathbf{0})\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0}\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\\ &= 1.\end{aligned}$$

That is, each right message $\rho_{c\to v}$ is already normalized, or $\rho_{c\to v}=\rho_{c\to v}^{\text{norm}}$. Then

$$\begin{aligned}\rho_{c\to v}^{\text{norm}}(\mathbf{0})+\rho_{c\to v}^{\text{norm}}(\mathbf{1}) &= \rho_{c\to v}(\mathbf{0})+\rho_{c\to v}(\mathbf{1})\\ &= [L_{v,c}=0]\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1}\lambda_{u\to c}^{\text{norm}}(\mathbf{0})\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0}\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\\ &\quad+[L_{v,c}=1]\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1}\lambda_{u\to c}^{\text{norm}}(\mathbf{0})\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0}\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\\ &= \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1}\lambda_{u\to c}^{\text{norm}}(\mathbf{0})\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0}\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\\ &= \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1}\left([L_{u,c}=1]\cdot\lambda_{u\to c}^{\text{norm}}(\mathbf{0})+[L_{u,c}=0]\cdot\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\right)\\ &\quad\cdot\prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0}\left([L_{u,c}=1]\cdot\lambda_{u\to c}^{\text{norm}}(\mathbf{0})+[L_{u,c}=0]\cdot\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\right)\\ &= \prod_{u\in V(c)\setminus\{v\}}\left([L_{u,c}=1]\cdot\lambda_{u\to c}^{\text{norm}}(\mathbf{0})+[L_{u,c}=0]\cdot\lambda_{u\to c}^{\text{norm}}(\mathbf{1})\right)\\ &\overset{(a)}{=} \prod_{u\in V(c)\setminus\{v\}}\Pi_{u\to c}^{\mathtt{u}\ \text{norm}}\\ &\overset{(b)}{=} \prod_{u\in V(c)\setminus\{v\}}\frac{\Pi_{u\to c}^{\mathtt{u}}}{\Pi_{u\to c}^{\mathtt{u}}+\Pi_{u\to c}^{\mathtt{s}}+\Pi_{u\to c}^{*}}\\ &= \eta_{c\to v},\end{aligned}$$

where equality (a) is due to the assumed left correspondence, and equality (b) follows from the definition of $\Pi_{v\to c}^{\mathtt{u}\ \text{norm}}$. Thus we have shown that if the left correspondence holds, then the right correspondence holds.

Now suppose that the right correspondence holds, namely that $\eta_{c\to v} = \rho^{\text{norm}}_{c\to v}(\mathbf{0}) + \rho^{\text{norm}}_{c\to v}(\mathbf{1})$ for every $(v-c)$ pair. Following the PTP message-update equations (51) to (53), we have

$$
\begin{aligned}
& [L_{v,c}=0]\cdot\lambda_{v\to c}(\mathbf{0}) + [L_{v,c}=1]\cdot\lambda_{v\to c}(\mathbf{1}) \\
= \;& [L_{v,c}=0]\cdot\left(\prod_{b\in C(v)\setminus\{c\}}(\rho^{\text{norm}}_{b\to v}(\mathbf{0})+\rho^{\text{norm}}_{b\to v}(\mathbf{01})) - \gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{01})\right) \\
& +[L_{v,c}=1]\cdot\left(\prod_{b\in C(v)\setminus\{c\}}(\rho^{\text{norm}}_{b\to v}(\mathbf{1})+\rho^{\text{norm}}_{b\to v}(\mathbf{01})) - \gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{01})\right) \\
= \;& [L_{v,c}=0]\cdot\prod_{b\in C(v)\setminus\{c\}}(\rho^{\text{norm}}_{b\to v}(\mathbf{0})+\rho^{\text{norm}}_{b\to v}(\mathbf{01})) + [L_{v,c}=1]\cdot\prod_{b\in C(v)\setminus\{c\}}(\rho^{\text{norm}}_{b\to v}(\mathbf{1})+\rho^{\text{norm}}_{b\to v}(\mathbf{01})) \\
& -\gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) \\
\overset{(68)}{=} \;& [L_{v,c}=0]\cdot\prod_{b\in C^{\mathtt{s}}_{c}(v)}(\rho^{\text{norm}}_{b\to v}(\mathbf{0})+\rho^{\text{norm}}_{b\to v}(\mathbf{01}))\cdot\prod_{b\in C^{\mathtt{u}}_{c}(v)}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) \\
& +[L_{v,c}=1]\cdot\prod_{b\in C^{\mathtt{s}}_{c}(v)}(\rho^{\text{norm}}_{b\to v}(\mathbf{1})+\rho^{\text{norm}}_{b\to v}(\mathbf{01}))\cdot\prod_{b\in C^{\mathtt{u}}_{c}(v)}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) - \gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) \\
\overset{(67)}{=} \;& [L_{v,c}=0]\cdot\prod_{b\in C^{\mathtt{u}}_{c}(v)}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) + [L_{v,c}=1]\cdot\prod_{b\in C^{\mathtt{u}}_{c}(v)}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) - \gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) \\
= \;& \prod_{b\in C^{\mathtt{u}}_{c}(v)}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) - \gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\text{norm}}_{b\to v}(\mathbf{01}) \\
= \;& \prod_{b\in C^{\mathtt{u}}_{c}(v)}\rho^{\text{norm}}_{b\to v}(\mathbf{01})\cdot\left(1-\gamma\prod_{b\in C^{\mathtt{s}}_{c}(v)}\rho^{\text{norm}}_{b\to v}(\mathbf{01})\right) \\
= \;& \prod_{b\in C^{\mathtt{u}}_{c}(v)}(1-\rho^{\text{norm}}_{b\to v}(\mathbf{0})-\rho^{\text{norm}}_{b\to v}(\mathbf{1}))\cdot\left(1-\gamma\prod_{b\in C^{\mathtt{s}}_{c}(v)}(1-\rho^{\text{norm}}_{b\to v}(\mathbf{0})-\rho^{\text{norm}}_{b\to v}(\mathbf{1}))\right) \\
\overset{(c)}{=} \;& \prod_{b\in C^{\mathtt{u}}_{c}(v)}(1-\eta_{b\to v})\cdot\left(1-\gamma\prod_{b\in C^{\mathtt{s}}_{c}(v)}(1-\eta_{b\to v})\right) \\
= \;& \Pi^{\mathtt{s}}_{v\to c},
\end{aligned}
$$

where equality (c) above is due to the assumed right correspondence. We will denote this result by (A).

Following very similar procedures, it can be shown that

$$\begin{aligned} & [L_{v,c}=0]\cdot\lambda_{v\to c}(\mathbf{1})+[L_{v,c}=1]\cdot\lambda_{v\to c}(\mathbf{0}) \\ = & \prod_{b\in C_c^{\mathtt{s}}(v)}(1-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{0})-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{1}))\cdot\left(1-\gamma\prod_{b\in C_c^{\mathtt{u}}(v)}(1-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{0})-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{1}))\right) \\ = & \Pi_{v\to c}^{\mathtt{u}} \end{aligned}$$

We will denote this result by (B).

Similarly,

$$\begin{aligned} \lambda_{v\to c}(\mathbf{01}) &= \gamma\prod_{b\in C_c^{\mathtt{s}}(v)}(1-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{0})-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{1}))\cdot\prod_{b\in C_c^{\mathtt{u}}(v)}(1-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{0})-\rho_{b\to v}^{\mathrm{norm}}(\mathbf{1})) \\ &= \Pi_{v\to c}^{*}. \end{aligned}$$

We will denote this result by (C).

Combining results (A), (B) and (C), we have

$$\lambda_{v\to c}(\mathbf{0})+\lambda_{v\to c}(\mathbf{1})+\lambda_{v\to c}(\mathbf{01})=\Pi_{v\to c}^{\mathtt{u}}+\Pi_{v\to c}^{\mathtt{s}}+\Pi_{v\to c}^{*}.$$

That is, the scaling constant for normalizing $(\lambda_{v\to c}(\mathbf{0}),\lambda_{v\to c}(\mathbf{1}),\lambda_{v\to c}(\mathbf{01}))$ and that for normalizing $(\Pi_{v\to c}^{\mathtt{u}},\Pi_{v\to c}^{\mathtt{s}},\Pi_{v\to c}^{*})$ are identical. Then results (A), (B) and (C) respectively translate to

$$\begin{aligned} [L_{v,c}=1]\cdot\lambda_{v\to c}^{\mathrm{norm}}(\mathbf{1})+[L_{v,c}=0]\cdot\lambda_{v\to c}^{\mathrm{norm}}(\mathbf{0}) &= \Pi_{v\to c}^{\mathtt{s}\ \mathrm{norm}} \\ [L_{v,c}=0]\cdot\lambda_{v\to c}^{\mathrm{norm}}(\mathbf{1})+[L_{v,c}=1]\cdot\lambda_{v\to c}^{\mathrm{norm}}(\mathbf{0}) &= \Pi_{v\to c}^{\mathtt{u}\ \mathrm{norm}} \\ \lambda_{v\to c}^{\mathrm{norm}}(\mathbf{01}) &= \Pi_{v\to c}^{*\ \mathrm{norm}}. \end{aligned}$$

At this point we have established the correspondence between the passed messages in weighted PTP and those in weighted SP. We now prove the summary correspondence.

Starting from Lemma 5, we have

$$\begin{aligned}
\mu_v(\mathbf{0}) &= \prod_{c\in C(v)} \left(\rho^{\text{norm}}_{c\to v}(\mathbf{0}) + \rho^{\text{norm}}_{c\to v}(\mathbf{01})\right) - \gamma \prod_{c\in C(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}) \\
&= \prod_{c\in C^1(v)} \left(\rho^{\text{norm}}_{c\to v}(\mathbf{0}) + \rho^{\text{norm}}_{c\to v}(\mathbf{01})\right) \prod_{c\in C^0(v)} \left(\rho^{\text{norm}}_{c\to v}(\mathbf{0}) + \rho^{\text{norm}}_{c\to v}(\mathbf{01})\right) - \gamma \prod_{c\in C(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}) \\
&\overset{(67),(68)}{=} \prod_{c\in C^1(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}) - \gamma \prod_{c\in C(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}) \\
&= \left(1 - \gamma \prod_{c\in C^0(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01})\right) \prod_{c\in C^1(v)} \rho^{\text{norm}}_{c\to v}(\mathbf{01}) \\
&= \left(1 - \gamma \prod_{c\in C^0(v)} \left(1 - \rho^{\text{norm}}_{c\to v}(\mathbf{0}) - \rho^{\text{norm}}_{c\to v}(\mathbf{1})\right)\right) \prod_{c\in C^1(v)} \left(1 - \rho^{\text{norm}}_{c\to v}(\mathbf{0}) - \rho^{\text{norm}}_{c\to v}(\mathbf{1})\right) \\
&\overset{(d)}{=} \left(1 - \gamma \prod_{c\in C^0(v)} (1 - \eta_{c\to v})\right) \prod_{c\in C^1(v)} (1 - \eta_{c\to v})) \\
&= \zeta_v^0
\end{aligned}$$

where (d) above is due to the right correspondence that we just proved.

Symmetrically, it can be shown that

$$\begin{aligned}
\mu_v(\mathbf{1}) &= \left(1 - \gamma \prod_{c\in C^1(v)} \left(1 - \rho^{\text{norm}}_{c\to v}(\mathbf{0}) - \rho^{\text{norm}}_{c\to v}(\mathbf{1})\right)\right) \prod_{c\in C^0(v)} \left(1 - \rho^{\text{norm}}_{c\to v}(\mathbf{0}) - \rho^{\text{norm}}_{c\to v}(\mathbf{1})\right) \\
&= \left(1 - \gamma \prod_{c\in C^1(v)} (1 - \eta_{c\to v})\right) \prod_{c\in C^0(v)} (1 - \eta_{c\to v})) \\
&= \zeta_v^1.
\end{aligned}$$

Finally, it is straight-forward to see

$$\begin{aligned}
\mu_v(\mathbf{01}) &= \gamma \prod_{c\in C^0(v)} \left(1 - \rho^{\text{norm}}_{c\to v}(\mathbf{0}) - \rho^{\text{norm}}_{c\to v}(\mathbf{1})\right) \prod_{c\in C^1(v)} \left(1 - \rho^{\text{norm}}_{c\to v}(\mathbf{0}) - \rho^{\text{norm}}_{c\to v}(\mathbf{1})\right) \\
&= \gamma \prod_{c\in C^0(v)} (1 - \eta_{c\to v}) \prod_{c\in C^1(v)} (1 - \eta_{c\to v}) \\
&= \zeta_v^*.
\end{aligned}$$

This proves the summary correspondence and completes the proof. ■

This theorem asserts that weighted SP developed for k-SAT problems is an instance of weighted PTP that we propose in this paper, or alternatively phrased, weighted PTP generalizes weighted SP from the context of k-SAT problems to arbitrary CSPs with arbitrary variable alphabets. When specifying parameter γ to be 1, this result immediately implies that non-weighted SP is non-weighted PTP for k-SAT problems.

Additionally, we note that in the correspondence between the summary messages of weighted PTP and weighted SP in the above theorem, it is clear that symbols $0, 1,$ and $*$ in weighted SP (or SP) corresponds to tokens (sets) $\mathbf{0}$, $\mathbf{1}$ and $\mathbf{01}$ respectively. In addition, if we use notation $\mathbf{L_{v,c}}$, we may re-write the correspondence between the left messages of weighted SP and those of weighted PTP in the above theorem as

$$\begin{aligned}
\Pi^{\mathtt{s}}_{v\to c} &\leftrightarrow \lambda_{v\to c}(\mathbf{L_{v,c}}) \\
\Pi^{\mathtt{u}}_{v\to c} &\leftrightarrow \lambda_{v\to c}(\bar{\mathbf{L}}_{\mathbf{v,c}}) \\
\Pi^{*}_{v\to c} &\leftrightarrow \lambda_{v\to c}(\mathbf{01})
\end{aligned}$$

That is, symbols "s" and "u" in SP respectively correspond to singleton set $\mathbf{L_{v,c}}$ and $\bar{\mathbf{L}}_{\mathbf{v,c}}$. These observations suggest that, although blurred by the addition of single symbol $*$ to the variable alphabet, the true alphabet used as the support of SP messages is the set of all tokens associated with the variable, or equivalently, the power set of the original alphabet with the empty set removed.

At this point, questions may naturally arise pertaining to what PTP and weighted PTP do towards the goal of solving a CSP. Although rigorous question this question remains largely open at this point, we present some preliminary results in Appendix B. From Appendix B, intuitively one may view PTP or weighted PTP as essentially updating a *random rectangle* whose sides are independently distributed random variables; as PTP iterates, it drives some side of the random rectangle to being deterministically biased towards a singleton that contains the solution of the CSP. The reader is referred to Appendix B for more detailed exposition.

VI. The Reduction of SP from BP

At this point, we have identified SP with an equivalent but probabilistically interpretable algorithmic procedure, PTP, and generalized weighted SP from the special case of k-SAT and binary problems to arbitrary CSPs, in terms of weighted PTP. Now we are in the position to discuss the reduction of SP from BP, where we will refer to SP exclusively as PTP, and weighted SP exclusively as weighted PTP.

As is well known, the derivation of the BP algorithm is based on a well-defined factoring function, or seen from a probabilistic perspective, a Markov random field (MRF). Thus, whether PTP or weighted PTP may be reduced from BP boils down to whether there is an MRF formulation on which the derived BP algorithm coincides with PTP or weighted PTP. In [15], an MRF is constructed for k-SAT problem, on which BP reduces to what we now call weighted PTP. In [17], similar results are shown using a different MRF formalism, where (generalized) states are introduced and the MRF is represented by a Forney graph or normal realization[18]. Although in some sense, the normally realized MRF formalism of [17] is equivalent to the MRF of [15], the Forney-graph formalism in [17] makes the development cleaner and more transparent, and the explicit introduction of states provides a better correspondence with the weighted PTP messages.

In this section, we first generalize the MRF formalism, in the style of [15] or [17], to arbitrary CSPs, and derive the corresponding BP algorithm. We then investigate whether the derived BP algorithm may be reduced to PTP or weighted PTP. We will begin this investigation with the special case of k-SAT problems, and then proceed to the 3-COL problems and to general CSPs. Re-developing the results of [15] and [17] for k-SAT problems, we show that the BP algorithm on the normally realized MRF is readily reducible to weighted PTP as long as the BP messages are initialized to satisfying certain condition. We note that this condition, when satisfied in the first BP iteration, will necessarily be satisfied in later iterations in k-SAT problems. Identifying the important role of this condition, we call this condition the *state-decoupling condition*. However, as we proceed to show, in 3-COL problems, it is impossible for the state-decoupling condition to hold true non-trivially across all BP iterations. Nevertheless, if one manually manipulate the BP messages to impose this condition in every iteration, which results in a modified BP message-update rule referred to as *state-decoupled BP* or SDBP in short, then the (SD)BP messages will still reduce to PTP. This on one hand justifies the role of the state-decoupling condition in BP-to-PTP reduction, and on the other hand suggests that for general CSPs, PTP (or SP) is not a special case of the BP algorithm. We then proceed further by investigating whether the state-decoupling condition is sufficient for BP to reduce to PTP or weighted PTP for general CSPs.

To that end, we show that yet another "local compatibility" condition concerning the structure of the CSP (in terms of the interaction between neighboring constraints) is required for SDBP to reduce to PTP or weighted PTP.

A. Normally Realized Markov Random Field

Given a CSP represented by factor graph G, we now define its corresponding *normally realized Markov random field* $\tilde{G}$ using a Forney graph representation [18]. We note that random variables involved in the probability mass function (PMF) represented by $\tilde{G}$ are no longer those associated with factor graph (or equivalently MRF) G, but rather a new set of random variables, each distributed over the set of *tokens* associated with a coordinate. Additionally, as the central component of the Forney graph, another set of random variables, typically called *generalized states* or simply *states*, are also included.

Specifically, as a graph, $\tilde{G}$ can be constructed by adding a "half-edge" to each variable vertex of G. As a factor graph, $\tilde{G}$ uses a different notation: edges and half edges are interpreted as "variables" and vertices are interpreted as local functions; a variable is an argument of the function if and only if the corresponding edge or half edge is incident on the corresponding vertex. We now define each variable and local function in $\tilde{G}$.

- Each local function (or vertex) in $\tilde{G}$ corresponding to variable vertex x_v in G will be denoted by $g_v(\cdot)$, and referred to as a *left function*.
- Each local function (or vertex) in $\tilde{G}$ corresponding to function vertex Γ_c will be denoted by $f_c(\cdot)$, and referred to as a *right function*.
- The half edge incident on g_v represents variable y_v, referred to as a *side*, taking values from $(\chi^*)^{\{v\}}$.
- The edge connecting left function g_v and right function f_c represents variable $s_{v,c}$, referred to as a *state*, taking values from $(\chi^*)^{\{v\}} \times (\chi^*)^{\{v\}}$. We will also write state $s_{v,c}$ as pair $\left(s^L_{v,c}, s^R_{v,c}\right)$ of *left state* $s^L_{v,c}$ and *right state* $s^R_{v,c}$.
- Left function g_v for $v \in V$ is defined as
$$g_v(y_v, s_{v,C(v)}) := \omega_v \left(y_v \,\middle|\, \bigcap_{c \in C(v)} s^R_{v,c} \right) \cdot \prod_{c \in C(v)} [s^L_{v,c} = y_v], \tag{69}$$
where $s_{v,C(v)}$ is the short-hand notation for $\langle s_{v,c} \rangle_{c \in C(v)}$ and ω_v is an obedience conditional on $(\chi^*)^{\{v\}}$.

- Right function f_c for each $c \in C$ is defined as

$$f_c(s_{V(c),c}) := \prod_{v \in V(c)} [s^R_{v,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{v\},c})], \tag{70}$$

where $s_{V(c),c}$ is the short-hand notation for $\langle s_{v,c} \rangle_{v \in V(c)}$.

- The global function represented by $\tilde{G}$ is

$$F(y_V, s_{V,C}) := \prod_{v \in V} g_v(y_v, s_{v,C(v)}) \cdot \prod_{c \in C} f_c(s_{V(c),c}), \tag{71}$$

where $s_{V,C}$ is the short-hand notation for $\{s_{v,c} : \forall (v - c)\}$.

It is clear that upon normalization, function F may represent a PMF and the factorization of F encoded by $\tilde{G}$ realizes an MRF. An example of such normally realized MRF, corresponding to the toy 3-SAT problem in Figure 1, is given in Fig. 4.

Using the "intention-command" analogy, one may view that for any v, both y_v and each left state $s^L_{v,c}$ stores the intention of variable x_v, and that for any given c, each right state $s^R_{v,c}$ stores the command of constraint Γ_c sent to variable v. The intention of variable x_v depends on the intersection of all incoming commands probabilistically via the obedience conditional ω_v. The command of Γ_c sent to each variable x_v need to equal the forced token by the rectangle formed by the intentions from all other neighboring variables.

We say that a configuration of $(y_V, s_{V,C})$ is *valid* under F if it is in the support of function F (namely, if it gives rise to a non-zero value of function F). Further, rectangle y_V is said to be *valid* under F if there exists a configuration of $s_{V,C}$ such that $(y_V, s_{V,C})$ is valid under F. Then it immediately follows that the PMF represented by MRF $\tilde{G}$, upon marginalizing over states $s_{V,C}$, characterizes the set of all valid rectangles under F (via the support of the marginal of F on y_V). We now give an intuitive explanation of the MRF defining the distribution of rectangle y_V.

A simple property of such MRFs is given in the following lemma, which immediately follows from the definition of the left functions.

Lemma 7: If configuration $(y_V, s_{V,C})$ is valid under F, then it holds for every $(v - c)$ that

$$s^L_{v,c} = y_v \subseteq s^R_{v,c}.$$

Now we consider applying the BP message-update rule on the Forney graph $\tilde{G}$ we just defined, where we will use $\rho_{c\to v}$ (referred to as a right message) and $\lambda_{v\to c}$ (referred to as a left message)

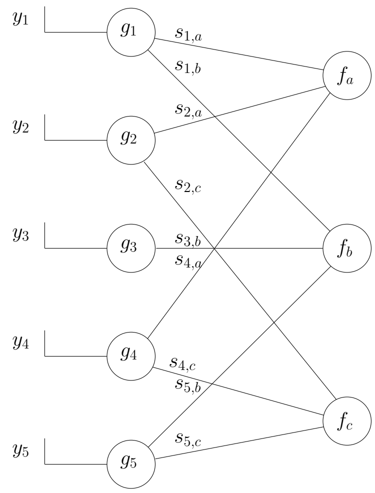


Fig. 4. The Forney graph representing the normal realization of the toy problem in Figure 1.

to denote the message passed from a right function f_c to a left function g_v and the message passed from left function g_v to right function f_c respectively, and use μ_v to denote the summary message at variable y_v. We note that both right message $\rho_{c\to v}$ and left message $\lambda_{v\to c}$ are functions on the state space $(\chi^*)^{\{v\}} \times (\chi^*)^{\{v\}}$.

Lemma 8: The BP message-update rule on Forney graph $\tilde{G}$ is:

$$\lambda_{v\to c}(s^L_{v,c}, s^R_{v,c}) := \sum_{s^R_{v,C(v)\setminus\{c\}}} \omega_v\left(s^L_{v,c}\,\middle|\,\bigcap_{b\in C(v)} s^R_{v,b}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(s^L_{v,c}, s^R_{v,b}) \tag{72}$$

$$\rho_{c\to v}(s^L_{v,c}, s^R_{v,c}) := \sum_{s^L_{V(c)\setminus\{v\},c}} \left[s^R_{v,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{v\},c})\right] \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(s^L_{u,c}, \mathtt{F}_c(s^L_{V(c)\setminus\{u\},c})) \tag{73}$$

$$\mu_v(y_v) := \sum_{s^R_{v,C(v)}} \omega_v\left(y_v\,\middle|\,\bigcap_{c\in C(v)} s^R_{v,c}\right) \prod_{c\in C(v)} \rho_{c\to v}(y_v, s^R_{v,c}). \tag{74}$$

Before proving this lemma, it is useful to note the following elementary results.

Lemma 9: 1) For any function ϕ,

$$\sum_y \phi(x,y)[y=z] = \phi(x,z). \tag{75}$$

2) For any collection of functions $\phi_1, \phi_2, \ldots, \phi_m$,

$$\sum_{x_1,x_2,\ldots,x_n} \prod_{i=1}^n \phi_i(x_i) = \prod_{i=1}^n \sum_{x_i} \phi_i(x_i). \tag{76}$$

We now prove Lemma 8.

Proof:

$$\begin{aligned}
\lambda_{v\to c}(s^L_{v,c}, s^R_{v,c}) &= \sum_{y_v} \sum_{s_{v,C(v)\setminus\{c\}}} \omega_v\left(y_v \,\middle|\, \bigcap_{b\in C(v)} s^R_{v,b}\right) \prod_{b\in C(v)} [s^L_{v,b} = y_v] \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(s^L_{v,b}, s^R_{v,b}) \\
&= \sum_{y_v} [s^L_{v,c} = y_v] \sum_{s^R_{v,C(v)\setminus\{c\}}} \omega_v\left(y_v \,\middle|\, \bigcap_{b\in C(v)} s^R_{v,b}\right) \sum_{s^L_{v,C(v)\setminus\{c\}}} \prod_{b\in C(v)\setminus\{c\}} \left(\rho_{b\to v}(s^L_{v,b}, s^R_{v,b}) \cdot [s^L_{v,b} = y_v]\right) \\
&\overset{(76)}{=} \sum_{y_v} [s^L_{v,c} = y_v] \sum_{s^R_{v,C(v)\setminus\{c\}}} \omega_v\left(y_v \,\middle|\, \bigcap_{b\in C(v)} s^R_{v,b}\right) \prod_{b\in C(v)\setminus\{c\}} \sum_{s^L_{v,b}} \left(\rho_{b\to v}(s^L_{v,b}, s^R_{v,b}) \cdot [s^L_{v,b} = y_v]\right) \\
&\overset{(75)}{=} \sum_{y_v} [s^L_{v,c} = y_v] \sum_{s^R_{v,C(v)\setminus\{c\}}} \omega_v\left(y_v \,\middle|\, \bigcap_{b\in C(v)} s^R_{v,b}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(y_v, s^R_{v,b}) \\
&= \sum_{s^R_{v,C(v)\setminus\{c\}}} \omega_v\left(s^L_{v,c} \,\middle|\, \bigcap_{b\in C(v)} s^R_{v,b}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(s^L_{v,c}, s^R_{v,b}).
\end{aligned}$$

$$\begin{aligned}
\rho_{c\to v}(s^L_{v,c}, s^R_{v,c}) &= \sum_{s_{V(c)\setminus\{v\},c}} \prod_{u\in V(c)} [s^R_{u,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{u\},c})] \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(s^L_{u,c}, s^R_{u,c}) \\
&= \sum_{s^L_{V(c)\setminus\{v\},c}} [s^R_{v,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{v\},c})] \sum_{s^R_{V(c)\setminus\{v\},c}} \prod_{u\in V(c)\setminus\{v\}} \left([s^R_{u,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{u\},c})] \cdot \lambda_{u\to c}(s^L_{u,c}, s^R_{u,c})\right) \\
&\overset{(76)}{=} \sum_{s^L_{V(c)\setminus\{v\},c}} [s^R_{v,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{v\},c})] \prod_{u\in V(c)\setminus\{v\}} \sum_{s^R_{u,c}} \left([s^R_{u,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{u\},c})] \cdot \lambda_{u\to c}(s^L_{u,c}, s^R_{u,c})\right) \\
&\overset{(75)}{=} \sum_{s^L_{V(c)\setminus\{v\},c}} [s^R_{v,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{v\},c})] \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(s^L_{u,c}, \mathtt{F}_c(s^L_{V(c)\setminus\{u\},c})).
\end{aligned}$$

$$\begin{aligned}
\mu_v(y_v) &= \sum_{s_{v,C(v)}} \omega_v\left(y_v \,\middle|\, \bigcap_{c\in C(v)} s^R_{v,c}\right) \prod_{c\in C(v)} [s^L_{v,c} = y_v] \prod_{c\in C(v)} \rho_{c\to v}(s^L_{v,c}, s^R_{v,c}) \\
&= \sum_{s^R_{v,C(v)}} \omega_v\left(y_v \,\middle|\, \bigcap_{c\in C(v)} s^R_{v,c}\right) \sum_{s^L_{v,C(v)}} \prod_{c\in C(v)} \left([s^L_{v,c} = y_v] \cdot \rho_{c\to v}(s^L_{v,c}, s^R_{v,c})\right) \\
&\overset{(76)}{=} \sum_{s^R_{v,C(v)}} \omega_v\left(y_v \,\middle|\, \bigcap_{c\in C(v)} s^R_{v,c}\right) \prod_{c\in C(v)} \sum_{s^L_{v,c}} \left([s^L_{v,c} = y_v] \cdot \rho_{c\to v}(s^L_{v,c}, s^R_{v,c})\right) \\
&\overset{(75)}{=} \sum_{s^R_{v,C(v)}} \omega_v\left(y_v \,\middle|\, \bigcap_{c\in C(v)} s^R_{v,c}\right) \prod_{c\in C(v)} \rho_{c\to v}(y_v, s^R_{v,c}).
\end{aligned}$$

■

B. *Weighted PTP as BP for k-SAT*

Now we show that for k-SAT problems, weighted PTP *is* an instance of BP when the parametrization of weighted PTP is consistent with the parametrization of the normally realized MRF from which BP is derived.

We begin with introducing a simplification of notations. For any $(v-c)$ and edge label $L_{v,c}$, we will write $\mathbf{L_{v,c}}$ as $\mathbf{L}$, and $\bar{\mathbf{L}}_{\mathbf{v},\mathbf{c}}$ as $\bar{\mathbf{L}}$. This suppression of the dependency of $\mathbf{L_{v,c}}$ and $\bar{\mathbf{L}}_{\mathbf{v},\mathbf{c}}$ on their subscripts should not result in any ambiguity, when the context clearly indicates the subscript (v,c) or the edge to which the edge label $L_{v,c}$ refers. Additionally, for any $v \in V$, we will write $\mathbf{01}_v$ as $*$. Thus, each left or right state will take configurations from set $\{\mathbf{L}, \bar{\mathbf{L}}, *\}$, where the interpretation of $\mathbf{L}$ and $\bar{\mathbf{L}}$ depends on the edge with which the state is associated. For any given configuration of a state $(s^L_{v,c}, s^R_{v,c})$, we will suppress the comma between the left-state configuration and the right-state configuration. For example, state configurations $(\mathbf{L}, *)$, $(\bar{\mathbf{L}}, *)$, $(*, *)$ and $(\mathbf{L}, \mathbf{L})$ will be written respectively as $\mathbf{L}*, \bar{\mathbf{L}}*, **$ and $\mathbf{LL}$.

Lemma 10: Let F be defined via (69), (70) and (71), where each weighting function ω_v is defined in (50). If $(y_V, s_{V,C})$ is valid under F, then

1) for every $(v-c)$, it holds that $s^R_{v,c} \neq \bar{\mathbf{L}}$, $s_{v,c} \neq \bar{\mathbf{L}}\mathbf{L}$ and that $s_{v,c} \neq *\mathbf{L}$, and
2) $F(y_V, s_{V,C}) = \gamma^{n_{*|*}(y_V, s_{V,C})} \cdot (1-\gamma)^{n_{\cdot|*}(y_V, s_{V,C})}$, where $n_{*|*}(y_V, s_{V,C})$ and $n_{\cdot|*}(y_V, s_{V,C})$ are respectively the cardinalities of set $\{v \in V : y_v = \bigcap_{c \in C(v)} s^R_{v,c} = *\}$ and set $\{v \in V : y_v \subset \bigcap_{c \in C(v)} s^R_{v,c} = *\}$.

Proof: For part 1, first we observe that $s^R_{v,c} \neq \bar{\mathbf{L}}$, directly following from the definition of the right function (70). Then by Lemma 7, it is easy to see that $s_{v,c} \neq \bar{\mathbf{L}}\mathbf{L}$ and that $s_{v,c} \neq *\mathbf{L}$.

For part 2, we may proceed as follows.

$$\begin{aligned}
F(y_V, s_{V,C}) &= \prod_{v \in V} g_v(y_v, s_{v,C(v)}) \cdot \prod_{c \in C} f_c(s_{V(c),c}) \\
&\overset{(69),(70)}{=} \prod_{v \in V} \left(\omega_v \left(y_v \middle| \bigcap_{c \in C(v)} s^R_{v,c} \right) \cdot \prod_{c \in C(v)} [s^L_{v,c} = y_v] \right) \cdot \prod_{c \in C} \prod_{v \in V(c)} [s^R_{v,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{v\},c})] \\
&\overset{(a)}{=} \prod_{v \in V} \omega_v \left(y_v \middle| \bigcap_{c \in C(v)} s^R_{v,c} \right) \\
&\overset{(b)}{=} \gamma^{n_{*|*}(y_V, s_{V,C})} \cdot (1-\gamma)^{n_{\cdot|*}(y_V, s_{V,C})},
\end{aligned}$$

where equality (a) is due to the fact that $(y_V, s_{V,C})$ is valid under F, and equality (b) follows from the definition of the weighting function $\omega\left(y_v \middle| \bigcap_{c\in C(v)} s^R_{v,c}\right)$ in (50). ■

The second part of this lemma, as a slight digression, suggests that the PMF under this MRF model is identical to that of [15], since an equivalent result is shown for the MRF in [15]. We note that the MRF in [15] serves as a combinatorial framework for the study of k-SAT problems, which leads to further insights of SP for k-SAT problems (the reader is referred to [15] for additional results). To a certain extent, one may expect that the normally realized MRF presented here may serve similar purposes for general CSPs.

The first part of this lemma suggests that although each state takes on values from $\{\mathbf{L}, \bar{\mathbf{L}}, *\} \times \{\mathbf{L}, \bar{\mathbf{L}}, *\}$, there are in fact only four possible state configurations that contribute to defining a valid rectangle. When applying the BP message-update rule on the Forney graph representation of a k-SAT problem, this implies that messages $\lambda_{v\to c}$, $\rho_{c\to v}$ and μ_v are all supported by $\{\mathbf{L}\mathbf{L}, \bar{\mathbf{L}}*, \mathbf{L}*, **\}$.

The BP message-update rule is given in Lemma 11, which directly follows from equations (72) to (74).

Lemma 11: The BP message-update rule applied on Forney graph $\tilde{G}$ of a k-SAT problem gives rise to:

$$\lambda_{v\to c}(\mathbf{LL}) := \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \prod_{b\in C_c^{\mathtt{s}}(v)} (\rho_{b\to v}(\mathbf{LL}) + \rho_{b\to v}(\mathbf{L}*)) \tag{77}$$

$$\lambda_{v\to c}(\bar{\mathbf{L}}*) := \prod_{b\in C_c^{\mathtt{s}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \left(\prod_{b\in C_c^{\mathtt{u}}(v)} (\rho_{b\to v}(\mathbf{L}*) + \rho_{b\to v}(\mathbf{LL})) - \gamma \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\mathbf{L}*) \right) \tag{78}$$

$$\lambda_{v\to c}(\mathbf{L}*) := \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \left(\prod_{b\in C_c^{\mathtt{s}}(v)} (\rho_{b\to v}(\mathbf{L}*) + \rho_{b\to v}(\mathbf{LL})) - \gamma \prod_{b\in C_c^{\mathtt{s}}(v)} \rho_{b\to v}(\mathbf{L}*) \right) \tag{79}$$

$$\lambda_{v\to c}(**) := \gamma \prod_{b\in C_c^{\mathtt{u}}(v)\cup C_c^{\mathtt{s}}(v)} \rho_{b\to v}(**) \tag{80}$$

$$\rho_{c\to v}(\mathbf{LL}) := \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \tag{81}$$

$$\begin{aligned} \rho_{c\to v}(\bar{\mathbf{L}}*) := & \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) \\ & + \sum_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{LL}) - \lambda_{u\to c}(\mathbf{L}*) - \lambda_{u\to c}(**)) \prod_{w\in V(c)\setminus\{u,v\}} \lambda_{w\to c}(\bar{\mathbf{L}}*) \\ & - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \end{aligned} \tag{82}$$

$$\rho_{c\to v}(\mathbf{L}*) := \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \tag{83}$$

$$\rho_{c\to v}(**) := \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \tag{84}$$

$$\mu_v(\mathbf{0}) := \prod_{c\in C^1(v)} \rho_{c\to v}(\bar{\mathbf{L}}*) \left(\prod_{c\in C^0(v)} (\rho_{c\to v}(\mathbf{LL}) + \rho_{c\to v}(\mathbf{L}*)) - \gamma \prod_{c\in C^0(v)} \rho_{c\to v}(\mathbf{L}*) \right) \tag{85}$$

$$\mu_v(\mathbf{1}) := \prod_{c\in C^0(v)} \rho_{c\to v}(\bar{\mathbf{L}}*) \left(\prod_{c\in C^1(v)} (\rho_{c\to v}(\mathbf{LL}) + \rho_{c\to v}(\mathbf{L}*)) - \gamma \prod_{c\in C^1(v)} \rho_{c\to v}(\mathbf{L}*) \right) \tag{86}$$

$$\mu_v(*) := \gamma \prod_{c\in C(v)} \rho_{c\to v}(**). \tag{87}$$

Now we are ready to investigate how these BP messages may reduced to (weighted) PTP messages. It turns out that the following condition has a special role to play in this reduction.

$$\rho_{c\to v}(\mathbf{L}*) = \rho_{c\to v}(\bar{\mathbf{L}}*) = \rho_{c\to v}(**) \tag{88}$$

Proposition 1: In k-SAT problems, if the BP messages are initialized to satisfy condition (88), then this condition is satisfied in every BP iteration.

Proof: We only need to show that if (88) is satisfied during initialization, then it is satisfied in the first iteration after initialization. – In fact, noting that $\rho_{c\to v}(\mathbf{L}*) = \rho_{c\to v}(**)$ necessarily holds in each BP iteration due to (83) and (84), we only need to prove that $\rho_{c\to v}(\bar{\mathbf{L}}*) = \rho_{c\to v}(\mathbf{L}*)$ holds in the first iteration provided BP messages are initialized to satisfy (88).

Under this initialization condition, we have, in the first BP iteration after,

$$
\begin{aligned}
\lambda_{v\to c}(\mathbf{L}*) + \lambda_{v\to c}(**) &= \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \times \left(\prod_{b\in C_c^{\mathtt{s}}(v)} (\rho_{b\to v}(\mathbf{L}*) + \rho_{b\to v}(\mathbf{LL})) - \gamma \prod_{b\in C_c^{\mathtt{s}}(v)} \rho_{b\to v}(\mathbf{L}*) \right) \\
&\quad + \gamma \prod_{b\in C_c^{\mathtt{u}}(v)\cup C_c^{\mathtt{s}}(v)} \rho_{b\to v}(**) \\
&= \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \prod_{b\in C_c^{\mathtt{s}}(v)} (\rho_{b\to v}(\mathbf{L}*) + \rho_{b\to v}(\mathbf{LL})) \\
&\quad - \gamma \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \prod_{b\in C_c^{\mathtt{s}}(v)} \rho_{b\to v}(\mathbf{L}*) + \gamma \prod_{b\in C_c^{\mathtt{u}}(v)\cup C_c^{\mathtt{s}}(v)} \rho_{b\to v}(**) \\
&\overset{(a)}{=} \prod_{b\in C_c^{\mathtt{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \prod_{b\in C_c^{\mathtt{s}}(v)} (\rho_{b\to v}(\mathbf{L}*) + \rho_{b\to v}(\mathbf{LL})) \\
&= \lambda_{v\to c}(\mathbf{LL}),
\end{aligned}
$$

where equality (a) is due to the initialization condition (88).

Then in the subsequent update of the right messages, we have

$$
\begin{aligned}
\rho_{c\to v}(\bar{\mathbf{L}}*) &= \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) \\
&\quad + \sum_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{LL}) - \lambda_{u\to c}(\mathbf{L}*) - \lambda_{u\to c}(**)) \prod_{w\in V(c)\setminus\{u,v\}} \lambda_{w\to c}(\bar{\mathbf{L}}*) \\
&\quad - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \\
&\overset{(b)}{=} \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \\
&= \rho_{c\to v}(\mathbf{L}*),
\end{aligned}
$$

where equality (b) is due to the above result $\lambda_{v\to c}(\mathbf{LL}) = \lambda_{v\to c}(\mathbf{L}*) + \lambda_{v\to c}(**)$.

■

Theorem 3: In a k-SAT problem, suppose that the following two conditions are imposed in the BP messages.

1) For every $(v-c)$, the BP messages are initialized such that (88) is satisfied.
2) In each BP iteration, $\lambda_{v\to c}$ is scaled to $\lambda^{\rm norm}_{v\to c}$ such that $\lambda^{\rm norm}_{v\to c}(\mathbf{L}*)+\lambda^{\rm norm}_{v\to c}(\bar{\mathbf{L}}*)+\lambda^{\rm norm}_{v\to c}(**)=1$, before it is passed along the edge; that is, $\lambda^{\rm norm}_{v\to c}(s^L_{v,c},s^R_{v,c}):=\frac{1}{\sum_{s^L_{v,c}}\lambda_{v\to c}(s^L_{v,c},*)}\cdot\lambda_{v\to c}(s^L_{v,c},s^R_{v,c})$ for every $(s^L_{v,c},s^R_{v,c})$ in the support of $\lambda_{v\to c}$ and the right messages are updated based on the normalized left messages, namely,

$$\rho_{c\to v}(\mathbf{LL}) := \prod_{u\in V(c)\setminus\{v\}} \lambda^{\rm norm}_{u\to c}(\bar{\mathbf{L}}*) \tag{89}$$

$$\begin{aligned}\rho_{c\to v}(\bar{\mathbf{L}}*) := & \prod_{u\in V(c)\setminus\{v\}} (\lambda^{\rm norm}_{u\to c}(\mathbf{L}*)+\lambda^{\rm norm}_{u\to c}(**)+\lambda^{\rm norm}_{u\to c}(\bar{\mathbf{L}}*)) \\ & + \sum_{u\in V(c)\setminus\{v\}} (\lambda^{\rm norm}_{u\to c}(\mathbf{LL})-\lambda^{\rm norm}_{u\to c}(\mathbf{L}*)-\lambda^{\rm norm}_{u\to c}(**)) \prod_{w\in V(c)\setminus\{u,v\}} \lambda^{\rm norm}_{w\to c}(\bar{\mathbf{L}}*) \\ & - \prod_{u\in V(c)\setminus\{v\}} \lambda^{\rm norm}_{u\to c}(\bar{\mathbf{L}}*)\end{aligned} \tag{90}$$

$$\rho_{c\to v}(\mathbf{L}*) := \prod_{u\in V(c)\setminus\{v\}} (\lambda^{\rm norm}_{u\to c}(\mathbf{L}*)+\lambda^{\rm norm}_{u\to c}(**)+\lambda^{\rm norm}_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda^{\rm norm}_{u\to c}(\bar{\mathbf{L}}*) \tag{91}$$

$$\rho_{c\to v}(**) := \prod_{u\in V(c)\setminus\{v\}} (\lambda^{\rm norm}_{u\to c}(\mathbf{L}*)+\lambda^{\rm norm}_{u\to c}(**)+\lambda^{\rm norm}_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda^{\rm norm}_{u\to c}(\bar{\mathbf{L}}*). \tag{92}$$

Then the correspondence between BP messages and weighted PTP messages is

$$\lambda^{\rm norm(BP)}_{v\to c}(\mathbf{L}*) \leftrightarrow [L_{v,c}=0]\cdot\lambda^{\rm norm(PTP)}_{v\to c}(\mathbf{0})+[L_{v,c}=1]\cdot\lambda^{\rm norm(PTP)}_{v\to c}(\mathbf{1}) \tag{93}$$

$$\lambda^{\rm norm(BP)}_{v\to c}(\bar{\mathbf{L}}*) \leftrightarrow [L_{v,c}=0]\cdot\lambda^{\rm norm(PTP)}_{v\to c}(\mathbf{1})+[L_{v,c}=1]\cdot\lambda^{\rm norm(PTP)}_{v\to c}(\mathbf{0}) \tag{94}$$

$$\lambda^{\rm norm(BP)}_{v\to c}(**) \leftrightarrow \lambda^{\rm norm(PTP)}_{v\to c}(*) \tag{95}$$

$$\rho^{\rm (BP)}_{c\to v}(\mathbf{L}*) \leftrightarrow \rho^{\rm norm(PTP)}_{c\to v}(*) \tag{96}$$

$$\rho^{\rm (BP)}_{c\to v}(\mathbf{LL}) \leftrightarrow \rho^{\rm norm(PTP)}_{c\to v}(\mathbf{0})+\rho^{\rm norm(PTP)}_{c\to v}(\mathbf{1}) \tag{97}$$

$$\mu^{\rm (BP)}_{v}(\mathbf{0}) \leftrightarrow \mu^{\rm (PTP)}_{v}(\mathbf{0}) \tag{98}$$

$$\mu^{\rm (BP)}_{v}(\mathbf{1}) \leftrightarrow \mu^{\rm (PTP)}_{v}(\mathbf{1}) \tag{99}$$

$$\mu^{\rm (BP)}_{v}(*) \leftrightarrow \mu^{\rm (PTP)}_{v}(*). \tag{100}$$

Proof:

Note that based on Proposition 1, condition $\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*) = \rho_{c\to v}^{(\mathrm{BP})}(\bar{\mathbf{L}}*) = \rho_{c\to v}^{(\mathrm{BP})}(**)$ holds in every BP iteration. From the proof of Proposition 1, it also holds in every BP iteration that

$$\lambda_{v\to c}^{\mathrm{norm(BP)}}(\mathbf{L}*) + \lambda_{v\to c}^{\mathrm{norm(BP)}}(**) = \lambda_{v\to c}^{\mathrm{norm(BP)}}(\mathbf{LL}). \tag{101}$$

Now we will prove this theorem by first proving that the "left correspondence" ((93) to (95)) implies the "right correspondence" ((96) and (97)) and conversely that the "right correspondence" implies the "left correspondence", whereby proving the correspondence in the passed messages. We then prove the summary correspondence ((98) to (100)).

First suppose that left correspondence holds, namely that $\lambda_{v\to c}^{\mathrm{norm(BP)}}(\mathbf{L}*) = [L_{v,c} = 0]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{0}) + [L_{v,c} = 1]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{1})$, $\lambda_{v\to c}^{\mathrm{norm(BP)}}(\bar{\mathbf{L}}*) = [L_{v,c} = 0]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) + [L_{v,c} = 1]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{0})$, and $\lambda_{v\to c}^{\mathrm{norm(BP)}}(**) = \lambda_{v\to c}^{\mathrm{norm(PTP)}}(*)$. Following PTP message-updating equations (54) to (56), we have

$$\begin{aligned}
\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{0}) \quad + \quad & \rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{1}) \overset{(a)}{=} \rho_{c\to v}^{(\mathrm{PTP})}(\mathbf{0}) + \rho_{c\to v}^{(\mathrm{PTP})}(\mathbf{1}) \\
= \quad & [L_{v,c} = 0]\cdot \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1} \lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{0}) \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0} \lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) \\
& +[L_{v,c} = 1]\cdot \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1} \lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{0}) \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0} \lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) \\
= \quad & \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1} \lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{0}) \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0} \lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) \\
= \quad & \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1} \left([L_{u,c} = 0]\cdot\lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) + [L_{u,c} = 1]\cdot\lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{0})\right) \\
& \times \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0} \left([L_{u,c} = 0]\cdot\lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) + [L_{u,c} = 1]\cdot\lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{0})\right) \\
= \quad & \prod_{u\in V(c)\setminus\{v\}} \left([L_{u,c} = 0]\cdot\lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) + [L_{u,c} = 1]\cdot\lambda_{u\to c}^{\mathrm{norm(PTP)}}(\mathbf{0})\right) \\
\overset{(b)}{=} \quad & \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}^{\mathrm{norm(BP)}}(\bar{\mathbf{L}}*) \\
= \quad & \rho_{c\to v}^{(\mathrm{BP})}(\mathbf{LL})
\end{aligned}$$

where equality (a) is due to the fact that $\rho_{c\to v}^{\mathrm{norm(PTP)}} = \rho_{c\to v}^{(\mathrm{PTP})}$ as is shown in the proof of Theorem 2, equality (b) is due to the assumed left correspondence.

Similarly, we have

$$\begin{aligned}
\rho_{c\to v}^{\text{norm(PTP)}}(*) &= \rho_{c\to v}^{\text{(PTP)}}(*) \\
&= 1 - \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=1} \lambda_{u\to c}^{\text{norm(PTP)}}(\mathbf{0}) \prod_{u\in V(c)\setminus\{v\}:L_{u,c}=0} \lambda_{u\to c}^{\text{norm(PTP)}}(\mathbf{1}) \\
&= 1 - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}^{\text{norm(BP)}}(\bar{\mathbf{L}}*) \\
&\overset{(c)}{=} \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}^{\text{norm(BP)}}(\mathbf{L}*) + \lambda_{u\to c}^{\text{norm(BP)}}(\bar{\mathbf{L}}*) + \lambda_{u\to c}^{\text{norm(BP)}}(**)) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}^{\text{norm(BP)}}(\bar{\mathbf{L}}*) \\
&= \rho_{c\to v}^{\text{(BP)}}(\mathbf{L}*),
\end{aligned}$$

where equality (c) is due to the fact that $\lambda_{u\to c}^{\text{norm(BP)}}(\mathbf{L}*) + \lambda_{u\to c}^{\text{norm(BP)}}(\bar{\mathbf{L}}*) + \lambda_{u\to c}^{\text{norm(BP)}}(**) = 1$.

Thus we proved that if the left correspondence holds, then the right correspondence holds.

Now suppose that the right correspondence holds, namely that $\rho_{c\to v}^{\text{(BP)}}(\mathbf{L}*) = \rho_{c\to v}^{\text{norm(PTP)}}(*)$, and $\rho_{c\to v}^{\text{(BP)}}(\mathbf{LL}) = \rho_{c\to v}^{\text{norm(PTP)}}(\mathbf{0}) + \rho_{c\to v}^{\text{norm(PTP)}}(\mathbf{1})$. We then have

$$\begin{aligned}
\rho_{c\to v}^{\text{(BP)}}(\mathbf{L}*) + \rho_{c\to v}^{\text{(BP)}}(\mathbf{LL}) &= \rho_{c\to v}^{\text{norm(PTP)}}(*) + \rho_{c\to v}^{\text{norm(PTP)}}(\mathbf{0}) + \rho_{c\to v}^{\text{norm(PTP)}}(\mathbf{1}) \\
&= 1.
\end{aligned}$$

Following PTP message-update equations (51) to (53), we have

$$
\begin{aligned}
& [L_{v,c}=0]\cdot\lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{0})+[L_{v,c}=1]\cdot\lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{1}) \\
= \;& [L_{v,c}=0]\cdot\left(\prod_{b\in C(v)\setminus\{c\}}(\rho^{\mathrm{norm(PTP)}}_{b\to v}(\mathbf{0})+\rho^{\mathrm{norm(PTP)}}_{b\to v}(*))-\gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)\right) \\
& +[L_{v,c}=1]\cdot\left(\prod_{b\in C(v)\setminus\{c\}}(\rho^{\mathrm{norm(PTP)}}_{b\to v}(\mathbf{1})+\rho^{\mathrm{norm(PTP)}}_{b\to v}(*))-\gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)\right) \\
= \;& [L_{v,c}=0]\cdot\prod_{b\in C(v)\setminus\{c\}}(\rho^{\mathrm{norm(PTP)}}_{b\to v}(\mathbf{0})+\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)) \\
& +[L_{v,c}=1]\cdot\prod_{b\in C(v)\setminus\{c\}}(\rho^{\mathrm{norm(PTP)}}_{b\to v}(\mathbf{1})+\rho^{\mathrm{norm(PTP)}}_{b\to v}(*))-\gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*) \\
\overset{(68)}{=} \;& [L_{v,c}=0]\cdot\prod_{b\in C^{\mathfrak{s}}_c(v)}(\rho^{\mathrm{norm(PTP)}}_{b\to v}(\mathbf{0})+\rho^{\mathrm{norm(PTP)}}_{b\to v}(*))\cdot\prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*) \\
& +[L_{v,c}=1]\cdot\prod_{b\in C^{\mathfrak{s}}_c(v)}(\rho^{\mathrm{norm(PTP)}}_{b\to v}(\mathbf{1})+\rho^{\mathrm{norm(PTP)}}_{b\to v}(*))\cdot\prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*) \\
& -\gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*) \\
\overset{(67)}{=} \;& [L_{v,c}=0]\cdot\prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)+[L_{v,c}=1]\cdot\prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)-\gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*) \\
= \;& \prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)-\gamma\prod_{b\in C(v)\setminus\{c\}}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*) \\
= \;& \prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)\left(1-\gamma\prod_{b\in C^{\mathfrak{s}}_c(v)}\rho^{\mathrm{norm(PTP)}}_{b\to v}(*)\right) \\
\overset{(d)}{=} \;& \prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{L}*)\left(1-\gamma\prod_{b\in C^{\mathfrak{s}}_c(v)}\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{L}*)\right) \\
\overset{(e)}{=} \;& \prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{L}*)\left(\prod_{b\in C^{\mathfrak{s}}_c(v)}(\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{L}*)+\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{LL}))-\gamma\prod_{b\in C^{\mathfrak{s}}_c(v)}\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{L}*)\right) \\
\overset{(f)}{=} \;& \prod_{b\in C^{\mathfrak{u}}_c(v)}\rho^{(\mathrm{BP})}_{b\to v}(\bar{\mathbf{L}}*)\left(\prod_{b\in C^{\mathfrak{s}}_c(v)}(\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{L}*)+\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{LL}))-\gamma\prod_{b\in C^{\mathfrak{s}}_c(v)}\rho^{(\mathrm{BP})}_{b\to v}(\mathbf{L}*)\right) \\
= \;& \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*)
\end{aligned}
$$

where equality (d) is due to the assumed right correspondence, equality (e) is due to the fact that

$\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*)+\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{LL})=1$, and equality (f) is due to that the condition $\rho_{b\to v}^{(\mathrm{BP})}(\mathbf{L}*)=\rho_{b\to v}^{(\mathrm{BP})}(\bar{\mathbf{L}}*)$ is satisfied in every iteration. We will denote this result by (A).

Similarly, we have

$$\begin{aligned} & [L_{v,c}=0]\cdot\lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{1})+[L_{v,c}=1]\cdot\lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{0}) \\ = \; & [L_{v,c}=0]\cdot\left(\prod_{b\in C(v)\setminus\{c\}}(\rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{1})+\rho_{b\to v}^{\mathrm{norm(PTP)}}(*))-\gamma\prod_{b\in C(v)\setminus\{c\}}\rho_{b\to v}^{\mathrm{norm(PTP)}}(*)\right) \\ & +[L_{v,c}=1]\cdot\left(\prod_{b\in C(v)\setminus\{c\}}(\rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{0})+\rho_{b\to v}^{\mathrm{norm(PTP)}}(*))-\gamma\prod_{b\in C(v)\setminus\{c\}}\rho_{b\to v}^{\mathrm{norm(PTP)}}(*)\right) \\ = \; & \lambda_{v\to c}^{(\mathrm{BP})}(\bar{\mathbf{L}}*). \end{aligned}$$

We will denote this result by (B).

Finally, we have

$$\begin{aligned} \lambda_{v\to c}^{(\mathrm{PTP})}(*) &= \gamma\prod_{b\in C(v)\setminus\{c\}}\rho_{b\to v}^{\mathrm{norm(PTP)}}(*) \\ &= \gamma\prod_{b\in C(v)\setminus\{c\}}\rho_{b\to v}^{(\mathrm{BP})}(**) \\ &= \lambda_{v\to c}^{(\mathrm{BP})}(**). \end{aligned}$$

We will denote this result by (C).

Combining results of (A), (B) and (C), we have

$$\lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{0})+\lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{1})+\lambda_{v\to c}^{(\mathrm{PTP})}(*)=\lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*)+\lambda_{v\to c}^{(\mathrm{BP})}(\bar{\mathbf{L}}*)+\lambda_{v\to c}^{(\mathrm{BP})}(**).$$

That is, the scaling constant for normalizing $(\lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{0}),\lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{1}),\lambda_{v\to c}^{(\mathrm{PTP})}(*))$ and that for normalizing $(\lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*),\lambda_{v\to c}^{(\mathrm{BP})}(\bar{\mathbf{L}}*),\lambda_{v\to c}^{(\mathrm{BP})}(**))$ are identical. Therefore, result (A), (B) and (C) respectively translate to

$$\begin{aligned} [L_{v,c}=0]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{0})+[L_{v,c}=1]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{1}) &= \lambda_{v\to c}^{\mathrm{norm(BP)}}(\mathbf{L}*) \\ [L_{v,c}=0]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{1})+[L_{v,c}=1]\cdot\lambda_{v\to c}^{\mathrm{norm(PTP)}}(\mathbf{0}) &= \lambda_{v\to c}^{\mathrm{norm(BP)}}(\bar{\mathbf{L}}*) \\ \lambda_{v\to c}^{\mathrm{norm(PTP)}}(*) &= \lambda_{v\to c}^{\mathrm{norm(BP)}}(**). \end{aligned}$$

At this point we have proved the correspondence between the passed messages in BP and those in weighted PTP.

We now prove the summary correspondence. Following the PTP message-update equations (57) to (59), we have

$$\begin{aligned}
\mu_v^{(\mathrm{PTP})}(\mathbf{0}) &= \prod_{c\in C(v)} \left(\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{0}) + \rho_{c\to v}^{\mathrm{norm(PTP)}}(*)\right) - \gamma \prod_{c\in C(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*) \\
&= \prod_{c\in C^1(v)} \left(\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{0}) + \rho_{c\to v}^{\mathrm{norm(PTP)}}(*)\right) \prod_{c\in C^0(v)} \left(\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{0}) + \rho_{c\to v}^{\mathrm{norm(PTP)}}(*)\right) \\
&\quad -\gamma \prod_{c\in C(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*) \\
&\overset{(67),(68)}{=} \prod_{c\in C^1(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*) - \gamma \prod_{c\in C(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*) \\
&= \prod_{c\in C^1(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*) \left(1 - \gamma \prod_{c\in C^0(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*)\right) \\
&= \prod_{c\in C^1(v)} \rho_{c\to v}^{(\mathrm{BP})}(\bar{\mathbf{L}}*) \left(\prod_{c\in C^0(v)} (\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{LL}) + \rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*) - \gamma \prod_{c\in C^0(v)} \rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*)\right) \\
&= \mu_v^{(\mathrm{BP})}(\mathbf{0}).
\end{aligned}$$

Following a similar procedure, we have

$$\begin{aligned}
\mu_v^{(\mathrm{PTP})}(\mathbf{1}) &= \prod_{c\in C(v)} (\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{1}) + \rho_{c\to v}^{\mathrm{norm(PTP)}}(*)) - \gamma \prod_{c\in C(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*) \\
&= \prod_{c\in C^0(v)} \rho_{c\to v}^{(\mathrm{BP})}(\bar{\mathbf{L}}*) \left(\prod_{c\in C^1(v)} (\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{LL}) + \rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*) - \gamma \prod_{c\in C^1(v)} \rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*)\right) \\
&= \mu_v^{(\mathrm{BP})}(\mathbf{1}).
\end{aligned}$$

Finally, we have

$$\begin{aligned}
\mu_v^{(\mathrm{PTP})}(*) &= \gamma \prod_{c\in C(v)} \rho_{c\to v}^{\mathrm{norm(PTP)}}(*) \\
&= \gamma \prod_{c\in C(v)} \rho_{c\to v}^{(\mathrm{BP})}(**) \\
&= \mu_v^{(\mathrm{BP})}(*),
\end{aligned}$$

which proves the summary correspondence. ■

C. State-Decoupled BP

In this subsection, we will consider reducing PTP from BP for 3-COL problems, where we only focus on the non-weighted version of PTP, namely that each weighting function ω_v is defined as

$$\omega_v(a|b) := [a = b]. \tag{102}$$

This gives the form of BP messages in the form specified in the following lemma, easily obtainable from BP update equations (72) to (74).

Lemma 12: The BP message-update rule for 3-COL problems is as follow:

$$\begin{aligned}
\lambda_{v\to c}(\mathbf{i},\mathbf{ij}) &:= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ik}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) &(103)\\
\lambda_{v\to c}(\mathbf{i},\mathbf{ijk}) &:= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ik}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ik}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) + \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(\mathbf{i},\mathbf{ijk}) &(104)\\
\lambda_{v\to c}(\mathbf{ij},\mathbf{ij}) &:= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{ij},\mathbf{ij}) + \rho_{b\to v}(\mathbf{ij},\mathbf{ijk})) &(105)\\
\lambda_{v\to c}(\mathbf{ij},\mathbf{ijk}) &:= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{ij},\mathbf{ij}) + \rho_{b\to v}(\mathbf{ij},\mathbf{ijk})) - \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(\mathbf{ij},\mathbf{ijk}) &(106)\\
\lambda_{v\to c}(\mathbf{ijk},\mathbf{ijk}) &:= \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(\mathbf{ijk},\mathbf{ijk}) &(107)
\end{aligned}$$

$$\rho_{c\to v}(\mathbf{i},\mathbf{ij}) := \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{k},\mathbf{jk}) \tag{108}$$

$$\rho_{c\to v}(\mathbf{i},\mathbf{ijk}) := \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{jk},\mathbf{jk}) \tag{109}$$

$$\rho_{c\to v}(\mathbf{ij},\mathbf{ij}) := \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{k},\mathbf{ijk}) \tag{110}$$

$$\begin{aligned}\rho_{c\to v}(\mathbf{ij},\mathbf{ijk}) := {} & \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ij},\mathbf{ijk}) + \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{jk},\mathbf{ijk}) + \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ik},\mathbf{ijk}) \\ & +\lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ijk},\mathbf{ijk})\end{aligned} \tag{111}$$

$$\begin{aligned}\rho_{c\to v}(\mathbf{ijk},\mathbf{ijk}) := {} & \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ij},\mathbf{ijk}) + \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{jk},\mathbf{ijk}) + \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ik},\mathbf{ijk}) \\ & +\lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ijk},\mathbf{ijk})\end{aligned} \tag{112}$$

$$\begin{aligned}\mu_v(\mathbf{i}) := {} & \prod_{c\in C(v)} \left(\rho_{c\to v}(\mathbf{i},\mathbf{ij}) + \rho_{c\to v}(\mathbf{i},\mathbf{ik}) + \rho_{c\to v}(\mathbf{i},\mathbf{ijk})\right) \\ & - \prod_{c\in C(v)} \left(\rho_{c\to v}(\mathbf{i},\mathbf{ij}) + \rho_{c\to v}(\mathbf{i},\mathbf{ijk})\right) \\ & - \prod_{c\in C(v)} \left(\rho_{c\to v}(\mathbf{i},\mathbf{ik}) + \rho_{c\to v}(\mathbf{i},\mathbf{ijk})\right) + \prod_{c\in C(v)} \rho_{c\to v}(\mathbf{i},\mathbf{ijk})\end{aligned} \tag{113}$$

$$\mu_v(\mathbf{ij}) := \prod_{c\in C(v)} \left(\rho_{c\to v}(\mathbf{ij},\mathbf{ij}) + \rho_{c\to v}(\mathbf{ij},\mathbf{ijk})\right) - \prod_{c\in C(v)} \rho_{c\to v}(\mathbf{ij},\mathbf{ijk}) \tag{114}$$

$$\mu_v(\mathbf{ijk}) := \prod_{c\in C(v)} \rho_{c\to v}(\mathbf{ijk},\mathbf{ijk}). \tag{115}$$

Before we begin to consider the BP-to-PTP reduction for 3-COL problems, it is helpful to take a closer look at the BP-to-PTP reduction mechanism for k-SAT problems.

In Theorem 3, one may notice the two conditions governing the BP-to-PTP reduction for k-SAT problems, namely, the initialization condition and the normalization condition. It is arguable that the normalization condition imposed on the BP messages, although serving to simplify the form of BP messages and possibly to alter the interpretation of the messages, does not have a critical impact on the message-passing dynamics. This is because the normalization condition merely involves a scaling operation, without which BP messages and PTP messages for k-SAT would still be equivalent up to a scaling factor. On the other hand, the initialization condition in Theorem 3 plays an important role on the message-passing dynamics. In essence, the initialization condition assures that any right message depends only on the right state it involves. Using the "intention-command" analogy, in which one views each right state as storing the "command" sent from a constraint and each left state as storing the "intention" of a variable, this condition simply restricts that the *distribution* of the command sent to any variable does *not* depend on the intention of the variable. It is remarkable that this interpretation of the initialization condition in Theorem 3 (or (88)) is consistent with the PTP message-passing rule, in which any right message (i.e., outgoing distribution of command) sent to a variable is independent of (or, not a function of,) the incoming intention from that variable. This is however not the case for the right messages of BP in general.

We are then motivated to formalize this condition for general CSPs as what we call the "state-decoupling" condition and impose it on the right messages of BP, so as to achieve a consistency with PTP. It is intuitively sensible that such a consistency is needed in the reduction of PTP from BP.

Definition 2 (State-Decoupling Condition): For an arbitrary CSP and at any given iteration, the BP messages based on the MRF formalism defined by (69), (70), and (71) are said to satisfy the state-decoupling condition if for every $(v-c)$, the right message $\rho_{c\to v}(s_{v,c})$ is only a function of the right state $s^R_{v,c}$, namely, if for any fixed $s^R_{v,c} \in (\chi^*)^{\{v\}}$ and any $s^L_{v,c} \subset s^R_{v,c}$, $\rho_{c\to v}(s^L_{v,c}, s^R_{v,c}) = \rho_{c\to v}(s^R_{v,c}, s^R_{v,c})$.

It is clear that the initialization condition for BP-to-PTP reduction for k-SAT in Theorem 3 is equivalent to this condition, where we note that the condition in Theorem 3 only puts restrictions on the right messages with right state equal to $*$, since for the remaining case with right state equal to $\mathbf{L}$ this condition is trivially satisfied.

It is interesting to observe, as shown in Proposition 1, that for k-SAT problems, as long as the state-decoupling condition is imposed in the initialization of the BP messages, the condition is preserved in every iteration. This serves as the basis for BP to reduce to PTP as shown in Theorem 3 and its proof. For 3-COL problems, however, the corresponding result to Proposition 1 does not hold.

Lemma 13: For 3-COL problems, if the state-decoupling condition holds for BP messages both in iteration l and in iteration $l+1$, then the right message in iteration l must satisfy for every $(v-c)$

$$\rho_{c\to v}(s^L, s^R) = 0$$

as long as right state $s^R \neq \mathbf{123}$.

Proof: In 3-COL problems, the state-decoupling condition can be expressed as

$$\begin{aligned}\rho_{c\to v}(\mathbf{i},\mathbf{ij}) &= \rho_{c\to v}(\mathbf{ij},\mathbf{ij})\\ \rho_{c\to v}(\mathbf{i},\mathbf{ijk}) &= \rho_{c\to v}(\mathbf{ij},\mathbf{ijk}) = \rho_{c\to v}(\mathbf{ijk},\mathbf{ijk}).\end{aligned}$$

Note that we only need to prove the Lemma for s^R being a pair of assignments,since when s^R is a singleton, all right messages equal 0 by the construction of the MRF and Lemma 12 describing the BP message-update rule for 3-COL.

In iteration $l+1$, following 3-COL message-update equations (103) to (112) and using a superscript to denote the iteration number, we have

$$\begin{aligned}\rho^{(l+1)}_{c\to v}(\mathbf{i},\mathbf{ij}) &= \lambda^{(l+1)}_{V(c)\setminus\{v\}\to c}(\mathbf{k},\mathbf{jk})\\ &= \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}}\left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk})+\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik})+\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\ &\quad -\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}}\left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk})+\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right), \qquad (116)\end{aligned}$$

$$\begin{aligned}\rho^{(l+1)}_{c\to v}(\mathbf{ij},\mathbf{ij}) &= \lambda^{(l+1)}_{V(c)\setminus\{v\}\to c}(\mathbf{k},\mathbf{ijk})\\ &= \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}}\left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk})+\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik})+\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\ &\quad -\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}}\left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik})+\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\ &\quad -\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}}\left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk})+\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\ &\quad +\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}}\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk}). \qquad (117)\end{aligned}$$

Now suppose that the state-decoupling condition as expressed above can be satisfied both in iteration l and in iteration $l+1$. Then we may equate the right-hand sides of (116) and (117), namely,

$$
\begin{aligned}
&\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\
&\quad - \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\
= &\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\
&\quad - \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\
&\quad - \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right) + \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk}),
\end{aligned}
$$

which implies

$$
\begin{aligned}
&\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\
= &\prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right)\\
&\quad + \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \left(\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{jk}) + \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk})\right) - \prod_{b\in C(V(c)\setminus\{v\})\setminus\{c\}} \rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ijk}).
\end{aligned}
$$

Since every right message must be non-negative, when the state-decoupling condition is satisfied in iteration l, the only way to make the above equality hold is the case where

$$\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{k},\mathbf{ik}) = 0.$$

Under the state-decoupling condition, this also means $\rho^{(l)}_{b\to V(c)\setminus\{v\}}(\mathbf{ik},\mathbf{ik}) = 0$. Thus we establish this lemma.

■

This lemma suggests that when the BP messages satisfy the state-decoupling condition in two consecutive iterations, then the right messages must take a trivial form — equal to $[s^R = \mathbf{123}]$ up to scale, and contain no information.

At this point, one is left with either the option of concluding that PTP (or SP) is *not* an instance of BP for 3-COL problems (and hence for general CSPs) or the option of doubting the usefulness of the state-decoupling condition in BP-to-SP reduction. In the remainder of this subsection, we will clear this doubt and assert the usefulness of the state-decoupling condition by showing that when the state-decoupling condition is *manually* imposed on the BP messages in each iteration, BP still reduces to PTP for 3-COL problems. That will allow us to conclude that PTP (or SP) is not a special case of BP.

To force the state-decoupling condition to be satisfied in each BP iteration, now we modify the message-passing rule of BP on the Forney graph representation of general CSPs, and introduce a "new" message-passing procedure which we refer to as the *state-decoupled BP* or SDBP. We note that introducing this "new" message-passing procedure is solely for the purpose of verifying the usefulness of the state-decoupling condition and hopefully arriving at a unified reduction mechanism for PTP to reduce from BP (or more precisely from SDBP). Beyond this purpose, we have no intention to justify the introduction of SDBP.

Identical to BP at local function vertices, SDBP differs from BP in that messages passed from the right functions need an additional processing (so that the state-decoupling condition is satisfied) before they are passed to the left functions. In SDBP, there are three kinds of messages: *right message* $\rho_{c\to v}$ is computed at right function f_c to pass along the edge to g_v; *state-decoupled right message* $\rho^*_{c\to v}$ is computed at the edge connecting f_c and g_v, which satisfies the state-decoupling condition, computed only based on the right message $\rho_{c\to v}$ on the same edge and to be passed to left function g_v; *left message* $\lambda_{v\to c}$ is computed at the left function g_v to pass along the edge connecting to f_c. The precise definition of SDBP message-update rule is given next.

Definition 3: The SDBP message-update rule is defined as follows.

$$\lambda_{v\to c}(s^L_{v,c}, s^R_{v,c}) := \sum_{s^R_{v,C(v)\setminus\{c\}}} \omega_v\left(s^L_{v,c}\,\middle|\,\bigcap_{b\in C(v)} s^R_{v,b}\right)\cdot\prod_{b\in C(v)\setminus\{c\}}\rho^*_{b\to v}(s^R_{v,b}) \tag{118}$$

$$\rho^*_{c\to v}(s^R_{v,c}) := \delta\cdot\rho_{c\to v}(s^R_{v,c}, s^R_{v,c}) \tag{119}$$

$$\rho_{c\to v}(s^L_{v,c}, s^R_{v,c}) := \sum_{s^L_{V(c)\setminus\{v\},c}}\left[s^R_{v,c} = \mathtt{F}_c(s^L_{V(c)\setminus\{v\},c})\right]\prod_{u\in V(c)\setminus\{v\}}\lambda_{u\to c}(s^L_{u,c}, \mathtt{F}_c(s^L_{V(c)\setminus\{u\},c})) \tag{120}$$

$$\mu_v(y_v) := \sum_{s^R_{v,C(v)}}\omega_v\left(y_v\,\middle|\,\bigcap_{c\in C(v)} s^R_{v,c}\right)\prod_{c\in C(v)}\rho^*_{c\to v}(s^R_{v,c}) \tag{121}$$

where $\delta = 1/\sum_{s^R_{v,c}\in(\chi^*)^{\{v\}}}\rho_{c\to v}(s^R_{v,c}, s^R_{v,c})$.

Comparing this definition with the BP message-update rule in Lemma 8, the following remarks are in order. First, the expression of right messages ρ in terms of left messages λ is identical to that in BP. Second, each state-decoupled message $\rho^*_{c\to v}$ may be regarded as a function of

$(s^L_{v,c}, s^R_{v,c})$ but the value of the function only depends the $s^R_{v,c}$ component, namely that the (state-decoupled) right message satisfies the state-decoupling condition. Furthermore, the expression of λ in terms of ρ^* is precisely the same as the expression of λ in terms of ρ in BP[7].

Following this definition, the next lemma summarizes the SDBP message-update rule for 3-COL problems.

Lemma 14: Let $\{\omega_v : v \in V\}$ in 3-COL problems be defined as in (102). The SDBP message-update rule is then :

$$\begin{aligned}
\lambda_{v\to c}(\mathbf{i},\mathbf{ij}) &:= \prod_{b\in C(v)\setminus\{c\}} \left(\rho^*_{b\to v}(\mathbf{ij}) + \rho^*_{b\to v}(\mathbf{ik}) + \rho^*_{b\to v}(\mathbf{ijk})\right) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} \left(\rho^*_{b\to v}(\mathbf{ij}) + \rho^*_{b\to v}(\mathbf{ijk})\right) \qquad (122)\\
\lambda_{v\to c}(\mathbf{i},\mathbf{ijk}) &:= \prod_{b\in C(v)\setminus\{c\}} \left(\rho^*_{b\to v}(\mathbf{ij}) + \rho^*_{b\to v}(\mathbf{ik}) + \rho^*_{b\to v}(\mathbf{ijk})\right) - \prod_{b\in C(v)\setminus\{c\}} \left(\rho^*_{b\to v}(\mathbf{ij}) + \rho^*_{b\to v}(\mathbf{ijk})\right) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} \left(\rho^*_{b\to v}(\mathbf{ik}) + \rho^*_{b\to v}(\mathbf{ijk})\right) + \prod_{b\in C(v)\setminus\{c\}} \rho^*_{b\to v}(\mathbf{ijk}) \qquad (123)\\
\lambda_{v\to c}(\mathbf{ij},\mathbf{ij}) &:= \prod_{b\in C(v)\setminus\{c\}} \left(\rho^*_{b\to v}(\mathbf{ij}) + \rho^*_{b\to v}(\mathbf{ijk})\right) \qquad (124)\\
\lambda_{v\to c}(\mathbf{ij},\mathbf{ijk}) &:= \prod_{b\in C(v)\setminus\{c\}} \left(\rho^*_{b\to v}(\mathbf{ij}) + \rho^*_{b\to v}(\mathbf{ijk})\right) - \prod_{b\in C(v)\setminus\{c\}} \rho^*_{b\to v}(\mathbf{ijk}) \qquad (125)\\
\lambda_{v\to c}(\mathbf{ijk},\mathbf{ijk}) &:= \prod_{b\in C(v)\setminus\{c\}} \rho^*_{b\to v}(\mathbf{ijk}) \qquad (126)
\end{aligned}$$

$$\begin{aligned}
\rho^*_{c\to v}(\mathbf{ij}) &:= \delta\cdot\lambda_{V(c)\setminus\{v\}\to c}(\mathbf{k},\mathbf{ijk}) \qquad (127)\\
\rho^*_{c\to v}(\mathbf{ijk}) &:= \delta\cdot\Big(\lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ij},\mathbf{ijk}) + \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ik},\mathbf{ijk}) + \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{jk},\mathbf{ijk}) \\
&\quad + \lambda_{V(c)\setminus\{v\}\to c}(\mathbf{ijk},\mathbf{ijk})\Big) \qquad (128)
\end{aligned}$$

[7]Although it is possible to formulate SDBP in more compact form by, for example, suppressing ρ and expressing the message-update rule only using ρ^* and λ, we feel the current way of formulating SDBP makes it easier to compare SDBP with BP.

$$\mu_v(\mathbf{i}) := \prod_{c\in C(v)} \left(\rho^*_{c\to v}(\mathbf{ij}) + \rho^*_{c\to v}(\mathbf{ik}) + \rho^*_{c\to v}(\mathbf{ijk})\right) - \prod_{c\in C(v)} \left(\rho^*_{c\to v}(\mathbf{ij}) + \rho^*_{c\to v}(\mathbf{ijk})\right)$$
$$- \prod_{c\in C(v)} \left(\rho^*_{c\to v}(\mathbf{ik}) + \rho^*_{c\to v}(\mathbf{ijk})\right) + \prod_{c\in C(v)} \rho^*_{c\to v}(\mathbf{ijk}) \tag{129}$$

$$\mu_v(\mathbf{ij}) := \prod_{c\in C(v)} \left(\rho^*_{c\to v}(\mathbf{ij}) + \rho^*_{c\to v}(\mathbf{ijk})\right) - \prod_{c\in C(v)} \rho^*_{c\to v}(\mathbf{ijk}) \tag{130}$$

$$\mu_v(\mathbf{ijk}) := \prod_{c\in C(v)} \rho^*_{c\to v}(\mathbf{ijk}), \tag{131}$$

where δ is such that

$$\rho^*_{c\to v}(\mathbf{ijk}) + \sum_{\mathbf{ij}} \rho^*_{c\to v}(\mathbf{ij}) = 1.$$

It is now possible to establish a correspondence between PTP and SDBP messages for 3-COL problems.

Theorem 4: For 3-COL problems, the correspondence between PTP and SDBP message-update rules is

$$\lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{i}) \leftrightarrow \lambda^{(\mathrm{SDBP})}_{v\to c}(\mathbf{i}, \mathbf{ijk}) \tag{132}$$

$$\lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{ij}) \leftrightarrow \lambda^{(\mathrm{SDBP})}_{v\to c}(\mathbf{ij}, \mathbf{ijk}) \tag{133}$$

$$\lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{ijk}) \leftrightarrow \lambda^{(\mathrm{SDBP})}_{v\to c}(\mathbf{ijk}, \mathbf{ijk}) \tag{134}$$

$$\rho^{\mathrm{norm}(\mathrm{PTP})}_{c\to v}(\mathbf{ij}) \leftrightarrow \rho^{*(\mathrm{SDBP})}_{c\to v}(\mathbf{ij}) \tag{135}$$

$$\rho^{\mathrm{norm}(\mathrm{PTP})}_{c\to v}(\mathbf{ijk}) \leftrightarrow \rho^{*(\mathrm{SDBP})}_{c\to v}(\mathbf{ijk}) \tag{136}$$

$$\mu^{(\mathrm{PTP})}_{v}(\mathbf{i}) \leftrightarrow \mu^{(\mathrm{SDBP})}_{v}(\mathbf{i}) \tag{137}$$

$$\mu^{(\mathrm{PTP})}_{v}(\mathbf{ij}) \leftrightarrow \mu^{(\mathrm{SDBP})}_{v}(\mathbf{ij}) \tag{138}$$

$$\mu^{(\mathrm{PTP})}_{v}(\mathbf{ijk}) \leftrightarrow \mu^{(\mathrm{SDBP})}_{v}(\mathbf{ijk}). \tag{139}$$

Proof: We will first prove that if the “right correspondence” (namely that (135) and (136)) holds, then the “left correspondence” (namely that (132) to (134)) holds.

Suppose that the right correspondence holds (where the symbol $\leftrightarrow$ in (135) and (136) is understood as equality). Then

$$\begin{aligned}
\lambda_{v\to c}^{(\mathrm{SDBP})}(\mathbf{i},\mathbf{ijk}) &= \prod_{b\in C(v)\setminus\{c\}} \left(\rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ij}) + \rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ik}) + \rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk})\right) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} \left(\rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ij}) + \rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk})\right) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} \left(\rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ik}) + \rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk})\right) + \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk}) \\
&= \prod_{b\in C(v)\setminus\{c\}} \left(\rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ij}) + \rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ik}) + \rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ijk})\right) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} \left(\rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ij}) + \rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ijk})\right) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} \left(\rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ik}) + \rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ijk})\right) + \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}^{\mathrm{norm(PTP)}}(\mathbf{ijk}) \\
&= \lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{i}).
\end{aligned}$$

Similarly, we can prove that $\lambda_{v\to c}^{(\mathrm{SDBP})}(\mathbf{ij},\mathbf{ijk}) = \lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{ij})$ and $\lambda_{v\to c}^{(\mathrm{SDBP})}(\mathbf{ijk},\mathbf{ijk}) = \lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{ijk})$. It then follows that the left correspondence holds.

Now we prove that if the left correspondence holds, then the right correspondence holds. Suppose that the left correspondence holds, then we have

$$\begin{aligned}
\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{ij}) &= \alpha\cdot\rho_{c\to v}^{(\mathrm{PTP})}(\mathbf{ij}) \\
&= \alpha\cdot\lambda_{V(c)\setminus\{v\}\to c}^{\mathrm{norm(PTP)}}(\mathbf{k}) \\
&= \alpha\left(\beta\cdot\lambda_{V(c)\setminus\{v\}\to c}^{(\mathrm{PTP})}(\mathbf{k})\right) \\
&= \alpha\beta\cdot\lambda_{V(c)\setminus\{v\}\to c}^{(\mathrm{SDBP})}(\mathbf{k},\mathbf{ijk})
\end{aligned}$$

where $\alpha = 1/\sum_{t\in(\chi^*)^v}\rho_{c\to v}^{(\mathrm{PTP})}(t)$ and $\beta = 1/\sum_{t\in(\chi^*)^{V(c)\setminus\{v\}}}\lambda_{V(c)\setminus\{v\}\to c}^{(\mathrm{PTP})}(t)$. We also have

$$\rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ij}) = \delta\cdot\lambda_{V(c)\setminus\{v\}\to c}^{(\mathrm{SDBP})}(\mathbf{k},\mathbf{ijk}).$$

Since both $\rho_{c\to v}^{*(\mathrm{SDBP})}$ and $\rho_{c\to v}^{\mathrm{norm(PTP)}}$ are normalized, it must hold that $\alpha\beta = \delta$. This indicates that $\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{ij}) = \rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ij})$. Following a similar procedure, one can show that $\rho_{c\to v}^{\mathrm{norm(PTP)}}(\mathbf{ijk}) = \rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk})$. This implies that the right correspondence holds.

At this point, we have established the correspondence between passed messages in PTP and those in SDBP.

Now we will prove the summary correspondence (namely, that (137) to (139)).

$$
\begin{aligned}
\mu_v^{(\mathrm{SDBP})}(\mathbf{i}) &= \prod_{c \in C(v)} \left(\rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ij}) + \rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ik}) + \rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk})\right) \\
&\quad - \prod_{c \in C(v)} \left(\rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ij}) + \rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk})\right) \\
&\quad - \prod_{c \in C(v)} \left(\rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ik}) + \rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk})\right) + \prod_{c \in C(v)} \rho_{c\to v}^{*(\mathrm{SDBP})}(\mathbf{ijk}) \\
&= \prod_{c \in C(v)} \left(\rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ij}) + \rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ik}) + \rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ijk}\right) \\
&\quad - \prod_{c \in C(v)} \left(\rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ij}) + \rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ijk})\right) \\
&\quad - \prod_{c \in C(v)} \left(\rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ik}) + \rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ijk})\right) + \prod_{c \in C(v)} \rho_{c\to v}^{\mathrm{norm}(\mathrm{PTP})}(\mathbf{ijk}) \\
&= \mu_v^{(\mathrm{PTP})}(\mathbf{i}).
\end{aligned}
$$

Similarly, we can prove that $\mu_v^{(\mathrm{SDBP})}(\mathbf{ij}) = \mu_v^{(\mathrm{PTP})}(\mathbf{ij})$ and $\mu_v^{(\mathrm{SDBP})}(\mathbf{ijk}) = \mu_v^{(\mathrm{PTP})}(\mathbf{ijk})$. This proves the summary correspondence.

■

At this end, it should be convincing that the state-decoupling condition is an important ingredient in the reduction of BP to PTP. It is worth noting that in the case of k-SAT problems, this condition can be imposed simply by the initialization of BP messages. However in the case of 3-COL problems, one needs to manually impose this condition at each iteration, namely, carrying out SDBP instead of BP, so as to arrive at an equivalence to PTP messages. This extra complexity involved in 3-COL problems then suggests that for 3-COL problems, PTP and hence SP are not a special case of BP. Thus at this end, one may conclude that SP is not BP for general CSPs.

Now it remains to investigate, for general CSPs, whether the state-decoupling condition is sufficient for PTP or weighted PTP to reduce from BP, or equivalently *whether* and *when* PTP and weighted PTP are SDBP.

D. The Reduction of Weighted PTP from SDBP for General CSPs

Up to this point, we see that the state-decoupling condition critically governs the reduction of BP to PTP (or weighted PTP) for k-SAT problems and 3-COL problems. In this subsection, we will however show that the state-decoupling condition is not sufficient for BP (more precisely SDBP) to reduce to PTP and that an additional condition is needed in the general context.

Definition 4 (Forceable Token): For any $(v - c)$, we say that a token $t_v \in (\chi^*)^{\{v\}}$ is *forceable* by Γ_c if there exists a rectangle $\prod\limits_{u \in V(c)\setminus\{v\}} t_u$ on $V(c) \setminus \{v\}$ such that $\mathtt{F}_c \left(\prod\limits_{u \in V(c)\setminus\{v\}} t_u \right) = t_v$.

We will denote by $\mathcal{F}_c(v)$ the set of all tokens on v that are forceable by Γ_c. Let $\mathcal{A}_c(v) := \bigcup_{t \in \mathcal{F}_c(v)} t$. Since $\mathcal{A}_c(v) = \mathtt{F}_c \left(\prod_{u \in V(c)\setminus\{v\}} (\chi^*)^{\{u\}} \right)$, it follows that $\mathcal{A}_c(v)$ is always forceable. In fact, it is easy to see that $\mathcal{A}_c(v)$ is the "largest" forceable token on v by Γ_c — in the sense of containing all other forceable tokens as its subsets — due to the monotonicity of $\mathtt{F}_c(\cdot)$.

In k-SAT problems, for any $(v - c)$, it is easy to see that $\mathcal{F}_c(v) = \{*, \mathbf{L}\}$, and $\mathcal{A}_c(v) = *$. In 3-COL problems, for any $(v - c)$, it is easy to see that $\mathcal{F}_c(v) = \{\mathbf{123}, \mathbf{12}, \mathbf{23}, \mathbf{13}\}$, and $\mathcal{A}_c(v) = \mathbf{123}$.

For any $(c - v)$, let $\mathcal{A}_{\sim c}(v)$ be defined by

$$\mathcal{A}_{\sim c}(v) := \bigcap_{b \in C(v)\setminus\{c\}} \mathcal{A}_b(v).$$

Definition 5 (Locally Compatible Constraint): A constraint Γ_c is said to be locally compatible if for any $v \in V(c)$, any forceable token $t_v \in \mathcal{F}_c(v)$, any rectangle $t' \in \mathtt{F}_c^{-1}(t_v)$ on $V(c) \setminus \{v\}$ (where $\mathtt{F}_c^{-1}(t_v)$ is the set of all rectangles $y_{V(c)\setminus\{v\}}$ on $V(c) \setminus \{v\}$ such that $\mathtt{F}_c(y_{V(c)\setminus\{v\}}) = t_v$) and any $u \in V(c) \setminus \{v\}$, it holds that

$$\mathcal{A}_{\sim c}(u) \subseteq \mathtt{F}_c \left(t_v \times t'_{:V(c)\setminus\{u,v\}} \right).$$

We note that the local compatibility of a constraint Γ_c as defined above is not simply a property of Γ_c itself. It also relies on the structure of all constraints that are distance-2 away from Γ_c in the factor graph.

Theorem 5: Let the set of obedience conditionals $\{\omega_v : v \in V\}$ be given, where each $v \in V$ corresponds to a coordinate of a CSP. Let both the MRF of the CSP (that specified via (69), (70) and (71)) and the weighted PTP for the CSP be both parametrized by $\{\omega_v : v \in V\}$. Then if every constraint of the CSP is locally compatible, the SDBP derived from the MRF is equivalent to the weighted PTP, where the correspondence is

$$\rho_{c\to v}^{\mathrm{norm(PTP)}} \leftrightarrow \rho_{c\to v}^{*(\mathrm{SDBP})}.$$

Conversely, if such an equivalence holds for every choice of $\{\omega_v : v \in V\}$, then every constraint of the CSP must be locally compatible.

Alternatively phrased, Theorem 5 suggests that if the state-decoupling condition is satisfied in every iteration of BP, the local compatibility condition on all constraints is the necessary and sufficient condition for weighted PTP to reduce from BP. — We note that Theorem 5 only refers to the equivalence of right messages. It is however straight-forward to verify (as seen in earlier proofs of equivalent results in this paper) that right equivalence implies the summary equivalence.

This theorem answers the question *when* SP is SDBP in a general setting.

Proof:

Following the message-update rule of SDBP,

$$\begin{aligned}
\rho_{c\to v}^{*(\mathrm{SDBP})}(s_{v,c}^R) \quad &\propto \quad \sum_{s^L_{V(c)\setminus\{v\},c}} \left(\left[s^R_{v,c} = \mathtt{F}_c\left(s^L_{V(c)\setminus\{v\},c}\right)\right] \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}^{(\mathrm{SDBP})}\left(s^L_{u,c}, \mathtt{F}_c\left(s^L_{V(c)\setminus\{u\},c}\right)\right)\right) \\
&= \quad \sum_{s^L_{V(c)\setminus\{v\},c}} \left(\left[s^R_{v,c} = \mathtt{F}_c\left(s^L_{V(c)\setminus\{v\},c}\right)\right] \prod_{u\in V(c)\setminus\{v\}} \sum_{s^R_{u,C(u)\setminus\{c\}}} \right. \\
&\quad \omega_u\left(s^L_{u,c} \middle| \left(\bigcap_{b\in C(u)\setminus\{c\}} s^R_{u,b}\right) \cap \mathtt{F}_c\left(s^L_{V(c)\setminus\{u,v\},c} \times s^R_{v,c}\right)\right) \\
&\quad \left. \cdot \prod_{b\in C(u)\setminus\{c\}} \rho_{b\to u}^{*(\mathrm{SDBP})}(s^R_{u,b}) \right)
\end{aligned} \tag{140}$$

Similarly following the message-update rule of weighted PTP, we have

$$\begin{aligned}
\rho_{c\to v}^{\mathrm{norm(PTP)}}(t_{c\to v}) \quad &\propto \quad \sum_{t_{V(c)\setminus\{v\}\to c}} \left(\left[t_{c\to v} = \mathtt{F}_c(t_{V(c)\setminus\{v\}\to c})\right] \prod_{u\in V(c)\setminus\{v\}} \sum_{t_{C(u)\setminus\{c\}\to u}} \right. \\
&\quad \left. \omega_u\left(t_{u\to c} \middle| \bigcap_{b\in C(u)\setminus\{c\}} t_{b\to u}\right) \cdot \left(\prod_{b\in C(u)\setminus\{c\}} \rho_{b\to u}^{\mathrm{norm(PTP)}}(t_{b\to u})\right)\right).
\end{aligned} \tag{141}$$

Identifying every right state $s^R_{v,c}$ in (140) with token $t_{c\to v}$ in (141) and every left state $s^L_{v,c}$ (140) with token $t_{v\to c}$ in (141), the only difference between (140) and (141) is the argument of function ω_u. (We note that since both $\rho^{*(\mathrm{SDBP})}_{c\to v}$ and $\rho^{\mathrm{norm(PTP)}}_{c\to v}$ are normalized, the scaling constant in (140) and (141) are necessarily the same.) We now prove the sufficiency and necessity of the local compatibility condition for the equivalence between $\rho^{\mathrm{norm(PTP)}}_{c\to v}$ and $\rho^{*(\mathrm{SDBP})}_{c\to v}$ via the following chain of two-way implications.

$$\rho^{*(\mathrm{SDBP})}_{c\to v} \leftrightarrow \rho^{\mathrm{norm(PTP)}}_{c\to v}, \forall v \in V(c)$$

$$\Leftrightarrow \quad \omega_u\left(s^L_{u,c}\,\middle|\,\left(\bigcap_{b\in C(u)\setminus\{c\}} s^R_{u,b}\right)\cap \mathtt{F}_c\left(s^L_{V(c)\setminus\{u,v\},c}\times s^R_{v,c}\right)\right) = \omega_u\left(s^L_{u,c}\,\middle|\,\bigcap_{b\in C(u)\setminus\{c\}} s^R_{u,b}\right)$$

$\forall v \in V(c)$ and every $\left(s^R_{v,c}, s^L_{V(c)\setminus\{v\},c}\right)$ in the support of $\left[s^R_{v,c} = \mathtt{F}_c\left(s^L_{V(c)\setminus\{v\},c}\right)\right]$,

$\forall u \in V(c)\setminus\{v\}$ and every choice of $|C(u)\setminus\{c\}|$ tokens on $\{u\}, \left\{s^R_{u,b} : b \in C(u)\setminus\{c\}\right\}$,

with each $s^R_{u,b}$ in the support of $\rho^{(\mathrm{PTP})}_{b\to u}$.

$$\Leftrightarrow \quad \left(\bigcap_{b\in C(u)\setminus\{c\}} s^R_{u,b}\right)\cap \mathtt{F}_c\left(s^L_{V(c)\setminus\{u,v\},c}\times s^R_{v,c}\right) = \bigcap_{b\in C(u)\setminus\{c\}} s^R_{u,b}$$

$\forall v \in V(c)$ and every $\left(s^R_{v,c}, s^L_{V(c)\setminus\{v\},c}\right)$ such that $s^R_{v,c} \in \mathcal{F}_c(v)$ and $s^L_{V(c)\setminus\{v\},c} \in \mathtt{F}_c^{-1}(s^R_{v,c})$,

$\forall u \in V(c)\setminus\{v\}$ and every choice of $|C(u)\setminus\{c\}|$ tokens on $\{u\}, \left\{s^R_{u,b} : b \in C(u)\setminus\{c\}\right\}$,

with each $s^R_{u,b} \in \mathcal{F}_b(u)$.

$$\Leftrightarrow \quad \bigcap_{b\in C(u)\setminus\{c\}} s^R_{u,b} \subseteq \mathtt{F}_c\left(s^L_{V(c)\setminus\{u,v\},c}\times s^R_{v,c}\right)$$

$\forall v \in V(c)$ and every $\left(s^R_{v,c}, s^L_{V(c)\setminus\{v\},c}\right)$ such that $s^R_{v,c} \in \mathcal{F}_c(v)$ and $s^L_{V(c)\setminus\{v\},c} \in \mathtt{F}_c^{-1}(s^R_{v,c})$,

$\forall u \in V(c)\setminus\{v\}$ and every choice of $|C(u)\setminus\{c\}|$ tokens on $\{u\}, \left\{s^R_{u,b} : b \in C(u)\setminus\{c\}\right\}$,

with each $s^R_{u,b} \in \mathcal{F}_b(u)$.

$$\Leftrightarrow \quad \bigcap_{b\in C(u)\setminus\{c\}} \mathcal{A}_b(u) \subseteq \mathtt{F}_c\left(s^L_{V(c)\setminus\{u,v\},c}\times s^R_{v,c}\right)$$

$\forall v \in V(c)$ and every $\left(s^R_{v,c}, s^L_{V(c)\setminus\{v\},c}\right)$ such that $s^R_{v,c} \in \mathcal{F}_c(v)$ and $s^L_{V(c)\setminus\{v\},c} \in \mathtt{F}_c^{-1}(s^R_{v,c})$,

and every $u \in V(c)\setminus\{v\}$.

$$\Leftrightarrow \quad \mathcal{A}_{\sim c}(u) \subseteq \mathtt{F}_c\left(s^L_{V(c)\setminus\{u,v\},c} \times s^R_{v,c}\right),$$

$\forall v \in V(c)$ and every $\left(s^R_{v,c}, s^L_{V(c)\setminus\{v\},c}\right)$ such that $s^R_{v,c} \in \mathcal{F}_c(v)$ and $s^L_{V(c)\setminus\{v\},c} \in \mathtt{F}_c^{-1}(s^R_{v,c})$,

and every $u \in V(c) \setminus \{v\}$.

$\Leftrightarrow$ Constraint Γ_c is locally compatible.

Thus

$$\rho^{\text{norm(PTP)}}_{c\to v} \leftrightarrow \rho^{*(\text{SDBP})}_{c\to v}, \text{ for every } (x_v, \Gamma_c) \in E(G)$$

$$\Leftrightarrow \quad \text{Every constraint } \Gamma_c \text{ is locally compatible.}$$

■

Now it is easy to verify that for both k-SAT and 3-COL problems, the fact that PTP or weighted PTP can be reduced from BP with state-decoupling condition imposed is due to the fact that every constraint is locally compatible.

For k-SAT problems, as noted earlier, $\mathcal{F}_c(v) = \{\mathbf{L}, *\}$. If we pick t_v to be either token from $\mathcal{F}_c(v)$, then for any $t' \in \mathtt{F}_c^{-1}(t_v)$ and any $u \in V(c)\setminus\{v\}$, it can be verified that $\mathtt{F}_c\left(t'_{:V(c)\setminus\{u,v\}} \times t_v\right) = *$. This makes $\mathcal{A}_{\sim c}(u) \subseteq \mathtt{F}_c\left(t'_{:V(c)\setminus\{u,v\}} \times t_v\right)$ always satisfied, independent of the factor graph structure of the problem instance.

For 3-COL problems, as noted earlier, we see $\mathcal{F}_c(v) = \{\mathbf{123}, \mathbf{12}, \mathbf{23}, \mathbf{13}\}$. Suppose that u is the only other coordinate (except v) that is involved in constraint Γ_c. If we pick t_v to be any token from $\mathcal{F}_c(v)$, then $\mathtt{F}_c^u(t_v) = \mathbf{123}$. This again makes $\mathcal{A}_{\sim c}(u) \subseteq \mathtt{F}_c^u(t_v)$ always satisfied, independent of the factor graph structure of the problem instance.

That is, in both k-SAT and 3-COL problems, the structure of each local constraint *alone* guarantees the local compatibility condition satisfied by every constraint, irrespective of how a constraint interacts with other constraints (that are distant 2 apart) as is generally required in the local compatibility condition. We generalize this fact in the following corollary — immediately following Theorem 5 — which provides a sufficient condition for SDBP to reduce to PTP without relying on the interaction of neighboring constraints. For CSPs constructed with generic local constraint by random factor graph structure, the corollary may turn out to be useful.

Corollary 1: Let both the MRF of the CSP (specified via (69), (70) and (71)) and the weighted PTP for the CSP be parametrized by the same $\{\omega_v : v \in V\}$. Suppose that every constraint Γ_c

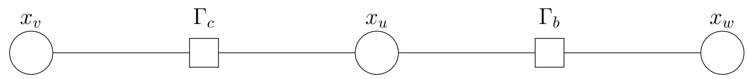


Fig. 5. A portion of a factor graph G.

is such that for any $v \in V(c)$, any forceable token $t_v \in \mathcal{F}_c(v)$, any rectangle $t' \in \mathtt{F}_c^{-1}(t_v)$ on $V(c) \setminus \{v\}$, and any $u \in V(c) \setminus \{v\}$, it holds that

$$\mathtt{F}_c\left(t_v \times t'_{:V(c)\setminus\{u,v\}}\right) = (\chi^*)^{\{v\}}.$$

Then SDBP derived from the MRF is equivalent to weighted PTP, where the correspondence is

$$\rho_{c\to v}^{\mathrm{norm(PTP)}} \leftrightarrow \rho_{c\to v}^{*\mathrm{(SDBP)}}.$$

For completeness, we conclude this section by constructing an example of CSP in which the local compatibility condition is not satisfied by every constraint.

Suppose that Γ_c and Γ_b are two of the constraints defining a CSP, and the factor graph representing the CSP locally obeys the structure shown in Figure 5. Suppose that each variable of the CSP has alphabet $\chi = \{0, 1, 2\}$ and that Γ_c is defined as $\Gamma_c := \{(0_v, 0_u), (0_v, 1_u), (1_v, 2_u), (2_v, 2_u)\}$. Suppose that Γ_b is defined as $\Gamma_b := \{(0_u, 0_w), (1_u, 1_w), (2_u, 1_w)\}$. Note that $\mathcal{F}_c(v) = \{\mathbf{0}_v, \mathbf{12}_v, \mathbf{012}_v\}$, and it is easy to verify that $\mathcal{A}_{\sim c}(u) = \mathcal{A}_b(u) = \mathtt{F}_b\left(\mathbf{012}_w\right) = \mathbf{012}_u$. Now if we pick $t_v = \mathbf{0}_v$, then we have $\mathcal{A}_{\sim c}(u) \nsubseteq \mathtt{F}_c(t_v) = \mathbf{01}_u$. Thus constraint Γ_c is not locally compatible, and following Theorem 5, PTP or weighted PTP can not be reduced from SDBP for this CSP.

With this example, we see that it is not always the case that SDBP is SP.

VII. Concluding Remarks

In this paper, we study the question whether SP algorithms (non-weighted and weighted) are special cases of BP for general constraint satisfaction problems.

The first contribution of this paper is a simple formulation of SP algorithms for general CSPs as the weighted PTP algorithm. An advantage of this formulation is that it has a probabilistically interpretable update rule which allows SP algorithms to be developed for arbitrary CSPs.

The second and main contribution of this paper is the answer to the titular question in the most general context. We show that in general, SP algorithms can not be reduced from the BP algorithm derived from the MRF formalism in the style of [15] and [17]. Such a reduction is only possible for certain special cases where the notions of state-decoupling condition and local compatibility condition are both satisfied.

It is worth noting that our answer to whether SP is BP is only restricted to the MRF formalism in the style of [15] or [17]. Although this restriction is not completely satisfactory, it appears to us that such an MRF formalism is the most natural in light of the natural correspondence between the states in the MRF and the SP messages (namely that left states correspond to the "intentions" of variables and right states correspond to the "commands" of the constraints). An additional and perhaps even stronger justification of this MRF is its combinatorial descriptive power as is elaborated in [15] for k-SAT problems, which — using the terminology of this paper — captures the connectivity of the solution in the space of all "rectangles". In fact, we conjecture that further investigation of this perspective may provide useful insights into the algorithm design for solving hard instances of CSPs, whether or not SP or BP is considered as the choice of algorithms. [8]

Further we note that the BP algorithm has been understood as a special case of Generalized Belief Propagation (GBP) [20]. In that perspective, BP may be derived from iterative minimization of the Bethe-approximation of the notion of free energy [20]. The framework of GBP allows a variety of ways (unified under the notion of "region graphs") to approximate the free energy whereby leading to a much richer family of BP-like algorithms. Given the results of this paper, one may not want to exclude the possibility that certain choice of free-energy approximation allows the corresponding GBP to reduce to SP algorithms for general CSPs. Research along that direction may still be of interest.

As the final remark, however, the authors of this paper would like to raise a philosophical question, in light of the simplicity in the (weighted) PTP formulation of SP and, in contrast, the complexity involved in reducing BP to SP: Should we attempt to seek a complicated explanation for a simple algorithm? Does the simplicity of SP (understood in terms of weighted PTP) imply a more natural, simpler but very different underlying graphical model — beyond MRF — that may better explain SP?

[8] In [15], under the MRF formalism, Gibbs sampling-based approach has also been presented as an algorithm for solving random k-SAT problems.

Appendix

We now present some results concerning the dynamics of SP, based on the formulation of PTP and weighted PTP. These results, although rather elementary, should help provide intuitions regarding what PTP is doing in solving a CSP. We will start with the deterministic precursor of PTP, DTP.

A. On the Dynamics of DTP

We will refer to a subgraph H of factor graph G as a *factor-subgraph* of G if for every constraint vertex Γ_c in H, all neighboring variable vertices of Γ_c in G are also in H. It is apparent that factor-subgraph H is a factor graph representing a CSP involving precisely a subset of the constraints in G. We will denote by $C[H]$ the index set of all constraint vertices in H, by $V[H]$ the index set of all variable vertices in H, and by Γ_H the set of all assignments on $V[H]$ that satisfy every constraint Γ_c, $c \in C[H]$.

If factor-subgraph H is a tree, it is also referred to as a *factor tree* of G. For any factor tree T of G, we will denote by $L[T]$ the index set of all leaf vertices of T. Since we have assumed that factor graph G contains no degree-1 constraint vertices, it is necessary that the leaf vertices of any factor tree T of G are all variable vertices, i.e., that $L[T]$ contains no index of any constraint vertex.

Suppose that T is a factor tree of factor graph G, $U \subset V[T]$, and $v \in V[T] \setminus U$. For any rectangle t_U on U, define

$$\mathtt{F}_T^{U\rightarrow v}(t_U) := \left(\left(t_U \times (\chi^*)^{V[T]\setminus U}\right) \cap \Gamma_T\right)_{:\{v\}}.$$

It is easy to see that function $\mathtt{F}_T^{U\rightarrow v}(\cdot)$ reduces to $\mathtt{F}_c^{v}(\cdot)$ introduced earlier, when T contains a single factor and U is $V(c) \setminus \{v\}$.

Given a factor tree T of G and two vertices in T indexed by a and b respectively, we will introduce another notation of message index, $a \xrightarrow{T} b$, which indexes the message sent by the vertex with index a along its only edge that is on the path (in T) leading to the vertex with index b. For example, suppose that in factor tree T, constraint vertex Γ_c has a neighbor of x_u and is on the path from x_u to x_v in T, then message index $u \xrightarrow{T} v$ is equivalent to $u \rightarrow c$, and $t_{u \xrightarrow{T} v}$ is equivalent to $t_{u\rightarrow c}$.

A factor tree T of G will be referred to as a (v,l)-tree of G if the variable vertex x_v is in T, every leaf vertex in T is distance $2l$ from vertex x_v, and all vertices in G that have distance to x_v no larger than $2l$ are contained in T. It is clear that given G, $v \in V$ and a positive integer l, if a (v,l)-tree of G exists, it is unique. We therefore denote it by T_v^l.

Given T_v^l of factor graph G, factor tree T_{v-c}^l of G is the subgraph of T_v^l induced by vertex x_v and all vertices of T_v^l whose paths to x_v (in T_v^l) traverse through vertex Γ_c. On the other hand, factor tree $T_{v\neq c}^l$ is the subgraph of T_v^l induced by vertex x_v and all vertices of T_v^l whose paths to x_v (in T_v^l) *do not* traverse through vertex Γ_c.

In what follows, we will use superscript (l) on a message to refer to the message in the l^{th} iteration.

Proposition 2: Suppose that $l \geq 1$ and that factor tree T_v^l of factor graph G exists. Then in iteration l of DTP,

$$t_{c\to v}^{(l)} = \mathtt{F}_{T_{v-c}^l}^{L[T_{v-c}^l]\to v} \left(\prod_{u \in L[T_{v-c}^l]} t^{(1)}_{u \xrightarrow{T_{v-c}^l} v} \right).$$

Proof: We will prove this result by induction on l.

For the base case, we have

$$\begin{aligned} t_{c\to v}^{(1)} &= \mathtt{F}_c^v \left(\prod_{u \in V(c)\setminus\{v\}} t_{u\to c}^{(1)} \right) \\ &= \mathtt{F}_{T_{v-c}^1}^{L[T_{v-c}^1]\to v} \left(\prod_{u \in L[T_{v-c}^1]} t^{(1)}_{u \xrightarrow{T_{v-c}^1} v} \right). \end{aligned}$$

As the inductive hypothesis, suppose that the result of this proposition holds for a given iteration number $l \geq 1$. This implies specifically that for every $u \in V(c) \setminus \{v\}$ and every $b \in C(u) \setminus \{c\}$,

$$t_{b\to u}^{(l)} = \mathtt{F}_{T_{u-b}^l}^{L[T_{u-b}^l]\to u} \left(\prod_{w \in L[T_{u-b}^l]} t^{(1)}_{w \xrightarrow{T_{u-b}^l} u} \right).$$

Then

$$\begin{aligned} t_{u\to c}^{(l+1)} &= \bigcap_{b\in C(u)\setminus\{c\}} t_{b\to u}^{(l)} \\ &= \bigcap_{b\in C(u)\setminus\{c\}} \mathtt{F}_{T_{u-b}^{l}}^{L[T_{u-b}^{l}]\to u}\left(\prod_{w\in L[T_{u-b}^{l}]} t^{(1)}_{w \xrightarrow{T_{u-b}^{l}} u}\right) \\ &= \mathtt{F}_{T_{u\neq c}^{l}}^{L[T_{u\neq c}^{l}]\to u}\left(\prod_{w\in L[T_{u\neq c}^{l}]} t^{(1)}_{w \xrightarrow{T_{u\neq c}^{l}} u}\right). \end{aligned}$$

Finally,

$$\begin{aligned} t_{c\to v}^{(l+1)} &= \mathtt{F}_c^v\left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c}^{(l+1)}\right) \\ &= \mathtt{F}_c^v\left(\prod_{u\in V(c)\setminus\{v\}} \mathtt{F}_{T_{u\neq c}^{l}}^{L[T_{u\neq c}^{l}]\to u}\left(\prod_{w\in L[T_{u\neq c}^{l}]} t^{(1)}_{w \xrightarrow{T_{u\neq c}^{l}} u}\right)\right) \\ &= \mathtt{F}_{T_{v-c}^{l+1}}^{L[T_{v-c}^{l+1}]\to v}\left(\prod_{w\in L[T_{v-c}^{l+1}]} t^{(1)}_{w \xrightarrow{T_{v-c}^{l+1}} v}\right). \end{aligned}$$

This completes the proof. ■

Translating this results to summary tokens, the following result can be obtained immediately.

Corollary 2: Suppose that $l \geq 1$ and that factor tree T_v^l of factor graph G exists. Then in iteration l of DTP,

$$t_v^{(l)} = \mathtt{F}_{T_v^l}^{L[T_v^l]\to v}\left(\prod_{u\in L[T_v^l]} t^{(1)}_{u \xrightarrow{T_v^l} v}\right).$$

The implication of this result is that on factor graph with sufficiently large girth, DTP is in fact very well-behaved: the summary token at any variable x_v in iteration l depends precisely on the initial tokens passed by variables that are $2l$ away from x_v. Specifically, one may view those tokens form a rectangle on $L[T_v^l]$, and the summary token at x_v in iteration l is precisely the set of all assignments on $\{v\}$ that can make $\Gamma_{T_v^l}$ satisfied, given the assignment on $L[T_v^l]$ is from that rectangle.

Now we develop some results of DTP that require no "local cycle-freeness" in the factor graph.

Lemma 15: At every $v \in V$ and for any l,

$$t_v^{(l)} = \bigcap_{c \in C(v)} t_{v \to c}^{(l+1)}.$$

Proof: Suppose that $x_v \in t_v^{(l)}$. Then $x_v \in t_{c \to v}^{(l)}$ for every $c \in C(v)$, by the definition of summary messages. It follows that $x_v \in t_{v \to c}^{(l+1)}$ for every $c \in C(v)$. Then $x_v \in \bigcap_{c \in C(v)} t_{v \to c}^{(l+1)}$. This shows that $t_v^{(l)} \subseteq \bigcap_{c \in C(v)} t_{v \to c}^{(l+1)}$.

On the other hand, suppose that $x_v \in \bigcap_{c \in C(v)} t_{v \to c}^{(l+1)}$. Then $x_v \in t_{v \to c}^{(l+1)} = \bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}^{(l)}$, for every $c \in C(v)$. It follows that $x_v \in t_{b \to v}^{(l)}$ for every $b \in C(v)$, giving rise to that $x_v \in \bigcap_{b \in C(v)} t_{b \to v}^{(l)} = t_v^{(l)}$. Thus $\bigcap_{c \in C(v)} t_{v \to c}^{(l+1)} \subseteq t_v^{(l)}$.

Therefore $t_v^{(l)} = \bigcap_{c \in C(v)} t_{v \to c}^{(l+1)}$. ■

Lemma 16: Suppose that $\hat{x}_V$ is a satisfying assignment on V, namely that $\hat{x}_V$ satisfies (1). If $\hat{x}_V \in \prod\limits_{v \in V} \bigcap\limits_{c \in C(v)} t_{v \to c}^{(l)}$ in some iteration l, then $\hat{x}_V \in \prod\limits_{v \in V} t_v^{(l)}$.

Proof: The fact that $\hat{x}_V \in \prod\limits_{v \in V} \bigcap\limits_{c \in C(v)} t_{v \to c}^{(l)}$ implies that for every $v \in V$ and $c \in C(v)$, $\hat{x}_{V:\{v\}} \in \bigcap\limits_{c \in C(v)} t_{v \to c}^{(l)} \subseteq t_{v \to c}^{(l)}$, and hence via the "monotonicity" of function $\mathtt{F}_c$, $\mathtt{F}_c\left(\{\hat{x}_{V:V(c)\setminus\{v\}}\}\right) \subseteq \mathtt{F}_c\left(\prod\limits_{u \in V(c)\setminus\{v\}} t_{u \to c}^{(l)}\right) = t_{c \to v}^{(l)}$. Incorporating that $\hat{x}_V$ is a satisfying assignment, we see that $\hat{x}_{V:\{v\}} \in \mathtt{F}_c\left(\{\hat{x}_{V:V(c)\setminus\{v\}}\}\right) \subseteq t_{c \to v}^{(l)}$, for every $v \in V$ and $c \in C(v)$. Thus $\hat{x}_{V:\{v\}} \in \bigcap\limits_{c \in C(v)} t_{c \to v}^{(l)} = t_v^{(l)}$. It then follows that $\hat{x}_V \in \prod\limits_{v \in V} t_v^{(l)}$. ■

Proposition 3: Suppose that $\hat{x}_V$ is a satisfying assignment and that the initialization of DTP is such that $\hat{x}_{V:\{v\}} \in t_{v \to c}^{(1)}$ for every $v \in V$ and $c \in C(v)$. Then in any iteration l, the rectangle $\prod\limits_{v \in V} t_v^{(l)}$ formed by the summary tokens contains $\hat{x}_V$.

Proof: At iteration 1, the fact that $\hat{x}_{V:\{v\}} \in t_{v \to c}^{(1)}$ for every $v \in V$ and $c \in C(v)$ implies that $\hat{x}_V \in \prod_{v \in V} \bigcap_{c \in C(v)} t_{v \to c}^{(1)}$. Followed by Lemma 16, we have $\hat{x}_V \in \prod_{v \in V} t_v^{(1)}$.

As the inductive hypothesis, suppose we have $\hat{x}_V \in \prod_{v \in V} t_v^{(l)}$ at iteration l. At iteration $l+1$, followed by Lemma 15, we have $\hat{x}_V \in \prod_{v \in V} \bigcap_{c \in C(v)} t_{v \to c}^{(l+1)}$. Then by Lemma 16, $\hat{x}_V \in \prod_{v \in V} t_v^{(t+1)}$.

Therefore, this proposition is proved by induction. ■

At this end, we have shown that if DTP is initialized to "containing" a satisfying assignment, then this assignment is contained in the rectangle formed by the summary tokens in all iterations. That is, the solution of the CSP will never get "lost" during DTP iteration provided that it is contained in the initial rectangle. This result (Proposition 3) and Corollary 2 presented earlier will become useful when we discuss the dynamics of PTP.

B. On the Dynamics of PTP and Weighted PTP

We now turn our attention to (non-weighted) PTP.

Denote by G_v^l the factor-subgraph of G which contains all factors whose messages have propagated to variable x_v by the end of PTP iteration l. That is, G_v^l is the factor-subgraph of G that contains variable vertex x_v and all vertices whose distances to x_v are no larger than $2l$. It is apparent that if G_v^l is a tree, then it is the (v, l) factor tree T_v^l.

Let l^* be the smallest l such that at least for one $v \in V$, T_v^l does not exist. Denote $m_v(l) := \left|\left(\Gamma_{G_v^l}\right)_{:\{v\}}\right|$. That is, $m_v(l)$ is the number of assignments of variable x_v that can make all constraints in G_v^l satisfied. Clearly, $m_v(l)$ is a non-increasing function of l.

We will first restrict the CSP to a "single-solution CSP", i.e., having exactly one satisfying assignment. We will denote this assignment on V by $\hat{x}_V$.

Let $\hat{l}$ be the smallest l for which $\min_v m_v(l) = 1$. It is worth noting that such $\hat{l}$ exists since the CSP has precisely one solution. Let $\hat{v}$ satisfy $m_{\hat{v}}(\hat{l}) = 1$.

Proposition 4: Let factor graph G represent a single-solution CSP. Suppose that the initialization of PTP is such that every left message $\lambda_{v \to c}^{(1)}(t)$ is strictly positive for every $t \in (\chi^*)^{\{v\}}$. If $\hat{l} < l^*$, then

$$\mu_{\hat{v}}^{\text{norm }(\hat{l})}(t) = [t = \{\hat{x}_{V:\{\hat{v}\}}\}].$$

Proof: This result relies on Corollary 2.

First, $\hat{l} < l^*$ implies that $(\hat{v}, \hat{l})$ factor tree $T_{\hat{v}}^{\hat{l}}$ exists. Then by Corollary 2, if DTP is initialized such that the tokens sent from the leaves of $T_{\hat{v}}^{\hat{l}}$ form $\prod\limits_{u \in L[T_{\hat{v}}^{\hat{l}}]} t^{(1)}_{u \xrightarrow{T_{\hat{v}}^{\hat{l}}} \hat{v}}$, then the summary token at v in the $\hat{l}^{\text{th}}$ iteration is $\mathtt{F}_{T_{\hat{v}}^{\hat{l}}}^{L[T_{\hat{v}}^{\hat{l}}] \to \hat{v}} \left(\prod\limits_{u \in L[T_{\hat{v}}^{\hat{l}}]} t^{(1)}_{u \xrightarrow{T_{\hat{v}}^{\hat{l}}} \hat{v}} \right)$.

Since $\hat{v}$ satisfies $m_{\hat{v}}(\hat{l}) = 1$, it is necessary that $\mathtt{F}_{T_{\hat{v}}^{\hat{l}}}^{L[T_{\hat{v}}^{\hat{l}}] \to \hat{v}} \left(\prod\limits_{u \in L[T_{\hat{v}}^{\hat{l}}]} t^{(1)}_{u \xrightarrow{T_{\hat{v}}^{\hat{l}}} \hat{v}} \right)$ is either token $\{\hat{x}_{V:\{v\}}\}$ or $\emptyset$, which depends on the rectangle initialized.

Now PTP on $T_{\hat{v}}^{\hat{l}}$, with respect to $x_{\hat{v}}$, may be understood as initializing a *random* rectangle on $L[T_{\hat{v}}^{\hat{l}}]$ (the distribution of which is characterized by the product of the initial messages), transforming the random rectangle to random token on $\hat{v}$ via a functional mapping $\mathtt{F}_{T_{\hat{v}}^{\hat{l}}}^{L[T_{\hat{v}}^{\hat{l}}]\rightarrow\hat{v}}(\cdot)$, and conditioning on the resulting token being valid (non-empty set). The fact that initial messages of PTP are strictly positive assures that every rectangle on $L[T_{\hat{v}}^{\hat{l}}]$ has non-zero probability during initialization. After conditioning on the resulting token being valid, the token $\emptyset$ is removed from the allowed realization of the resulting token and thus the resulting token equals $\{\hat{x}_{V:\{v\}}\}$ with probability 1. This completes the proof. ■

This result and its proof can be easily extended to a somewhat larger family of CSPs each containing multiple solutions, as shown in the next proposition.

Proposition 5: Suppose that in the CSP, there exist a coordinate $\hat{v} \in V$ and an assignment $\hat{x}_{\hat{v}} \in (\chi^*)^{\{v\}}$ such that every satisfying configuration $\tilde{x}_V \in \Gamma$ satisfies $\tilde{x}_{V:\{v\}} = \hat{x}_v$. If for some integer $\hat{l}$, $T_{\hat{v}}^{\hat{l}}$ exists and $m_{\hat{v}}(\hat{l}) = 1$, then

$$\mu_{\hat{v}}^{\text{norm }(\hat{l})}(t) = [t = \{\hat{x}_{\hat{v}}\}].$$

The proof is similar to that for proposition 4, which essentially relies on Corollary 2 and that the local tree rooted at $\hat{v}$ is large enough. Skipping the proof, we note that Proposition 4 may be viewed as a special case of Proposition 5.

Based on the results above, we provide some remarks concerning the dynamics of PTP and argue intuitively how it solves a CSP.

1) Similar to what was argued in the proof of Proposition 4, the key insight regarding what PTP is doing is that PTP updates a *random* rectangle whose sides are distributed independently.

   At the initialization stage, PTP defines a random rectangle on V, where the sides of the random rectangles are treated as independent random variables. In every iteration, PTP maps this random rectangle to a new random rectangle in the following steps.

   a) Apply a functional mapping defined by the right-message update rule and the left-message update rule.

   b) Eliminate the resulting empty rectangles (via conditioning on that each side of the resulting random rectangle is not the empty set and re-normalization).

c) Take the marginal distribution of the resulting random rectangle on each side variable, and treat all sides as being independent random variables. This defines a new random rectangle.

PTP iterates over these steps to continuously update the random rectangle.

2) For single-solution CSPs, based on Proposition 4, if the girth of the graph is large enough, at least one side of the new rectangle, after some iterations, becomes deterministic, namely the singleton set containing the correct assignment for that variable. This would allow the decimation procedure to fix this variable to the correct assignment and reduce the problem. Similar results hold for CSPs having more than one solutions but in which all solutions share a single assignment on some coordinate. By Proposition 5, in this case, when the local tree rooted at that variable is sufficiently large, PTP will find that variable and its correct assignment. Of course, the condition of Proposition 4 and that of Proposition 5, namely that there is a sufficiently large local tree rooted at a variable and that the variable only has one correct assignment, may not hold in reality. As a consequence, no side of the random rectangle is deterministically a singleton. In that case, the decimation procedure must deal with this ambiguity — resulted from non-ideal factor graph structure and the complexity of the solution space — and make a good guess to fix a variable.
3) Proposition 4 and Proposition 2 also suggest that when the graph has large girth (and when the solutions share one common assignment on some coordinate), as PTP iterates, the rectangles containing no solutions will be gradually removed from the sample space of the random rectangle.
4) Proposition 3 implies that regardless of cycle structure of the graph, all solution-containing rectangles will be kept (possibly in a form of combining each other) over PTP iterations.
5) Combining 3) and 4) above, one may view each PTP iteration as performing a “filtering” operation on the distribution of the random rectangle. As the distribution of the random rectangle evolves, the probability mass moves gradually to one biased to some solution-containing rectangles. When the graph has large girth and some coordinate is in a “favorable” position (in a sense combining its location in the graph and its role in the solution space), the summary message at this coordinate may become more deterministically biased to a singleton token, making decimation possible.

Finally, we briefly remark on weighted PTP.

Similar to PTP, weighted PTP also updates a random rectangle. However, instead of using a functional mapping, in step a) of the above procedure, it uses a conditional distribution. By examining the form of the obedience conditionals, it is intuitive that comparing with PTP, weighted PTP shifts the distribution of each side of the random rectangle more towards "smaller" tokens on each coordinate. (Here t_v is said to be smaller than t'_v if $t_v \subset t'_v$.) This provides the algorithm better opportunity to lead to some side of the random rectangle more deterministically biased to a singleton.

REFERENCES

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

[2] 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.

[3] 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, pp. 036702, 2003.

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

[5] 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.

[6] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[7] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Matco, CA, 1st edition, 1988.

[8] 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.

[9] Jr. G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.

[10] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo codes," *IEEE Trans. Comm.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[11] D. J. C. R. J. McEliece, D.J. C. MacKay and J. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE JSAC*, vol. 16, no. 2, pp. 140–152, 1998.

[12] F.Jelinek L.Bahl, J.Cocke and J.Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[13] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 325–343, Mar. 2000.

[14] A. Braunstein and R. Zeccchina, "Survey propagation as local equilibrium equations," *J. Stat. Mech.*, June 2004.

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

[16] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*, American Mathematical Society, Providence, RI, 1980.

[17] 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.

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

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

[20] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2282–2312, July 2005.