LETTER *Special Section on Foundations of Computer Science*

Survey Propagation as "Probabilistic Token Passing"

Ronghui TU[†a)], **Yongyi MAO**[†b)], ***Nonmembers, and*** **Jiying ZHAO**[†c)], ***Member***

SUMMARY In this paper, we present a clean and simple formulation of survey propagation (SP) for constraint-satisfaction problems as "probabilistic token passing". The result shows the importance of extending variable alphabets to their power sets in designing SP algorithms.
key words: *constraint-satisfaction problem, survey propagation, graph coloring, message-passing*

1. Introduction

Survey propagation (SP) [1] is a heuristic message-passing algorithm proposed recently for solving the classical NP-complete k-SAT problems. In [1], SP was shown as the first efficient solver for these problems even in the well-known hard regime. This celebrated discovery has since motivated the application of SP to other hard constraint-satisfaction problems (CSPs), such as the graph-coloring (or q-COL) problems [2], as well as problems in source coding [3] and channel coding [4], where great successes are demonstrated.

Since its discovery, much research attention has been attracted to a deeper understanding of the algorithmic nature of SP. In the context of k-SAT problems, it has been shown and perhaps widely accepted that SP may be interpreted intuitively as propagating "warnings" in a probabilistic manner on the factor-graph representation of the problem instance [5]. This understanding allows one to craft the SP algorithm for some other CSPs, including the coloring problems.

As is derived from statistical physics, SP in most literature has been formulated using the language of physics. Although for k-SAT problems, SP has been shown to be an instance of the *belief propagation* (BP) algorithm [6]–[8] well known in the communities of error control coding and artificial intelligence, the connection between SP and BP for gernral CSPs is yet to be fully clarified. Additionally, lacking simple mathematical formulations in the existing literature makes SP even harder for the wider community of information scientists. In this paper, we present a clean and simple formulation of SP for general CSPs, where SP is understood as "probabilistic token passing". Here a "token" has a precise mathematical definition, namely, a subset of the variable alphabet. This result asserts that extending variable alphabets to their *power sets* plays an essential role in designing SP algorithms.

In this paper, we present such an interpretation of SP for arbitrary CSPs, where the graph-coloring problem is taken as a running example.

2. Constraint-Satisfaction Problems and SP

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

Let C be another finite set indexing a set of constraints $\{\Gamma_c : c \in C\}$, the form of which will be specified subsequently. For each $c \in C$, let $V(c)$ be some subset of V, indexing the set of variables constrained by Γ_c. Symmetrically, for each $v \in V$, we will denote the set $\{c : v \in V(c)\}$ by $C(v)$, namely, $C(v)$ indexes the set of all constraints involving variable x_v. Since each constraint Γ_c applies only on variables $x_{V(c)}$, we will identify constraint Γ_c as a subset of the Cartesian product $\chi_{V(c)}$. Thus a constraint-satisfaction problem (CSP) may be specified by $(V, C, \{\chi_v : v \in V\}, \{V(c) : c \in C\}, \{\Gamma_c : c \in C\})$, with the objective of finding a solution for equation

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

Here the notation $[P]$, for any Boolean proposition P, is the Iverson's convention [9], namely, evaluates to 1 if P, and to 0 otherwise. Clearly, (1) can be represented by a factor graph [9], with variable vertices indexed by V and function vertices indexed by C.

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

SP has been developed for several classes of CSPs as a message-passing algorithm on the factor graph representing the problem. For those problems, messages are passed between variable vertices and constraint vertices, where the message passed from or to a variable x_v is a function on set $\chi_v \cup \{*\}$. The addition of the "joker" symbol $(*)$ to variable alphabet χ_v plays an important role in SP for the stud-

Manuscript received March 6, 2007.
[†]The authors are with the School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave., Ottawa, Ontario, Canada K1N 6N5.
a) E-mail: rtu@site.uottawa.ca
b) E-mail: yymao@site.uottawa.ca
c) E-mail: jyzhao@site.uottawa.ca
DOI: 10.1093/ietisy/e91–d.2.231

ied problems, where x_v equal to the joker indicates that it is free to take any value from its original alphabet, and that x_v equal to a non-joker symbol indicates that it is constrained to taking the designated value. In hard k-SAT problems, it is shown that the "joker" symbol connects the satisfying configurations, which would otherwise form a large number of disconnected "clusters", making local search strategies fail.

For 3-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 (or equivalently, messages are passed on graph (Δ, Ξ)). Following [2], a compact version of SP message-passing rule is given as follows, where the message passed from variable x_u to variable x_v is a quadruplet of real numbers $(\eta^1_{u\to v}, \eta^2_{u\to v}, \eta^3_{u\to v}, \eta^*_{u\to v})$.

$$\eta^r_{u\to v} = \frac{\prod_w (1-\eta^r_{w\to u}) - \sum_{p\neq r}\prod_w(\eta^*_{w\to u}+\eta^p_{w\to u}) + \prod_w \eta^*_{w\to u}}{\sum_p \prod_w (1-\eta^p_{w\to u}) - \sum_p \prod_w (\eta^*_{w\to u}+\eta^p_{w\to u}) + \prod_w \eta^*_{w\to u}} \tag{2}$$

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

$$\eta^*_{u\to v} = 1 - \sum_p \eta^p_{u\to v} \tag{3}$$

The SP messages are usually initialized randomly. Upon convergence, SP computes a "summary message" at each variable x_v, which may be interpreted as the probability or "bias" of each symbol in $\chi_v \cup \{*\}$. A *decimation* procedure is usually followed, where a variable is fixed to a symbol in χ_v if it is highly "biased" to this symbol. After the decimation procedure, the problem is then simplified and SP is applied again. This process iterates until the reduced problem is simple enough for a local search algorithm.

3. SP as Probabilistic Token Passing

For each variable x_v in a given problem, we define an *extended alphabet* χ^*_v as the power set of χ_v (i.e., $\chi^*_v = \{a : a \subseteq \chi_v\}$). For 3-COL problems, χ^*_v is then the set $\{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$ for every v. For each non-empty element t of χ^*_v, we also write it as a string containing the elements of t. For example, we may write $\{1,2\}$ as 12. For any subset $U \subseteq V$, a configuration $y_U := \{y_v \in \chi^*_v : v \in U\}$ is referred to as a *rectangle* on U and understood as the Cartesian product $\prod_{v\in U} y_v$. We note that if any component y_v in the Cartesian product is $\emptyset$, then rectangle y_U is $\emptyset$. We denote by χ^*_U the set of all rectangles on U, and clearly $\chi^*_U = \prod_{v\in U}\chi^*_v$.

For any U and S with $S \subset U \subseteq V$ and any $\Omega \subseteq \chi_U$, we denote $\Omega_{|S} := \{\alpha \in \chi_S : (\alpha, \beta) \in \Omega \text{ for some } \beta \in \chi_{U\setminus S}\}$, that is, $\Omega_{|S}$ is the *projection* of Ω on S.

Given an index $v \in V$ and a configuration $y_{V(c)\setminus\{v\}}$, we define $\mathrm{F}_c(y_{V(c)\setminus\{v\}}) := ((y_{V(c)\setminus\{v\}} \times \chi_v) \cap \Gamma_c)_{|\{v\}}$. That is, $\mathrm{F}_c(y_{V(c)\setminus\{v\}})$ is the largest subset of χ_v in which every element, when paired with some sequence in $\chi_{V(c)\setminus\{v\}}$, makes constraint Γ_c satisfied.

Given a CSP with alphabets extended, we define the *deterministic token-passing algorithm* on the factor-graph representation of the CSP as follows. Tokens are passed along the edges of the factor graph and the token passed from and to each variable x_v is an element of χ^*_v. For a pair of neighboring vertices x_v and Γ_c on the factor graph, the token $t_{v\to c}$ passed from variable x_v to constraint Γ_c depends on all incoming tokens passed to x_v except that passed from Γ_c. Similarly, the token $t_{c\to v}$ passed from constraint Γ_c to variable x_v depends on all incoming tokens passed to Γ_c except that passed from x_v. The token passing rules are given as follows.

$$t_{v\to c} = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \tag{4}$$

$$t_{c\to v} = \mathrm{F}_c\left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c}\right). \tag{5}$$

This algorithm can be simply extended to the *probabilistic token passing (PTP) algorithm* on the same factor graph, where token t passed along each edge is treated as a random variable not allowed to be $\emptyset$. That is, in PTP, instead of passing token t on an edge, the passed message is the distribution of t *conditioned on* $t \neq \emptyset$. Specifically, it is assumed that 1) all tokens passed to a given vertex are independent; 2) the distribution of each incoming token is the message associated with the token; 3) the distribution of an outgoing token is induced by the distributions of the incoming tokens where induction is according to functional dependency of the outgoing token on the incoming tokens specified in (4) and (5).

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

$$\lambda_{v\to c}(t_{v\to c}) = \sum_{t_{v\to c}=\bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}} \left(\prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})\right) \tag{6}$$

$$\rho_{c\to v}(t_{c\to v}) = \sum_{t_{c\to v}=\mathrm{F}_c(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c})} \left(\prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(t_{u\to c})\right). \tag{7}$$

We note that in (6) and (7), $t_{v\to c}$ and $t_{c\to v}$ range over all elements of χ^*_v except $\emptyset$.

On 3-COL problems, PTP, like SP, can be made more compact. However, instead of passing messages between variable vertices, the PTP messages more naturally reduce to messages passed between constraint vertices that are distance 2 apart. — Note that for any two constraint vertices Γ_c

and Γ_d, there is a unique variable vertex x_v, for which vertices Γ_c, x_v, and Γ_d form a path of length 2 from Γ_c to Γ_d. We then use $\mathcal{M}(c,d)$ to denote the index, v, of the unique variable x_v between Γ_c and Γ_d on the path. We now denote the message passed from constraint Γ_c to variable $x_{\mathcal{M}(c,d)}$ by $\rho^*_{c\to d}$. Then PTP-message update rules for 3-COL problems can be completely described, as in the following lemma, by the update of $\rho^*_{c\to d}$ for every pair of constraint vertices Γ_c and Γ_d that are distance 2 apart.

Lemma 1: The support of $\rho^*_{c\to d}$ is $\{12, 13, 23, 123\}$, and when using $\{i, j, k\}$ to represent the three distinct element of $\{1, 2, 3\}$

$$\rho^*_{c\to d}(ij) = Z\Bigg(\prod_{b\in N(c)\setminus\{d\}} [\rho^*_{b\to c}(ik) + \rho^*_{b\to c}(jk) + \rho^*_{b\to c}(123)] - \prod_{b\in N(c)\setminus\{d\}} [\rho^*_{b\to c}(ik) + \rho^*_{b\to c}(123)] - \prod_{b\in N(c)\setminus\{d\}} [\rho^*_{b\to c}(jk) + \rho^*_{b\to c}(123)] + \prod_{b\in N(c)\setminus\{d\}} \rho^*_{b\to c}(123)\Bigg), \tag{8}$$

and

$$\rho^*_{c\to d}(123) = Z\Bigg(\prod_{b\in N(c)\setminus\{d\}} [\rho^*_{b\to c}(12) + \rho^*_{b\to c}(123)] + \prod_{b\in N(c)\setminus\{d\}} [\rho^*_{b\to c}(13) + \rho^*_{b\to c}(123)] + \prod_{b\in N(c)\setminus\{d\}} [\rho^*_{b\to c}(23) + \rho^*_{b\to c}(123)] - 2\times \prod_{b\in N(c)\setminus\{d\}} \rho^*_{b\to c}(123)\Bigg) \tag{9}$$

where $N(c) = \{b : c \cap b \neq \emptyset, b \in \Xi\}$, and Z is a normalization constant such that $\sum_t \rho^*_{c\to d}(t)$ is 1.

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

Theorem 1: The correspondence between update Eqs. (8) and (9) and Eqs. (2) and (3) is: $\rho^*_{c\to d}(ij) = \eta^k_{u\to v}$, and $\rho^*_{c\to d}(123) = \eta^*_{u\to v}$, where $v = \mathcal{M}(c,d)$ and u is the other neighbor of c besides $\mathcal{M}(c,d)$.

We note that the equivalence between SP and PTP on k-SAT problems can also be shown similarly.

4. Conclusion

In this paper, we present a clean and simple formulation of SP for arbitrary constraint-satisfaction problems in terms of "probabilistic token passing", where we stress the role of extending variable alphabets.

We note that not only unifying SP algorithms for previously studied problems, the PTP algorithm is in fact more general. Specifically note that in SP, alphabet χ_v is extended to $\chi_v \cup \{*\}$, but in PTP, χ_v is extended to its power set χ^*_v. In other words, instead of adding one "joker" symbol (*) to the original alphabet in SP, we add many "jokers" in PTP, where a "joker" is a non-singleton subset of the original alphabet. This makes PTP more general than SP especially for problems involving non-binary variables. As such, PTP should be regarded as a general principle for constructing SP algorithms.

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] A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, "Polynomial iterative algorithms for coloring and analyzing random graphs," Phys. Rev. E, vol.68, no.3, 036702, 2003.

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

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

[5] A. Brauntein, M. Mézard, M. Weight, and R. Zecchina, "Constraint satisfaction by survey propagation," in Computational Complexity and Statistical Physics, ed. A. Percus, G. Istrate, and C. Moore, pp.107–124, Oxford University Press, 2003.

[6] A. Braunstein and R. Zeccchina, "Survey propagation as local equilibrium equations," J. Stat. Mech., issue 06, p.06007, June 2004.

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

[8] R. Tu, Y. Mao, and J. Zhao, "On generalized survey propagation: Normal realization and sum-product interpretation," Proc. IEEE Int. Symp. Inform. Theory, pp.2042–2046, 2006.

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