Finite Energy Survey Propagation for Constraint Satisfaction Problems

Hai Leong Chieu
Singapore MIT Alliance
National University of Singapore
Email: haileong@nus.edu.sg

***Abstract*— The Survey Propagation (SP) algorithm [1] has recently been shown to work well in the hard region for random K-SAT problems. SP has its origins in sophisticated arguments in statistical physics, and can be derived from an approach known as the cavity method, when applied at what is called the one-step replica symmetry breaking level. In its most general form, SP can be applied to general constraint satisfaction problems, and can also be used in the unsatisfiable region, where the aim is to minimize the number of violated constraints. In this paper, we formulate the SP-Y algorithm for general constraint satisfaction problems, applicable for minimizing the number of violated constraints. This could be useful, for example, in solving approximate subgraph isomorphism problems. Preliminary results show that SP can solve a few instances of induced subgraph isomorphism for which belief propagation failed to converge.**

***Index Terms*— Survey Propagation, Constraint Satisfaction Problems, Subgraph Isomorphism**

I. Introduction

The satisfiability problem is the archetypical NP-complete problem, believed to be intractable in the worst case. In particular, the random 3-SAT problem, consisting of randomly generated clauses of length 3, has been the focus of much research. Numerical simulations have shown that satisfiability of 3-SAT instances depends critically on the clause to variable ratio, $\alpha = M/N$, where M is the number of clauses and N the number of variables. It was observed that there exists a phase transition for K-SAT problems at a value α_c, such that instances of K-SAT with $\alpha < \alpha_c$ are generally satisfiable, while instances with $\alpha > \alpha_c$ are generally unsatisfiable. Moreover, instances that has α close to α_c has been found to be hard to solve by local search methods such as WalkSAT. Recently, the Survey Propagation (SP) algorithm has been shown to work well in this hard region of 3-SAT problems [1], [2]. SP can be generalized to deal with the MAX-SAT problem [3], where the algorithm has been named **SP-Y**. SP has been formulated for general constraint satisfaction problems (CSP) [4], where the derivation is given as an extension of the more popular Belief Propagation (BP) algorithm [5].

In this paper, we formulate SP-Y for general constraint satisfaction problems starting from the cavity method. The cavity method is a mathematical method used to compute properties of ground states in many condensed matter and optimization problems [6]. Taken to different levels, it can be shown to be equivalent to either the BP or the SP algorithms. When using the cavity method to derive the SP algorithm, a parameter y arises that has to be optimized before the SP-Y algorithm can be used. It has been shown that in the satisfiable region for random K-SAT, y needs to be taken to infinity, and this leads to simplifications that have been exploited when the SP equations for 3-SAT [2] and general CSP [4] were formulated. For unsatisfiable instances, y is finite and can be empirically determined [3], leading to the SP-Y algorithm. We extend the work of [4] and [3] and derive the SP-Y algorithm for general CSP.

In 3-SAT, each variable can take either a true value or a false value. The SP algorithm for 3-SAT can be understood as extending the BP algorithm by adding a third state to each variable, known as a "joker" state [7], [2]. One problem that arises in generalizing SP to general CSP is that the number of "joker" states is exponential in the cardinality of the variable. Each variable can be in one of many "joker" states, where each "joker" state forbids a particular subset of values to the variable. We suggests methods to get around this intractability, and show preliminary results using the subgraph isomorphism problem as an example. In our experiments, SP is able to solve a few instances of the induced subgraph isomorphism problem for which BP failed to converge.

II. Cavity Method for CSP

The Survey Propagation (SP) algorithm can be derived through either the cavity approach or the replica approach [1]. The **cavity approach** and the **replica approach** are two ways of explaining a technique used in physics to solve spin glass models. The replica approach can be taken to different depths, from the replica symmetric ansatz (RS), to one-step replica symmetry breaking (1RSB), and finally to full replica symmetry breaking (fRSB) [8]. The authors of SP papers usually explain things through the cavity approach, which, taken to different depths, can be equivalent to the RS ansatz (which leads to BP), or to 1RSB (which leads to SP or SP-Y). In this paper, I will derive the SP-Y equations for general CSPs by following the cavity approach.

In general, a constraint satisfaction problem can be represented by a set of variables (denoted by $i, j, k...$) and a set of constraints over these variables (denoted by $a, b, c...$). For simplicity of notation, we assume that all variables have cardinality q. Like BP, the cavity method can be explained as a message passing algorithm on factor graphs [5]. In factor graphs, constraint satisfaction problems (CSP) are represented

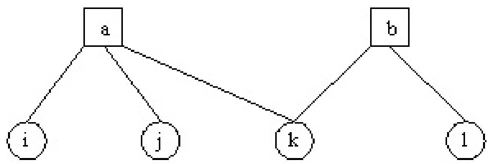

Fig. 1. Factor graph for the clause $(x_i \vee x_j \vee x_k) \wedge (\bar{x_k} \vee x_l)$, where $a = x_i \vee x_j \vee x_k$, and $b = \bar{x_k} \vee x_l$. Factor nodes correspond to constraints, and are represented as square nodes. Variable nodes are represented as circle nodes.

as a bipartite graph, consisting of factor nodes and variable nodes. Each factor node correspond to a constraint in the CSP, and is linked to all variable nodes that appear within that constraint (See Figure 1 for an example). Each factor a corresponds to a function $C_a(\mathbf{x_a})$, where $\mathbf{x_a}$ is the set of variables appearing in the constraint a. $C_a(\mathbf{x_a})$ is defined to equal zero if the constraint is satisfied by $\mathbf{x_a}$, and 1 otherwise. We consider the following joint probability function:

$$P(\mathbf{x}) = Z^{-1}\exp(-\beta H(\mathbf{x})), \tag{1}$$

$$H(\mathbf{x}) = \sum_a C_a(\mathbf{x_a}), \tag{2}$$

where Z is a normalization constant. In statistical physics, the term $\beta{=}1/T$, where T is the temperature. The study of spin glass models deals with variables with random couplings. The Hamiltonian $H(x)$ is the energy of a particular configuration, and physicists are interested in the value of the energy density of the global ground state (GGS), which is the configuration that minimizes the Hamiltonian. We denote by E_N the ground state energy of an N spin system, averaged over the distribution of random graphs and random couplings. It is postulated that the distribution of the ground state energy density E_N/N becomes more peaked as N increases, so that, as N tends to infinity, almost all samples have the same energy[1]:

$$U = \lim_{N\to\infty} \frac{E_N}{N}.$$

This has been proven to be true for random K-SAT [9].

The survey propagation (SP) algorithm arises from the application of the cavity method at zero temperature. While the cavity method involves averaging over disorder (i.e. over random graphs and random couplings), this can be done at the end, allowing the formulation of an algorithm that can be applied to specific instances of CSPs [6] .

At zero temperature, we are interested in finding a configuration that maximizes $P(\mathbf{x})$, or minimizes the Hamiltonian $H(\mathbf{x})$. If the CSP is satisfiable, this corresponds to finding a configuration (i.e. assignment of values to all variables) that gives rise to a zero Hamiltonian. In statistical physics, such configurations are known as a zero energy ground state. When the CSP is not satisfiable, we have finite energy ground states.

In the limit where N goes to infinity, we can define a "state" (at the zero temperature limit) to be *a cluster of configurations of equal energy, related to each other by single spin flip moves, which are "locally stable", in the sense that the energy cannot be decreased by any flip of a finite number of spins* [1]. For finite N, this definition is problematic (see Appendix C of [6] for a discussion), but we can understand states as local minima of the energy landscape. For large N, the optimization problem might present local minima (local ground states) that are strictly larger than the global minimum (global ground state). An optimization problem becomes difficult to solve by local search procedures when there are many local ground states (LGS), which act as traps for local search algorithms.

The cavity equations involve message passing between factor nodes and variable nodes. There are two kinds of messages: the $u_{a\to i}$ messages from factor a to variable i, and the $h_{i\to a}$, from variables i to factors a. Both $u_{a\to i}$ and $h_{i\to a}$ are vectors of length q (cardinality of variable i). We denote by v^σ the σ-th component of a vector v. At the 1RSB level, we are interested in the distribution over the messages, called surveys. This can be done efficiently at zero temperature, where we can assume that the components of the messages take integer values [1].

At zero temperature, the messages $u_{a\to i} \in \{0, -1\}^q$ can be understood as a kind of "warning" message: if $u^\sigma_{a\to i}$ equals -1, then factor a is sending a warning to variable i saying that if i takes the value σ, then the clause a will be violated. The messages $h_{j\to a}$ are also "warning" messages sent from variables to factors, telling the factor a that the variable j is only willing to take on values σ, for which $h^\sigma_{j\to a}$ equals zero. In statistical physics the h messages are called "local fields", and the u messages are called "local biases".

In the cavity approach, we remove a variable node (a site) from the graph, creating a cavity. The cavity equations are then derived by consistency considerations based on putting back this site, assuming a locally tree-like structure. For an in-depth discussion of the cavity method, please refer to [6]. In the rest of this section, I will state the assumptions behind the Replica Symmetric (RS) ansatz and one-step Replica Symmetry Breaking (1RSB).

A. Replica Symmetric Ansatz

The main assumption behind the RS ansatz is that the neighborhood of each node is locally tree-like, and that there exists only one state in the CSP. Following [10], we will also assume that the CSP is satisfiable, hence ignoring messages that lead to violated constraints[1]. This leads to the belief propagation algorithm [10].

First, we define the message passing mechanism: how the $u_{a\to i}$ messages and the $h_{j\to a}$ messages give rise to one another (See Figure 2). We denote $V(a)$ as the set of variables appearing in constraint a, and $V(i)$ the set of constraints containing variable i. The message that j sends to a, $h_{j\to a}$, is simply a sum of all incoming warnings from nodes $b \in V(j) - a$ (See Figure 2),

$$h_{j\to a} = \sum_{b\in V(j)-a} u_{b\to j}. \tag{3}$$

In order to define $u_{a\to i}$ in terms of incoming $\{h_{j\to a}\}_{j\in V(a)-i}$, we assume that we are adding a new site (the variable i) into the graph, and make use of consistency considerations to derive their relations. Denoting

[1]It is possible to define potentials so that BP minimizes the number of violated constraints. Here, however, we follow the formulation given in [10]

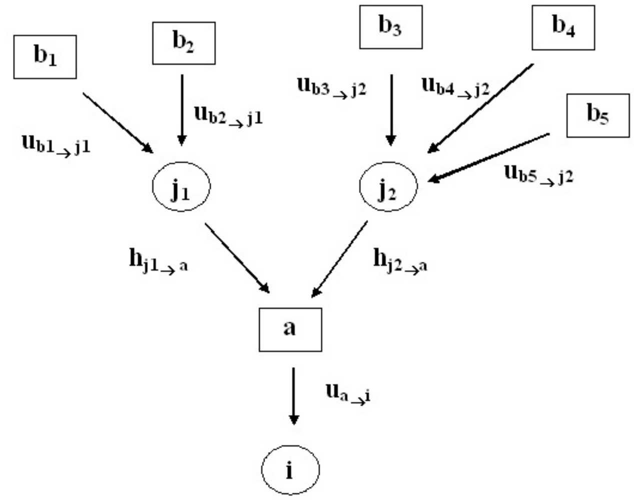

Fig. 2. Message passing between factor nodes and variable nodes.

by E^N (resp. E^{N+1}) the minimum energy before (resp. after) adding the cavity site i,

$$E^N(\{\sigma_j\}) = A - \sum_{\{a\in V(i)\}} \sum_{\{j\in V(a)-i\}} h^{\sigma_j}_{j\to a}, \tag{4}$$

$$E^{N+1}(\sigma_i,\{\sigma_j\}) = A + \sum_{\{a\in V(i)\}} [C_a(\sigma_i,\{\sigma_j\}) - \sum_{\{j\in V(a)-i\}} h^{\sigma_j}_{j\to a}], \tag{5}$$

where A is a constant. Figure 2 shows the messages involved when adding a site i that is connected to a single factor a. After adding the cavity site, E^{N+1} can also be written in terms of σ_i as follows

$$E^{N+1}(\sigma_i) = \min_{\{\sigma_j\}} E^{N+1}(\sigma_i,\{\sigma_j\}) = A - \sum_{a\in V(i)} [w_{a\to i} + u^{\sigma_i}_{a\to i}]. \tag{6}$$

In the above equation, $u_{a\to i}$ is a vector, while $w_{a\to i}$ is a scalar. Since the terms in the summation for each $a \in V(i)$ can be minimized independently (assuming a locally tree-like structure), we can write the definition of $u_{a\to i}$ and $w_{a\to i}$ as

$$-w_{a\to i} - u^{\sigma_i}_{a\to i} = \min_{\{\sigma_j\}}(C_a(\sigma_i,\{\sigma_j\}) - \sum_{\{j\in V(a)-i\}} h^{\sigma_j}_{j\to a}). \tag{7}$$

Note that the above definition does not uniquely define $u_{a\to i}$ and $w_{a\to i}$. In this paper, we will assume that $u_{a\to i}$ takes values in $\{0,-1\}$, and $w_{a\to i}$ absorbs the rest of the right-hand side of Equation 7. This is possible because the components of $u_{a\to i}$ differ by at most one.

Since we disallow violations of constraints, contradicting messages are forbidden. This means that in the right-hand side of Equation 7, the second term must attain the minimum value of zero. We define $T^0_{a\to i}(\{h_{j\to a}\})$ to be the set of allowed configurations of the variables $j \in V(a) - i$:

$$T^0_{a\to i}(\{h_{j\to a}\}) \quad = \quad \{\{\sigma_j\} | \forall j \in V(a) - i, h^{\sigma_j}_{j\to a} = 0\}. \tag{8}$$

From Equation 7, the outgoing message $u_{a\to i}$ can be calculated from the set of incoming messages $\{h_{j\to a}\}_{j\in V(a)-i}$ as follows:

$$U^{\sigma}_{a\to i}(S) \quad = \quad -\min_{\{s\}\in S} C_a(\{s\}|s_i \leftarrow \sigma), \tag{9}$$

$$u^{\sigma_i}_{a\to i} \quad = \quad U^{\sigma_i}_{a\to i}(T^0_{a\to i}(\{h_{j\to a}\})). \tag{10}$$

To simplify notation, we define the following sets of messages:

$$V_{j\to a}(h) = \{\{u_{b\to j}\}_{b\in V(j)-a} \,|\, \sum_{b\in V(j)-a} u_{b\to j} = h\}, \tag{11}$$

$$V^0_{a\to i}(u) = \{\{h_{j\to a}\}_{j\in V(a)-i} \,|\, U_{a\to i}(T^0_{a\to i}(\{h_{j\to a}\})) = u\} \tag{12}$$

$V^0_{j\to a}(h)$ is the set of incoming messages that could possibly give rise to the outgoing message h, and similarly for $V^0_{a\to i}(u)$.

With the message passing mechanism described above, we need to define the distributions over the messages. The probability of an outgoing message $u_{a\to i}$ is the sum of the probabilities over all incoming messages $h_{j\to a}$ that could possibly give rise to $u_{a\to i}$. Similarly for the probabilities of outgoing messages of type $h_{j\to a}$. Hence,

$$P_{j\to a}(h) \propto \sum_{V_{j\to a}(h)} \prod_{b\in V(j)-a} Q(u_{b\to j}), \tag{13}$$

$$Q_{a\to i}(u) \propto \sum_{V^0_{a\to i}(u)} |T^0_{a\to i}(\{h_{j\to a}\})| \prod_{j\in V(a)-i} P(h_{j\to a}), \tag{14}$$

where $|T^0_{a\to i}(\{h_{j\to a}\})|$ is the cardinality of the set $T^0_{a\to i}(\{h_{j\to a}\})$. Under the RS ansatz, we are working directly with distributions over configurations. Hence, when dealing with warning messages, we need to sum contributions of configurations that give rise to the same warning message. Although the above update equations are defined on warning messages, they are equivalent to the sum-product belief propagation defined on distributions over configurations [4].

When the above update equations converge, we are interested in the local fields (beliefs) at each variable

$$P_j(h) \quad \propto \quad \sum_{V_j(h)} \prod_{b\in V(j)} Q(u_{b\to j}), \tag{15}$$

where $V_j(h)$ is defined analogously to $V_{j\to a}(h)$ (Equation 11):

$$V_j(h) = \{\{u_{b\to j}\}_{b\in V(j)} \,|\, \sum_{b\in V(j)} u_{b\to j} = h\}. \tag{16}$$

B. One Step Replica Symmetry Breaking

Like RS, we assume a locally tree-like neighborhood for each site. Unlike RS, we assume that there are more than one state, and at each iteration, we have to take into account "level crossings", or interactions between different local state energies (LGS). Hence, under this framework, we allow violated constraints, and the aim is to minimize the number of violated constraints. This assumption has two consequences: (1) while under RS, we could simply keep track of distributions over configurations, under 1RSB, we have to keep track of surveys (distribution over sets of configurations). This is achieved by considering sets of configurations allowed by warning vectors. (2) To take into account crossings between LGS, we incorporate a penalty term to discourage moving into states with a higher energy. This penalty term has a natural interpretation in the derivation given below.

First, we define the message passing mechanism when violated constraints are allowed. From Equations 7 and 9,

$$T_{a\to i}(\{h_{j\to a}\}) \quad = \quad \arg\min_{\{\sigma_j\}}(-\sum_{\{j\in V(a)-i\}} h^{\sigma_j}_{j\to a}), \tag{17}$$

$$u^{\sigma_i}_{a\to i}(\{h_{j\to a}\}) \quad = \quad U^{\sigma_i}_{a\to i}(T_{a\to i}(\{h_{j\to a}\})). \tag{18}$$

And we define the counterpart for Equation 12

$$V_{a\to i}(u) = \{\{h_{j\to a}\}_{j\in V(a)-i} \mid U_{a\to i}(T_{a\to i}(\{h_{j\to a}\})) = u\} \quad (19)$$

In 1RSB, we make the following assumption: if we denote by $\eta_N(E)$ to be the number of states at an energy E, then

$$\eta_N(E) \approx \exp\left(N\Sigma(\epsilon)\right), \quad (20)$$

where $\epsilon = E/N$. $\Sigma(\epsilon)$ defined by the above equation is known as the *complexity*.

While adding a site i, (hence passing from N to $N+1$), denoting by $P(h_i, \delta E)$ as the joint probability of the local fields h_i and the change in energy, we have (expanding to first order)

$$\begin{aligned} \eta_{N+1}(E) &\propto \int P(h,\delta E) dh\, d(\delta E)\, \exp(N\Sigma(\frac{E-\delta E}{N})) \quad (21) \\ &\propto \int P(h,\delta E) dh\, d(\delta E)\, \exp(-y\delta E), \\ y &= \frac{d\Sigma(\epsilon)}{d\epsilon}. \quad (22) \end{aligned}$$

Hence, instead of Equation 13, we have [6]

$$\begin{aligned} P_{j\to a}(h) &\propto \int P(h,\delta E)\, d(\delta E)\, \exp(-y\delta E) \quad (23) \\ &\propto \sum_{V_{j\to a}(h)} \prod_{b\in V(j)-a} Q(u_{b\to j}) \exp(-y\delta E). \quad (24) \end{aligned}$$

Intuitively, the exponential term corresponds to a penalty to be paid for positive energy shifts. The relationship between ϵ and y can be defined using a Legendre transform of $\Sigma(\epsilon)$ given by $g(y) = y\Phi(y)$ as follows:

$$\begin{aligned} \Sigma(\epsilon) &= \min_y \epsilon y - y\Phi(y), \quad (25) \\ y\Phi(y) &= \min_\epsilon \epsilon y - \Sigma(\epsilon), \quad (26) \end{aligned}$$

and hence we rederive equation 22, as well as

$$\epsilon = \Phi(y) + y\frac{d\Phi(y)}{dy}. \quad (27)$$

Intuitively, denoting by E_N the global ground state and $U = \lim_{N\to+\infty} \frac{E_N}{N}$, we would expect $\Sigma(U) = 0$. By equation 25 and 27, this implies

$$\begin{aligned} \epsilon &= \Phi(y), \quad (28) \\ \frac{d\Phi}{dy} &= 0. \quad (29) \end{aligned}$$

Hence, to run SP, we have to determine the optimal value of y, given by $\frac{d\Phi}{dy} = 0$. An analytical expression of $\Phi(y)$ can be determined with Equation 28. This has been done in [1], [3] for K-SAT: in the SAT region, $\Phi(y)$ is monotonically increasing in the region $y \to \infty$, and hence y has to be taken to infinity. In the UNSAT region, $\Phi(y)$ has a finite optimum, which is dynamically determined in [3].

III. SP and SP-Y for general CSP

Since adding a cavity site means that new constraints could be violated, we can assume that δE is positive. In the SAT region for SAT problems, the optimal value of y is to take y to infinity [3]. From Equation 24, this means that terms with non-zero δE can be ignored. Hence, no additional constraints can be violated as a new site is added in the cavity method. This makes intuitive sense, since in the SAT region, no constraints should be violated in the global ground state. In the UNSAT region, on the other hand, there is a finite optimal value for y, and hence we have to take into account non zero values of δE. Since we are trying to minimize the number of violated constraints, we should allow violated constraints to come into the picture, although we would still like to penalize each violated constraint. This is achieved through the exponential penalty term in Equation 24.

We have not worked out the analytical form for $\Phi(y)$ to determine the behavior of y in the SAT and UNSAT region for general CSP, but by the above arguments, we develop SP for the SAT regions of CSPs by taking y to infinity. This gives rise to equations identical to those given in [4]. For the UNSAT region, we formulate the SP-Y equations with a finite y.

A. The SP-Y algorithm

We will first develop the more general SP-Y algorithm. In order to do that, the first step is to calculate the form of δE_i with Equations 4 and 6:

$$\begin{aligned} \delta E_i &= \min_{\sigma_i} E^{N+1}(\sigma_i) - \min_{\{\sigma_j\}} E^N(\{\sigma_j\}), \quad (30) \\ &= -\min_{\{\sigma_j\}}(-\sum_{\{a\in V(i)\}} \sum_{\{j\in V(a)-i\}} h^{\sigma_j}_{j\to a}) \\ &\quad - \sum_{a\in V(i)} w_{a\to i} + \min_{\sigma_i} \sum_{a\in V(i)} (-u^{\sigma_i}_{a\to i}). \quad (31) \end{aligned}$$

Now, imposing $u_{a\to i}$ to take values in $\{0, -1\}$ allows us to determine the value of $w_{a\to i}$ from Equation 7

$$w_{a\to i} = -\min_{\{\sigma_j\}}(-\sum_{\{j\in V(a)-i\}} h^{\sigma_j}_{j\to a}). \quad (32)$$

From Equations 32 and 31, we define δE_i and analogously, $\delta E_{i\to a}$,

$$\begin{aligned} \delta E_i &= \min_{\sigma_i} \sum_{a\in V(i)} (-u^{\sigma_i}_{a\to i}), \quad (33) \\ \delta E_{i\to a} &= \min_{\sigma_i} \sum_{b\in V(i)-a} (-u^{\sigma_i}_{b\to i}). \quad (34) \end{aligned}$$

There is a simple interpretation for δE for CSP: δE counts the number of violated constraints. In summary, the update equations for $P(h)$ and for $Q(u)$ are as follows:

$$\begin{aligned} P_{j\to a}(h) &\propto \sum_{V_{j\to a}(h)} \prod_{b\in V(j)-a} Q(u_{b\to j}) \exp(-y\delta E_{j\to a}), \quad (35) \\ Q_{a\to i}(u) &\propto \sum_{V_{a\to i}(u)} \prod_{j\in V(a)-i} P(h_{j\to a}). \quad (36) \end{aligned}$$

At convergence, we are interested in the local fields at each variable:

$$P_j(h) \propto \sum_{V_j(h)} \prod_{b\in V(j)} Q(u_{b\to j}) \exp(-y\delta E_j), \quad (37)$$

where $V_j(h)$ has been defined in Equation 16.

B. The SP algorithm

At infinite y, if constraints are violated, $\delta(E) \neq 0$ and $\exp(-y\delta(E))$ goes to zero: we work within the space of satisfied constraints. Messages of type h where all $h^\sigma \neq 0$ give rise to a positive δE and can be ignored. Hence the message passing mechanism here is similar to the message passing mechanism in the RS case.

We have the same update equations as SP-Y, without the exponential penalty:

$$P_{j\to a}(h) \propto \sum_{V_{j\to a}(h)} \prod_{b\in V(j)-a} Q(u_{b\to j}), \quad (38)$$

$$Q_{a\to i}(u) \propto \sum_{V^0_{a\to i}(u)} \prod_{j\in V(a)-i} P(h_{j\to a}), \quad (39)$$

$$P_j(h) \propto \sum_{V_j(h)} \prod_{b\in V(j)} Q(u_{b\to j}). \quad (40)$$

These equations are similar to those presented in [4], where they work directly with the infinite y assumption, and derived these equations by extending belief propagation to the 1RSB level. There are a couple of differences: the first difference is that in [4], the $u_{a\to i}$ take values in $\{0, +1\}$, whereas ours take values in $\{0, -1\}$. The second difference is that they used a max in Equation 3 in lieu of our sum: note that the definition of $T^0_{a\to i}(\{h_{j\to a}\})$ would be the same, for example, for $h = (-4, -2, 0), h = (-3, -3, 0)$, or $h = (-1, -1, 0)$. Hence, the h vectors can be grouped into equivalence classes. This is why the sum in Equation 3 can be replaced by a max: equivalent h are grouped together, so that h now takes values in $\{0, -1\}$.

C. Decimation

The outputs of SP or SP-Y are distributions over the local fields $P_j(h)$, for each variable j. To obtain a solution for a CSP from the outputs of SP or SP-Y, one typically has to perform a decimation process. In the decimation process, after each run of SP or SP-Y, (i) variables are selected to be set to certain values or (ii) certain values are forbidden to certain variables. Intuitively, if the local fields $P_j(h)$ reflects that the variable j is highly biased towards (resp. against) one value, then we can set it to that value (resp. forbid that value to the variable).

Different decimation methods have been used for different problems, such as 3-SAT [2], and 3-coloring [10]. In our initial experiments (Section VI), we have not implemented the decimation process. We only run SP once, and select a set of variables to set, based on the surveys for each variable.

IV. Deriving SP and SP-Y for 3-SAT

In this section, we derive the SP and SP-Y equations for 3-SAT presented in [2] and [3] respectively, using the formulation we have developed so far.

Define $e_+ = (-1, 0)$, $e_- = (0, -1)$ and $e_0 = (0, 0)$. For K-SAT, where variables are of cardinality $q = 2$,

$$u_{a\to i} \in \{e_-, e_0, e_+\} \quad (41)$$

$$h_{j\to a} = (m, n) \in \{0, -1, .., -\Gamma_j + 1\}^2, \quad (42)$$

where Γ_j is the number of neighboring factors to variable j. Moreover, each constraint a in 3-SAT can only forbid one value, so either $Q(u_{a\to i} = e_+)$ or $Q(u_{a\to i} = e_-)$ equals zero. This leads to an efficient parameterization for SP and SP-Y.

It is common to define the energy as follows (e.g. [3])

$$E = \sum_a \prod_{i=1}^{3} (1 + J_{a,i} s_i^a), \quad (43)$$

where s_i^a takes values in $\{+1, -1\}$, and $J_{a,i}$ equals -1 if the literal i is present in clause a, and $+1$ if literal i is negated in clause a. ($J_{a,i}$ is the "forbidden" value of s_i^a). With this formulation, instead of having u as vectors of size two, we define them as follows: $u = +1$ for e_+, $u = -1$ for e_-, and $u = 0$ for e_0.

A. Infinite y for KSAT

The cavity biases $u_{a\to i}$ take on values in $\{-1, 0, +1\}$, signifying a warning for variable i to take on the corresponding value (a value of zero means no warning). In the large y limit, conflicting warnings are disallowed. We work within the space of satisfiable constraints.

The exact value of $h_{j\to a}$ is unimportant in the large y limit. Since no constraints can be violated, for $h_{j\to a} = (m, n)$, either $m = 0$ or $n = 0$. To simplify things further, the $h_{j\to a}$ can be grouped into 3 classes. We note by $h = +1$ if $(m \neq 0, n = 0)$, by $h = 0$ if $(m = n = 0)$, and by $h = -1$ if $(m = 0, n \neq 0)$.

Following [2], we define $V_a^u(j)$ and $V_a^s(j)$ as neighbors which tend to make variable j satisfy or unsatisfy the constraint a, i.e.

$$V_+(j) = \{a | J_j^a = -1\}\ ;\ V_-(j) = \{a | J_j^a = +1\}, \quad (44)$$

$$\text{if } J_j^a = +1 : V_a^u(j) = V_+(j)\,;\, V_a^s(j) = V_-(j) - a, \quad (45)$$

$$\text{if } J_j^a = -1 : V_a^u(j) = V_-(j)\,;\, V_a^s(j) = V_+(j) - a. \quad (46)$$

To obtain the equations given in [2], we denote

$$Q(u = -J_{a,i}) = \eta_{a\to i}, \quad (47)$$

$$Q(u = 0) = 1 - \eta_{a\to i}, \quad (48)$$

$$Q(u = J_{a,i}) = 0, \quad (49)$$

$$P(sign(h) = J_{a,i}) = \Pi^u_{j\to a}, \quad (50)$$

$$P(sign(h) = -J_{a,i}) = \Pi^s_{j\to a}, \quad (51)$$

$$P(sign(h) = 0) = \Pi^0_{j\to a}. \quad (52)$$

Hence $\Pi^u_{j\to a}$, for example, is the joint probability of the event that at least one warning comes from $V_a^u(j)$, and no warnings come from $V_a^s(j)$. Hence, we get the SP equations presented in [2]:

$$\Pi^u_{j\to a} = [1 - \prod_{b\in V_a^u(j)} (1 - \eta_{b\to j})] \prod_{b\in V_a^s(j)} (1 - \eta_{b\to j}), \quad (53)$$

$$\Pi^s_{j\to a} = [1 - \prod_{b\in V_a^s(j)} (1 - \eta_{b\to j})] \prod_{b\in V_a^u(j)} (1 - \eta_{b\to j}), \quad (54)$$

$$\Pi^0_{j\to a} = \prod_{b\in V(j)} (1 - \eta_{b\to j}), \quad (55)$$

$$\eta_{a\to i} = \prod_{j\in V(a)-i} \frac{\Pi^u_{j\to a}}{\Pi^u_{j\to a} + \Pi^s_{j\to a} + \Pi^0_{j\to a}} \quad (56)$$

B. Finite y for KSAT

The finite y version of SP for K-SAT is formulated in [3]. It is stated in [3] that

$$\delta(E) = \sum_{a\in V(i)} |u_{a\to i}| - |\sum_{a\in V(i)} u_{a\to i}|. \quad (57)$$

This seems to count the number of violated constraints twice. However, since y is a parameter to be tuned, tuning y or $2y$ is equivalent. I would hence present the algorithm as it was presented in [3].

Following [3], we group $h_{j\to a} = (m,n)$ into equivalence classes, by denoting $h_{j\to a} = m - n$. This is because the actual values of m, n is unimportant. Only their relative values $(m-n)$ is important. We have, in the finite y case, $u_{a\to i} \in \{-1, 0, +1\}$, and $h_{j\to a} \in \{-\Gamma_j + 1, ..., -1, 0, 1, .., \Gamma_j - 1\}$, where Γ_j is the number of neighbors of j.

$$\begin{aligned} P_{j\to a}(h) \quad \propto \quad & \sum_{V_{j\to a}(h)} \prod_b Q_{b\to j}(u_{b\to j}) \\ & \exp(y(|\sum_j u_{a\to 0}| - \sum_j |u_{a\to 0}|)). \quad (58) \end{aligned}$$

The main update equations are as follows (Equations 13 to 16 in [3]):

$$\widetilde{P}^{(1)}_{j\to a}(h) = \eta^0_{b_1\to i}\delta(h) + \eta^+_{b_1\to i}\delta(h-1) + \eta^-_{b_1\to i}\delta(h+1), \quad (59)$$

$$\begin{aligned} \widetilde{P}^{(\gamma)}_{j\to a}(h) = {} & \eta^0_{b_\gamma\to i}\widetilde{P}^{(\gamma-1)}_{j\to a}(h) \\ & + \eta^+_{b_\gamma\to i}\widetilde{P}^{(\gamma-1)}_{j\to a}(h-1)\exp[-2y\theta(-h)] \\ & + \eta^-_{b_\gamma\to i}\widetilde{P}^{(\gamma-1)}_{j\to a}(h+1)\exp[-2y\theta(h)], \quad (60) \end{aligned}$$

where $\theta(h) = 1$ if $h \geq 0$, and zero otherwise. The above procedure calculates all the unnormalized $\widetilde{P}(h)$ at the same time, by multiplying on the fly, the penalty terms $\exp(-2y)$ for each violated constraint. The $\widetilde{P}_{j\to a}(h)$ are then normalized into $P_{j\to a}(h)$, and the second set of updates are as follows:

$$\eta^{J_{a,i}}_{a\to i} = \prod_{n=1}^{K-1} W^{J_{j_n},a}_{j_n\to a}, \quad \eta^{-J_{a,i}}_{a\to i} = 0, \quad \eta^0_{a\to i} = 1 - \eta^{J_{a,i}}_{a\to i}, \quad (61)$$

$$W^+_{j\to a} \quad = \quad \sum_{h=1}^{\Gamma_j - 1} P_{j\to a}(h), \quad (62)$$

$$W^-_{j\to a} \quad = \quad \sum_{h=-\Gamma_j+1}^{-1} P_{j\to a}(h). \quad (63)$$

We can understand the above procedure with the following correspondences:

$$\begin{aligned} \eta^\sigma_{a\to i} &= Q_{a\to i}(u = \sigma), \forall\sigma \in \{-1, 0, +1\} \quad (64) \\ W^\sigma_{j\to a} &= P_{j\to a}(sign(h) = \sigma), \forall\sigma \in \{+, -\}. \quad (65) \end{aligned}$$

V. Efficiency of SP and SP-Y

For general CSP, SP becomes quickly intractable for variables with large cardinality q. The $u_{a\to i}$ messages take values in $\{0, -1\}^q$, ($Q_{a\to i}(u)$ is a vector of size $O(2^q)$), while the $h_{j\to a}$ messages take values in $\{-\Gamma_j + 1, , ..., -1, 0\}$, where Γ_j equals the number of neighbors of the variable j ($P_{j\to a}(h)$ is a vector of size $O(\Gamma_j^q)$).

In practice, we could take advantage of special characteristics of problems. For example in K-SAT, the nature of the potentials result in the fact that the $h_{i\to a}$ can be grouped in three equivalence classes as $h > 0, h = 0$, or $h < 0$. Moreover, in K-SAT, the $u_{a\to i}$ warnings (-1,0) and (0,-1) are mutually exclusive (i.e. either $Q([0,-1])$ or $Q([-1,0])$ must equal zero). For graph coloring problems, due to the nature of the factors, the only possible values of $u_{a\to i}$ are all vectors of length q with at most a single -1 [10].

We will present preliminary results of the application of SP to the NP-complete problem of subgraph isomorphism. In this case, the natural formulation of the problem resulted in variables of large cardinalities. We get around this intractability by dynamically pruning warnings states of low probability.

VI. Preliminary Experimentation

In this section, we present preliminary results on the application of SP to the subgraph isomorphism problem. The SP-Y algorithm will be required for the approximate subgraph isomorphism problem, where the objective is to minimize the number of violated constraints. Due to time constraints, we have not done any experiments with SP-Y yet.

Subgraph Isomorphism is an important problem, both theoretically and in many practical applications such as object recognition [11], [12], scene analysis [13] and ontology alignment [14]. Subgraph Isomorphism (SGI) and Induced Subgraph Isomorphism (ISGI) are two different but related problems, defined as follows: The SGI problem consists of answering the following query: given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$, is G isomorphic to a subgraph of H? The ISGI problem, on the other hand, answers the following query: is G isomorphic to an induced subgraph of H? We generate random satisfiable instances of SGI problem as follows:

1) Generate H as a $G(nn, pp)$ graph: H has nn nodes, and each undirected edge is present with probability pp.
2) Randomly select mm nodes of H, and let G be the induced subgraph of H consisting of these mm nodes
3) Randomly remove each edge from H with probability qq. If any edges are removed, then G is a subgraph of H, but possibly not an induced subgraph.

To generate instances of ISGI, step (3) is omitted. In order to apply SP to this problem, we need to encode this problem as a factor graph. Let the nodes in G be numbered 1 to mm, and the nodes in H be numbered 1 to nn. Let $\{x_k\}_{k\in[1,mm]}$ be the variables, with cardinality nn. In this factor graph, $x_k = j$ means that node k in G maps to node j in H. Recall that a factor $C_a(\mathbf{x_a})$ equals 1 if the constraint is violated, and 0 otherwise.

For SGI, the constraints are the following

$$\begin{aligned} \forall(k1,k2) \in E_G, C_{(k1,k2)}(x_{k1}, x_{k2}) &= 0 \text{ if } (x_{k1}, x_{k2}) \in E_H \\ &= 1 \text{ otherwise} \\ \forall(k1,k2) \notin E_G, C_{(k1,k2)}(x_{k1}, x_{k2}) &= 1 \text{ if } x_{k1} = x_{k2} \\ &= 0 \text{ otherwise} \end{aligned}$$

TABLE I

NUMBER OF CONVERGENT RUNS OUT OF 100 RUNS OF INSTANCES OF SGI AND ISGI WITH BELIEF PROPAGATION.

nn	mm	pp	ISGI	SGI	
				qq=0.0	qq=0.1
40	20	0.2	74	0	0
40	20	0.3	99	0	0
40	20	0.4	100	0	0
40	20	0.5	100	0	0

TABLE II

NUMBER OF NON-CONVERGENT RUNS ON ISGI FOR $(nn = 10, mm = 5)$.

pp	BP	SP $numstate$			
		30	50	100	200
0.2	12	0	0	0	0
0.3	34	3	2	0	0
0.4	45	3	4	1	0
0.5	52	3	5	1	0

TABLE III

RUNS OF SP ON FAILED RUNS OF BP ON ISGI, WITH $numstate = 30$.

nn	mm	pp	BP	Div	Para	Mixed	Frozen	
							✓	✗
10	5	0.2	12	0	7	4	1	0
10	5	0.3	34	3	25	6	0	0
10	5	0.4	45	3	41	1	0	0
10	5	0.5	52	3	48	1	0	0
12	8	0.2	75	13	15	39	7	1
12	8	0.3	62	17	15	7	23	0
20	10	0.2	85	26	34	14	11	0
20	10	0.3	86	43	12	1	30	0

The first constraint imposes that edges in G must map to edges in H. The second constraint imposes that two nodes in G cannot map to the same node in H.

For ISGI, the constraints are the following

$$\begin{aligned}
\forall(k1,k2) \in E_G, C_{(k1,k2)}(x_{k1}, x_{k2}) &= 0 \text{ if } (x_{k1}, x_{k2}) \in E_H \\
&= 1 \text{ otherwise} \\
\forall(k1,k2) \notin E_G, C_{(k1,k2)}(x_{k1}, x_{k2}) &= 0 \text{ if } (x_{k1}, x_{k2}) \notin E_H \\
&= 1 \text{ otherwise}
\end{aligned}$$

The first constraint imposes that edges in G must map to edges in H. The second constraint imposes that nodes not connected in G must not be connected in H.

Kumar and Torr [12] used a variant of ISGI to match weighted graphs, and showed that both belief propagation and generalized belief propagation [15] worked reasonably well for ISGI. We show in Table I results of BP on 100 randomly generated instances of SGI and ISGI. (For BP, we used binary potential functions so that the joint probability is a uniform distribution over valid configurations). In the table, nn is the cardinality of $H = G(nn, pp)$, mm the cardinality of the subgraph G, and qq is the probability of removing an edge in the generation process described above. We see that SGI proves to be harder than ISGI for belief propagation.

Next, we study the performance of SP on the SGI and the ISGI problem. Even for small graph sizes, the number of warning states becomes large very quickly. To perform experiments efficiently, we initialize the local fields $h_{j\to a}$ with only "frozen states" (i.e. a single zero and all -1), and the universal "joker" state (i.e. all zeros). This means that we start from an initial solution where all other local fields have probability zero. After each update, we keep at most a fix number, $numstate$, of warning states in each message by pruning away warning states with the lowest probabilities. We run SP on small instances of both ISGI and SGI. SP has been successful for SAT problems in the critically constrained region. Instances of ISGI are more constrained than SGI, hence we expect SP to be more successful for ISGI than for SGI. Indeed, for SGI, in almost all our experiments, SP converges to a *paramagnetic* solution (local fields with probability 1 on the universal "joker" state), except a few rare cases where it fails to converge, possibly due to the pruning of warning states.

For ISGI, we first run 100 randomly generated instances with BP. We then run SP on instances for which BP fails to converge. When SP converges, we select, for each variable, the warning state with the highest probability. If all the warning states are the universal "joker" state, we say that SP converged to a paramagnetic solution. If there are "frozen states" among the warning states, then the result can be use for decimation. For each variable i in a "frozen state" e_{σ_i}, we set each v_i to the value σ_i, and then convert the ISGI into a SAT problem [16]. We solve this SAT problem by a SAT solver, and if the solution correctly finds a subgraph of H that is isomorph to G, then SP has correctly set the selected variables. Although SGI problems are known to be hard for SAT solvers [16], complete SAT solvers are able to solve these small instances readily.

The results for ISGI are shown in Table II and Table III. In Table II, we investigate the effects of pruning warning states. For matching graphs of size 5 to graphs of size 10, we have 5 variables of cardinality 10. This means that the messages $u_{a\to i}$ can take on $O(2^{10})$ possible distinct values. To investigate the effects of pruning, we perform experiments with values of $numstate$ ranging from 30 to 200. In Table II, the column **BP** shows the number of runs for which BP fails to converge, and the SP columns show the number of times SP fails to converge on runs where BP failed. We see that using small values of $numstate$ may cause divergence of the SP algorithm.

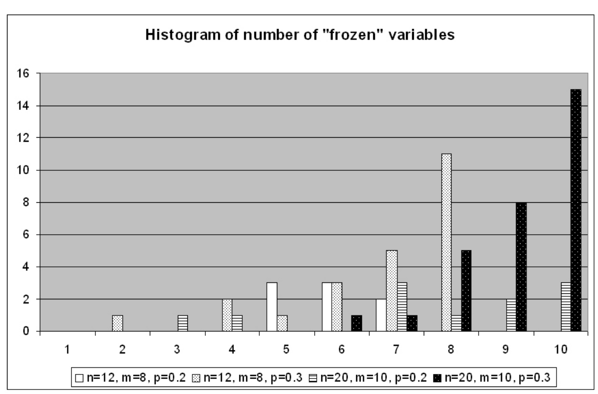

Fig. 3. Histogram of the number of variables set in the "frozen" runs.

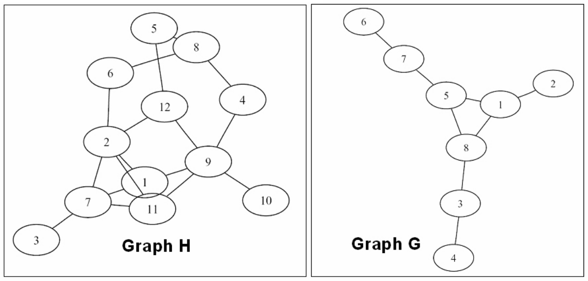

Fig. 4. An example of an instance of ISGI. SP correctly found the following mapping from graph G to graph H: (1,7), (2,3), (3,6), (4,8), (6,10), (7,9), and (8,2). The warning state for node 5 allows it to map to nodes 1 or 11 of graph H.

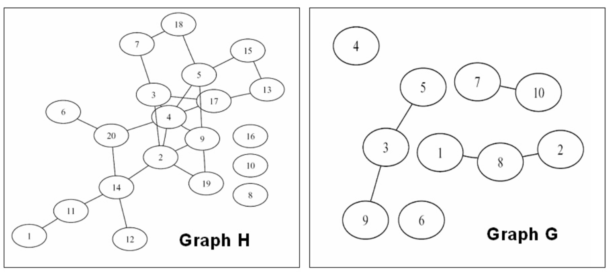

Fig. 5. An example of an instance of ISGI where SP returned a mixed joker solution. SP correctly found that most nodes in Graph G cannot map to the isolated nodes 8, 10 and 16 in graph H.

In Table III, we show the breakdown of results for SP runs on different graph sizes and edge densities, while keeping $numstate$ fixed at 30. The column **BP** is the number of runs for which BP fails to converge, the column **Div** the number of times for which SP fails to converge, the column **Para** the number of times convergent runs return a paramagnetic solution, and the column **Mixed** shows the number of times convergent runs return solutions with various kinds of "joker" states, but no "frozen" states. Under the column **Frozen**, we show the number of runs for which the frozen states correctly and wrongly set the variables. Among runs in the "Frozen" column, different numbers of variables are selected to be set to a particular value. We show the histogram of the number of "frozen" variables in Figure 3. For most successful runs, more than half of the variables are set correctly. For example, among the 23 successful instances for $(nn = 12, mm = 8, pp = 0.3)$, 11 of them correctly sets all 8 variables.

We show two examples in Figures 4 and 5. In Figure 4, we show one of the successful runs for $(mm = 12, nn = 8, pp = 0.3)$ where 7 of the 8 variables are set correctly. The last variable (variable 5) is assigned a "joker" state which allows it to take one of two possible values, 1 and 11. These are indeed the only two possibilities left for node 5. In Figure 5, we show a typical example of a "mixed" result for $(nn = 20, mm = 10, pp = .2)$. Isolated nodes in graph H has been found to be forbidden to nodes on connected components of graph G.

VII. Related Work

Survey Propagation has provided new insights into the difficulties of 3-SAT problem, which has been widely studied as the archetypical NP-complete problem. The 3-SAT problem becomes hard at the threshold of satisfiability, where the cluster of solutions in the easy phase breaks apart into many disconnected components [4]. The cavity method behind SP is useful in analyzing the phase transitions of random 3-SAT instances [4]. Since its formulation, SP has been applied successfully to instances of 3-SAT [2], Max-3-SAT [3], and graph 3-coloring problems [10]. Its wide application to general constraint satisfaction problem [4] is probably hindered by computational constraints due to the explosion of the number of "joker" states.

SP, with the large y limit, has been shown to be equivalent to belief propagation on a modified graph for 3-SAT [7], [2]. In the modified graph, each variable node can take a "joker" value, in addition to its true/false values. However, the structure of the factor graph has to be modified in order to take into account the semantics of the "joker" state, and this can be done in two different manners. Braunstein and Zecchina [7] defined a dual graph to the original graph, where BP on the dual graph is equivalent to SP on the original graph. Maneva, Mossel and Wainwright [17] defined the dual graph differently, introducing a parameter ρ that takes value in the $[0, 1]$ interval. They showed that BP on this dual graph corresponds to BP on the original graph when $\rho = 0$, and to SP when $\rho = 1$. This gives rise to a whole family of algorithms, and they argued (and showed experimentally) that the best ρ is near 1, but different from 1. SP has also been formulated at finite temperature [18]: at zero temperature, the u and h messages takes integer value while at finite temperature, surveys take values on a continuous interval. Wemmenhove and Kappen [18] showed preliminary results in the application of SP to a Sourlas code as a toy model.

Applications of SP and SP-Y to problems such as 3-SAT [2] and 3-coloring [10] rely on decimation processes that so far are task dependent. For SP and SP-Y to generally apply to CSPs, it would be interesting to develop a task-independent decimation process. Besides, when SP or SP-Y returns a paragmagnetic solution, the decimation processes in [10], [2] depends on walksat solvers to solve the reduced problem. For general CSP, this involves converting the CSP to a SAT problem. It would be interesting if BP [5] or other factor graph based inference process (e.g. [19], [20], [15]) can take over the problem when SP or SP-Y terminates in a paramagnetic solution.

We have used the NP-complete SGI and ISGI problems as examples for a general CSP for SP. Instances of SGI have been shown to be difficult for SAT solvers [16], and have been used as test cases in the SAT 2004 Competition. Theoretically, subgraph isomorphism is a common generalization of many important graph problems including Hamiltonian paths, cliques, matchings, girth, and shortest paths. A brute force search for solutions of SGI has been formulated as early as 1976 [21]. Applications of subgraph isomorphism can be found in a great variety of fields, ranging from computer vision [12], to applications in chemistry [21]. Kumar and Torr [12] has shown that generalized BP [15] and BP worked reasonably well for matching weighted graphs. Their definition of potentials aims to find approximate *induced* subgraphs, which we have observed to be an easier problem for belief propagation. Encoding ISGI and SGI into a factor graph resulted in variables with large cardinalities, making SP intractable due to the explosion of the number of "joker" states. Preliminary experiments have shown that pruning warning states allows SP to be runnable on instances of ISGI of small sizes. We intend to extend our work to SP-Y, with which we can use to solve the approximate ISGI and SGI problem by minimizing the number of violated constraints.

VIII. Conclusion

We have formulated SP and SP-Y for general CSP, and illustrated its application to instances of the subgraph isomorphism problem. In our experimental results, we have shown that SP can solve instances of ISGI for which BP failed to converge. This paper presents a preliminary study of the application of SP to general CSP, and much remains to be done. We hope to use SP to solve larger instances of the SGI and ISGI problem, and this involves formulating a good decimation process in order to reduce the size of the instances, and deciding what to do when SP returns a paramagnetic solution. We have also formulated SP-Y for general CSP, which should apply to the approximate SGI problem. SP-Y, is however, more expensive than SP, and further research is required to come up with an efficient version of SP-Y that can run on large instances of the SGI problem.

Acknowledgment

The author would like to thank Yee Whye Teh and Wee Sun Lee for suggestions and ideas in this paper.

References

[1] M. Mezard and R. Zecchina, "The random k-satisfiability problem: from an analytic solution to an efficient algorithm," *Physical Review E*, vol. 66, p. 056126, 2002. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0207194

[2] A. Braunstein, M. Mezard, and R. Zecchina, "Survey propagation: An algorithm for satisfiability," *Random Struct. Algorithms*, vol. 27, no. 2, pp. 201–226, 2005.

[3] D. Battaglia, M. Kolar, and R. Zecchina, "Minimizing energy below the glass thresholds," *Physical Review E*, vol. 70, p. 036107, 2004. [Online]. Available: doi:10.1103/PhysRevE.70.036107

[4] A. Braunstein, M. Mezard, M. Weigt, and R. Zecchina, "Constraint satisfaction by survey propagation," 2002. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0212451

[5] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, 2001.

[6] M. Mezard and G. Parisi, "The cavity method at zero temperature," *J.STAT.PHYS*, vol. 111, p. 1, 2003. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0207121

[7] A. Braunstein and R. Zecchina, "Survey propagation as local equilibrium equations," p. 06007, 2004. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0312483

[8] V. Dotsenko, *Introduction to the Replica Theory of Disordered Statistical Systems*. Prentice-Hall, 2001.

[9] A. Z. Broder, A. M. Frieze, and E. Upfal, "On the satisfiability and maximum satisfiability of random 3-cnf formulas," in *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1993, pp. 322–330.

[10] 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, p. 036702, 2003. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0304558

[11] A. R. Steven Gold, "A graduated assignment algorithm for graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377–388, 1996.

[12] M. P. Kumar and P. Torr, "Fast memory-efficient generalized belief propagation," in *Proceedings of ECCV*, 2006.

[13] R. Baeza-Yates and G. Valiente, "An image similarity measure based on graph matching," *spire*, vol. 00, p. 28, 2000.

[14] W. Hu, N. Jian, Y. Qu, and Y. Wang, "Gmo: A graph matching for ontologies," in *Proceedings of theI ntegrating Ontologies Workshop*, 2005.

[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, no. 7, 2005.

[16] C. Anton, "Structure and randomness of satisfiability," Ph.D. dissertation, University of Alberta, 2005.

[17] E. N. Maneva, E. Mossel, and M. J. Wainwright, "A new look at survey propagation and its generalizations," 2004. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0409012

[18] B. Wemmenhove and H. J. Kappen, "Survey propagation at finite temperature: application to a sourlas code as a toy model," 2005. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0508586

[19] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1120–1146, 2003.

[20] T. P. Minka and Y. A. Qi, "Tree-structured approximations by expectation propagation," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[21] J. R. Ullmann, "An algorithm for subgraph isomorphism." *J. ACM*, vol. 23, no. 1, pp. 31–42, 1976.