Postprint of Convergence Analysis of Warning Propagation Algorithm Based on Structural Entropy

Authors: Niu Jin, Wang Xiaofeng, Lin Qingwen

Date: 2020-09-28T00:00:00+00:00

Abstract

Convergence is a critical metric for evaluating the performance of message passing algorithms. When solving satisfiability problems, the structural characteristics of propositional formulas influence algorithmic convergence. Message passing algorithms do not always converge for formulas with complex structures. To systematically provide a theoretical explanation for this phenomenon, we employ structural entropy methods and techniques to propose a structural entropy model for propositional formulas along with its measurement methodology, and compute the structural entropy of random satisfiability instances. The Warning Propagation (WP) algorithm serves as a fundamental model of message passing algorithms; analyzing its convergence is of significant importance for studying the convergence of other such algorithms. We analyze the relationship between WP algorithm convergence and structural entropy, and provide decision conditions for WP algorithm convergence. Experimental analysis demonstrates that the proposed method is effective and feasible.

Full Text

Preamble

Vol. 38 No. 3
Application Research of Computers (ChinaXiv Cooperative Journal)

Convergence Analysis of Warning Propagation Algorithm Based on Structural Entropy

Niu Jin, Wang Xiaofeng, Lin Qingwen
School of Computer Science & Engineering, North Minzu University, Yinchuan 750021, China

Abstract: Convergence is a critical metric for evaluating the performance of information propagation algorithms. When solving satisfiability problems, the structural characteristics of propositional formulas significantly influence algorithmic convergence. For formulas with complex structures, information propagation algorithms do not always converge, yet systematic theoretical explanations for this phenomenon remain scarce. Leveraging methods and techniques from structural entropy theory, this paper proposes a structural entropy model for propositional formulas along with its measurement methodology, enabling the computation of structural entropy for random satisfiability instances. As a fundamental model of information propagation algorithms, the Warning Propagation (WP) algorithm serves as an important reference for studying the convergence of other such algorithms. We analyze the relationship between WP algorithm convergence and structural entropy, presenting convergence determination conditions for the WP algorithm. Experimental analysis demonstrates the effectiveness and feasibility of our approach.

Keywords: satisfiability problem; propositional formula; structural entropy; warning propagation algorithm; convergence

0 Introduction

Constraint satisfaction problems are ubiquitous in the real world and have garnered widespread attention in artificial intelligence research [?, ?, ?]. As a major subclass, the Boolean satisfiability problem (SAT) was the first problem proven to be NP-complete [?], meaning no algorithm can find exact solutions in polynomial time. Nevertheless, many real-world problems can be reduced to SAT problems [?, ?, ?], making SAT a focal point of ongoing research in computer science and mathematical logic. The problem seeks to determine whether there exists an assignment of Boolean variables to a given propositional formula that satisfies it (i.e., makes the formula true). Since any propositional logic formula can be converted to an equivalent conjunctive normal form (CNF) [?], this paper focuses exclusively on CNF formulas.

In the 1980s, physicists proposed an information propagation algorithm based on factor graphs. This algorithm' s performance heavily depends on the factor graph structure: when the factor graph is singly connected (with only one path between any two nodes), the algorithm generally converges correctly. While information propagation algorithms often exhibit good performance when solving SAT problems, convergence becomes problematic as factor graph structures grow complex, particularly when cycles emerge. Information may circulate infinitely within these cycles, preventing it from converging to a fixed point and causing algorithmic failure—a phenomenon known as non-convergence [?]. The Warning Propagation (WP) algorithm, as the most fundamental information propagation algorithm, provides important insights for studying other such algorithms. WP is a message-passing iterative algorithm operating on factor graphs that transmits information (called warning messages) along edges. These warnings indicate whether a variable' s assignment can fully determine clause satisfiability. The algorithm employs an iterative strategy to update warning messages until convergence or until reaching a maximum iteration count, at which point iteration is forcibly terminated. Consequently, convergence represents the core property of the WP algorithm.

Previous research has yielded several results on WP algorithm convergence. Reference [?] proved that the algorithm converges effectively when only one cycle exists in the factor graph, yielding valid approximations of variable marginal distributions. Reference [?] provided necessary and sufficient conditions for convergence based on a Gaussian model, though limited to cases where two variables cannot belong to the same clause. Reference [?] analyzed convergence using message matrices, presenting another necessary and sufficient condition restricted to positive semidefinite message matrices. References [?, ?] examined convergence in Gaussian models but required semidefinite programming for verification, which is inefficient and computationally difficult. Reference [?] analyzed convergence for mean-solving algorithms, requiring estimation of spectral radii for finite-scale matrices, which lacks practical feasibility. Reference [?] analyzed algorithmic convergence and provided a probabilistic condition for information propagation algorithm convergence, but this was primarily limited to planted-assignment random satisfiability instance generation models requiring sufficiently large probability p, essentially ensuring at most one assignment satisfies the instance with high probability.

Evidently, existing convergence analyses of warning propagation algorithms are based on specific models, suffer from narrow applicability, or involve complex verification conditions. Therefore, convergence analysis remains incomplete, particularly for general factor graph structures beyond simple tree models.

Since WP algorithm convergence heavily depends on factor graph structure, and as factor graph scale increases, the message-passing process among nodes becomes extremely complex and exhibits random uncertainty. Shannon [?] proposed information entropy as a measure of uncertainty—a static metric based on probability distributions. In information theory, when nodes are selected according to a probability distribution, entropy reflects the average information required to identify a node' s code. Thus, in communication, information entropy only measures point-to-point information. Building upon this, reference [?] first introduced structural entropy. Unlike information entropy, structural entropy is a dynamic metric that captures network complexity. Structural entropy compensates for information entropy' s inability to reflect network structural complexity precisely, enabling measurement of node interaction complexity and internal structural complexity. Introducing structural entropy allows us to measure the complexity and uncertainty of WP algorithms on factor graphs while reflecting the dynamic complexity of random walk processes within WP algorithms.

1 Basic Knowledge

1.1 Factor Graph

For a CNF formula $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$, let the variable set be $X = \{x_1, x_2, \dots, x_n\}$. The formula can be represented as a bipartite graph $G = (V, E)$ called a factor graph, where $V = X \cup C$ is the vertex set and $C = \{C_1, C_2, \dots, C_m\}$ is the clause set. The factor graph contains two types of edges: $(C_i, x_j) \in E$ if clause C_i contains positive literal x_j, and $(C_i, x_j) \in E$ if clause C_i contains negative literal $\neg x_j$, as shown in Figure 1 [Figure 1: see original paper].

1.2 WP Algorithm

On each edge of the factor graph, the WP algorithm defines a message (called a warning) as a Boolean value transmitted from clause node a to variable node i. The warning $u_{a\to i}$ indicates whether variable x_i's assignment can fully determine clause a' s satisfiability: $u_{a\to i} = 1$ means x_i' s assignment can determine clause a' s satisfiability, while $u_{a\to i} = 0$ means it cannot (i.e., other variables' assignments can also satisfy clause a). The WP message update iteration equation is:

$$u_{a\to i}^{(t+1)} = \prod_{j\in V(a)\setminus i} \theta\left(\sum_{b\in V(j)\setminus a} u_{b\to j}^{(t)}\right)$$

where $V(a)$ is the cavity domain of clause a. If variable x_i appears only in clause a, then $V(a) \setminus i = \emptyset$.

This can be rewritten as:

$$u_{a\to i}^{(t+1)} = \prod_{j\in V(a)\setminus i} \left(1 - \prod_{b\in V(j)\setminus a} (1 - u_{b\to j}^{(t)})\right)$$

The WP algorithm for solving CNF formula F is as follows:

Algorithm 1: Warning Propagation Algorithm

Input: CNF formula F, factor graph $G = (V, E)$, maximum iteration steps $t_{\max}$
Output: If WP does not converge after $t_{\max}$ steps, output UN-CONVERGED; otherwise, output all warning messages

1. At initial time $t = 0$, for each edge in factor graph G, randomly assign a value from $\{0, 1\}$ with probability $1/2$ to $u_{a\to i}^{(0)}$.
2. For $t = 1$ to $t_{\max}$:
   a. Update $u_{a\to i}^{(t)}$ for each edge in G in some order (e.g., lexicographic) using Equation (1).
   b. If $u_{a\to i}^{(t)} = u_{a\to i}^{(t-1)}$ for all edges, output all warning messages and terminate.

3. Output UN-CONVERGED.

1.3 Structural Entropy

Structural entropy can measure the dynamic complexity of WP algorithms on factor graphs. Message propagation on factor graphs is complex and random. Since clause nodes and variable nodes connect directly via edges, while clause nodes (and variable nodes) can only connect indirectly through other edges, this special relationship creates a multi-level structure in factor graphs. Therefore, we introduce structural entropy to measure such multi-level factor graph structures.

For a given undirected connected graph $G = (V, E)$, let $\mathcal{T}$ be a partition tree of G. The structural entropy is defined as:

$$\mathcal{H}^T(G) = -\sum_{\alpha\in\mathcal{T},\alpha\neq\lambda} \frac{g_\alpha}{\mathrm{Vol}(G)} \log_2 \frac{\mathrm{Vol}(\alpha)}{\mathrm{Vol}(\alpha^-)}$$

where g_α is the number of edges connecting nodes inside set α to nodes outside α, $\mathrm{Vol}(\alpha)$ is the volume of set α (sum of degrees of all nodes in α), and α^- is the parent node of α in the partition tree $\mathcal{T}$. For a weighted graph, g_α represents the sum of weights of all edges between nodes inside α and nodes outside α, and $\mathrm{Vol}(\alpha)$ is the sum of weights of all edges incident to nodes in α.

The K-dimensional structural entropy of a weighted undirected connected graph G is:

$$\mathcal{H}^K(G) = \min_{\mathcal{T}}\{\mathcal{H}^T(G) \mid \mathcal{T} \text{ is a partition tree of } G \text{ with height } K\}$$

For a weighted undirected disconnected graph G, the K-dimensional structural entropy is:

$$\mathcal{H}^K(G) = \sum_{j=1}^{L} \frac{\mathrm{Vol}(G_j)}{\mathrm{Vol}(G)} \cdot \mathcal{H}^K(G_j)$$

where $G_1, G_2, \dots, G_L$ are all connected components of G.

---

2 Structural Entropy of Propositional Formulas

Based on WP algorithm convergence analysis, when factor graphs contain multiple cycles rather than tree structures, convergence cannot be guaranteed. Therefore, we employ high-dimensional structural entropy to measure factor graph structural complexity, thereby analyzing WP algorithm convergence on general factor graphs containing multiple cycles.

2.1 Undirected Graph Conversion Technique

To compute the structural entropy of propositional formulas, we must convert the factor graph model to a weighted undirected graph model. Factor graphs use two node types to represent clauses and variables in CNF formulas, with dashed and solid edges expressing node relationships. This structure most intuitively captures the mapping between factor graphs and formulas and the WP algorithm' s computational process. We construct an adjacency matrix that maps both node types in the factor graph to a single node type in the undirected graph, using weights to represent node relationships.

Let variable nodes be a and b, and clause nodes be i and j. Based on Equation (1), when $\sum_{j\in V(a)\setminus i} J_{j\to a} = 1$ and $\sum_{j\in V(b)\setminus i} J_{j\to b} = 1$, we have $J_{a\to i} = 0$ and $J_{b\to i} = 0$. Two cases arise: (1) $J_{a\to i} = 1$ and $J_{b\to i} = -1$, or (2) $J_{a\to i} = -1$ and $J_{b\to i} = 1$. These cases indicate that variable j appears with opposite polarities in clauses a and b. Due to clause distinctness, in case (1) we have $J_{a\to i} = 1$ and $J_{b\to i} = -1$, while in case (2) we have $J_{a\to i} = -1$ and $J_{b\to i} = 1$ or $J_{a\to i} = 1$ and $J_{b\to i} = -1$.

When $J_{a\to i} = 1$ and $J_{b\to i} = -1$, the ratio of positive to negative literals in cycles is 0.5. If $J_{a\to i} = -1$ and $J_{b\to i} = 1$, similar analysis shows two groups of scenarios, each with ratios of 0.25, 0.25, and 0.5. When $J_{a\to i} = 1$ and $J_{b\to i} = -1$, two groups of scenarios exist with ratios of 0.25 and 0.25 each. When $J_{a\to i} = -1$ and $J_{b\to i} = 1$, since results are only affected by $J_{a\to i}$, this case encompasses all possibilities without affecting final outcomes, so we omit detailed analysis.

Clearly, in basic cycle models, the expected ratio of positive to negative literals is 0.5. We set this expectation as the weight for negative or positive literals. Since structural entropy only concerns factor graph structural information, whether dashed edges have weight 0.3 and solid edges 0.7 or the reverse assignment, both produce symmetric structures that do not affect structural information. For consistency, we assign weight 0.3 to dashed edges and 0.7 to solid edges. Using this method, the factor graph in Figure 1 [Figure 1: see original paper] converts to the weighted undirected graph shown in Figure 2 [Figure 2: see original paper].

2.2 Structural Entropy Model and Measurement Method

Given a weighted undirected connected graph $G = (V, E)$ with n nodes and m edges, we compute structural entropy by constructing a partition tree using community detection. The tree structure is built according to the following specifications, where $\mathcal{T}$ is a partition of node set V and λ denotes the root node:

a) Represent the root node as λ, defining the node set V. The direct successor nodes of λ are $X_1, X_2, \ldots, X_L$, with nodes sorted left-to-right in increasing order of λ.

b) For each node $\alpha \in \mathcal{T}$, $\alpha \subseteq V$ and $\alpha \neq \lambda$. When nodes α and β satisfy $\alpha \subset \beta$, we denote α as a left node of β.

c) For nodes α and β, let $\alpha \cap \beta$ represent the longest common prefix. When $\alpha \cap \beta = \emptyset$, α and β represent a partition of V.

d) For each $\alpha \in \mathcal{T}$, $h(\alpha)$ denotes the height of α (root node height is 0, and for each node α, $h(\alpha) = h(\alpha^{-}) + 1$).

e) For each $\alpha \in \mathcal{T}$, when $\alpha \neq \lambda$, α is the union of its children, thus $\alpha = \bigcup_{\beta^{-}=\alpha} \beta$.

f) For each leaf node $\alpha \in \mathcal{T}$, α is a singleton node containing a single node from V.

After initializing the partition tree, we compute structural entropy by performing minimum-cut segmentation on the weighted undirected graph to obtain two non-overlapping communities M_1 and M_2. These communities become the second-level tree nodes, with the root node λ as their parent (containing all nodes in the factor graph). If either community contains more than one node, we continue applying minimum-cut segmentation to M_1 or M_2, generating sub-communities M_{11}, M_{12} and M_{21}, M_{22} as third-level tree nodes. Here, M_{11} and M_{12} are direct successors of M_1, while M_{21} and M_{22} are direct successors of M_2, with $M_{11} \cup M_{12} = M_1$ and $M_{21} \cup M_{22} = M_2$. If sub-communities still contain multiple nodes, we repeat the segmentation process and construct the partition tree until each community contains only a single node.

For disconnected weighted undirected factor graphs, we traverse nodes to identify each connected component and construct partition trees for each component. The minimum-cut process for factor graphs and corresponding partition tree construction are illustrated in Figures 3 [Figure 3: see original paper] and 4 [Figure 4: see original paper] (using dashed and solid lines to represent weights for clarity).

After constructing the partition tree, we compute each node' s entropy using Equation (4), where node degrees correspond to degrees in the weighted undirected graph. Summing all node entropies yields $\mathcal{H}^T(G)$. Equation (5) shows that structural entropy seeks the minimum value, i.e., an optimal partition. Current academic research on community detection continues, and since minimum-cut algorithms produce communities with low inter-community edge density and high intra-community edge density, they are widely applied [**?**]. Using the expected ratio of positive to negative literals as edge weights, the minimum-cut based on these weights satisfies the community definition in structural entropy while minimizing information loss during structural abstraction. Thus, the result $\mathcal{H}^K(G)$ represents the structural information of the CNF formula.

Algorithm 2: CNF Formula Structural Information Acquisition

Input: CNF formula F
Output: Structural entropy $\mathcal{H}^K(G)$

1. Construct factor graph $G = (V, E)$ from $\{C_1, C_2, \dots, C_m\}$ and $\{x_1, x_2, \dots, x_n\}$.
2. Assign solid line weight $FL = 0.7$ and dashed line weight $DL = 0.3$ to construct weighted undirected graph G'.
3. Initialize partition tree $\mathcal{T}$ with $\mathcal{T} = \emptyset$.
4. If G' is connected: proceed to step 5 with flag $= 1$; else: traverse all connected components and proceed to step 5 with flag $= 0$.
5. Segment graph using minimum-cut.
6. If subgraph contains a single node: proceed to step 7; else: repeat step 5.
7. Construct partition tree from minimum-cut results.
8. If flag $= 1$: compute $\mathcal{H}^K(G)$ using Equation (5); else: compute $\mathcal{H}^K(G) = \sum_{j=1}^{L} \frac{\mathrm{Vol}(G_j)}{\mathrm{Vol}(G)} \cdot \mathcal{H}^K(G_j)$.
9. Return $\mathcal{H}^K(G)$.

As shown in Equation (4), structural entropy based on partition trees sums the entropies of all tree nodes, effectively aggregating structural entropies of different communities. Our factor graph structural entropy accounts for the dynamic, random information propagation process of WP algorithms on factor graphs. Based on the resulting community structure, Equation (5) effectively measures dynamic information interactions both between and within communities. The recursive community segmentation process considers virtually all possible community partitions, fundamentally analyzing factor graph structural complexity. Structural entropy thus provides a comprehensive analysis of dynamic complexity and uncertainty for such graph structures. Using the expected ratio of positive to negative literals as weights for minimum-cut effectively removes noise while minimizing information loss during abstraction to structural entropy. The algorithm computes the CNF formula' s structural entropy, which encapsulates structural complexity, uncertainty, and interaction properties, representing a high-level abstraction of the formula' s structural information.

Applying this algorithm to WP algorithm convergence analysis offers broad applicability without reliance on specific graph structures, provides simple verification conditions, fills gaps in previous WP convergence analyses, and offers systematic theoretical support for studying WP algorithm convergence on complex factor graphs, opening new research directions. We next present experimental validation of WP algorithm convergence based on structural entropy.

---

3 Convergence Analysis of WP Algorithm

We analyze the relationship between structural entropy and WP algorithm convergence experimentally. Taking the case where $n = 10$ as an example, the experimental procedure is: let $\alpha = m/n$, where the clause-to-variable ratio significantly impacts factor graph structure and WP algorithm convergence. We generate random instances using the model $G(n, m, 3)$, where n represents the number of variables, m the number of clauses, and each clause contains exactly

3 variables. Two datasets are selected with variable counts $n = 10$ and $n = 20$, with WP algorithm maximum iterations set to 100. By fixing n and controlling α through the model, we generate random instances of different scales. For accuracy, we generate 100 instances for each scale and segment each instance' s factor graph as described above, finally computing the propositional formula' s structural entropy using Algorithm 2. We then calculate the mean structural entropy across 100 instances as the group structural entropy.

As α increases from 0 to 5, the structural entropy variation for randomly generated 3-SAT instances is shown in Figures 5 [Figure 5: see original paper] and 6 [Figure 6: see original paper], where each data point represents the mean structural entropy across 100 random instances. Simultaneously, for each instance group, we count the number of instances where WP algorithm converges successfully to obtain the group' s convergence probability. The relationship between convergence probability and structural entropy is shown in Figures 7 [Figure 7: see original paper] and 8 [Figure 8: see original paper].

Experimental results show that as α increases, random instance structural entropy increases while the increment decreases. Figure 5 indicates that when $n = 10$, structural entropy grows rapidly, with moderate growth when $\alpha < 2$, and the increment approaching 0 as α nears 5. Figure 7 shows that when $n = 10$, as α approaches 5, instance entropy values densely distribute between 3.5324 and 3.8941, with 74% of instances falling in this range, converging toward 3.9. WP algorithm converges completely on all instances when entropy is below 3.7148, with convergence probability decreasing from 1 to 0.71 when entropy exceeds 3.7148.

When $n = 20$, instance entropy values densely distribute between 3.8136 and 4.3494, converging toward 4.35. WP algorithm converges almost completely when entropy is below 4.2795, with convergence probability dropping sharply from 1 to 0.66 when entropy exceeds 4.2795.

These structural entropy variations indicate that with fixed variable scale, factor graph dynamic complexity rises rapidly within a certain range as α increases, but the growth rate significantly slows beyond a threshold. In simple factor graph structures, any change—such as increased cycles, minor scale variations, or diameter changes—substantially impacts structure. However, when factor graph scale reaches a certain level, additional cycles and moderate scale increases have diminished impact on dynamic complexity, indicating more stable graph states. Convergence behavior shows that WP algorithm convergence probability decreases as factor graph dynamic complexity increases when the graph state is stable. However, when dynamic complexity remains below the threshold, WP algorithm converges with high probability.

Factor graph structure determines WP performance. The experiments provide sufficient conditions for WP algorithm convergence at different scales: when $n = 10$ and $\mathcal{H}^K(G) < 3.7148$, WP algorithm converges with high probability; when $n = 20$ and $\mathcal{H}^K(G) < 4.2795$, WP algorithm converges with high probability.

Convergence conditions for other factor graph scales can be obtained similarly.

---

4 Conclusion

This paper proposes a CNF formula structural information acquisition algorithm expressed through structural entropy. Experiments reveal structural entropy variations across different factor graph scales and WP algorithm convergence properties within different entropy ranges, providing sufficient convergence conditions for various scales. Future work will seek critical values or control parameters for propositional formula structural entropy to measure convergence of other information propagation algorithms, with further theoretical analysis and experimental validation of the relationship between structural entropy and factor graph structural information.

References

[1] Martin E. Dyer, Alan M. Frieze, Michael Molloy. A probabilistic analysis of randomly generated binary constraint satisfaction problems [J]. *Theoretical Computer Science*, 2003, 290 (3): 1815-1828.

[2] Creignou N, Hervé Daudé. The SAT–UNSAT transition for random constraint satisfaction problems [J]. *Discrete Mathematics*, 2009, 309 (8): 2083-2099.

[3] Martin O C, Monasson R, Zecchina R. Statistical mechanics methods and phase transitions in optimization [J]. *Theory of Computing Systems*, 2001, 265: 3-67.

[4] Cook, Stephen. The Complexity of Theorem-Proving Procedures [C]// *Proceedings of the third annual ACM symposium on Theory of computing*, 1971: 151-158.

[5] Clarke E, Biere A, Raimi R, et al. Bounded model checking using satisfiability solving [J]. *Formal Methods in System Design*, 2001, 19 (1): 7-34.

[6] Vizel Y, Weissenbacher G, Malik S. Boolean Satisfiability Solvers and Their Applications in Model Checking [J]. *Proceedings of the IEEE*, 2015, 103 (11): 2021-2035.

[7] Kuper L, Katz G, Gottschlich J, et al. Toward Scalable Verification for Safety-Critical Deep Networks [J]. *arXiv preprint arXiv:1801.05940*, 2018.

[8] Braunstein A, Mezard M, Zecchina R. Survey propagation: an algorithm for satisfiability [J]. *Random Structures & Algorithms*, 2002, 27 (2): 201-226.

[9] Weiss Y. Correctness of Local Probability Propagation in Graphical Models with Loops [J]. *Neural Computation*, 2000, 12 (1): 1-41.

[10] Weiss Y, Freeman W T. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology [J]. *Neural Computation*, 2001, 13 (10): 2173-2200.

[11] Wang Xiaofeng, Xu Daoyun, Wei Li. Randomly satisfy the convergence of the warning propagation algorithm on the instance set [J]. *Journal of Software*, 2013 (01): 5-15.

[12] Wang Xiaofeng, Xu Daoyun. Sufficient conditions for convergence of warning propagation algorithm [J]. *Journal of Software*, 2016 (12): 3003-3013.

[13] Su Qinliang, Wu Y C. Convergence analysis of the variance in gaussian belief propagation [J]. *IEEE Transactions on Signal Processing*, 2014, 62 (19): 5119-5131.

[14] Su Qinliang, Wu Y C. On Convergence Conditions of Gaussian Belief Propagation [J]. *IEEE Transactions on Signal Processing*, 2015, 63 (5): 1144-1155.

[15] Feige U, Mossel E, Vilenchik D. Complete convergence of message passing algorithms for some satisfiability problems [J]. *Theory of Computing*, 2013, 9 (19): 617-651.

[16] Maneva E, Mossel E, Wainwright M. A new look at survey propagation and its generalizations [J]. *Journal of the ACM*, 2007, 54: 1089-1098.

[17] Shannon C. The lattice theory of information [J]. *Transactions of the IRE Professional Group on Information Theory*, 2003, 1 (1): 105-107.

[18] Li Angsheng, Pan Yichen. Structural Information and Dynamical Complexity of Networks [J]. *IEEE Transactions on Information Theory*, 2016, 62 (6): 3290-3339.

[19] Yedidia J S, Freeman W T, Weiss Y. Understanding belief propagation and its generalizations [M]. Morgan Kaufmann, 2003: 239-269.

[20] Kschischang F R, Frey B J, Loeliger H A. Factor graphs and the sum-product algorithm [J]. *IEEE Transactions on Information Theory*, 2001, 47 (2): 498-519.

[21] Wang Xin, Wang Xiaofeng, Li Weimin. A min-cut algorithm for warning propagation [J]. *Acta Electronica Sinica*, 2019, 47 (11): 2386-2391.

Note: Figure translations are in progress. See original paper for figures.