entropy

MDPI

Article

A Structural Entropy Measurement Principle of Propositional Formulas in Conjunctive Normal Form

Zaijun Zhang [1,2], Daoyun Xu [1,*] and Jincheng Zhou [2]

[1] College of Computer Science and Technology, Guizhou University, Guiyang 550025, China; zjzhang1987@outlook.com
[2] Key Laboratory of Complex Systems and Intelligent Computing and School of Mathematics and Statistics, Qiannan Normal University for Nationalities, Duyun 558000, China; guideaaa@126.com
* Correspondence: dyxu@gzu.edu.cn; Tel.: +86-851-83620667

Abstract: The satisfiability (SAT) problem is a core problem in computer science. Existing studies have shown that most industrial SAT instances can be effectively solved by modern SAT solvers while random SAT instances cannot. It is believed that the structural characteristics of different SAT formula classes are the reasons behind this difference. In this paper, we study the structural properties of propositional formulas in conjunctive normal form (CNF) by the principle of structural entropy of formulas. First, we used structural entropy to measure the complex structure of a formula and found that the difficulty solving the formula is related to the structural entropy of the formula. The smaller the compressing information of a formula, the more difficult it is to solve the formula. Secondly, we proposed a λ-approximation strategy to approximate the structural entropy of large formulas. The experimental results showed that the proposed strategy can effectively approximate the structural entropy of the original formula and that the approximation ratio is more than 92%. Finally, we analyzed the structural properties of a formula in the solution process and found that a local search solver tends to select variables in different communities to perform the next round of searches during a search and that the structural entropy of a variable affects the probability of the variable being flipped. By using these conclusions, we also proposed an initial candidate solution generation strategy for a local search for SAT, and the experimental results showed that this strategy effectively improves the performance of the solvers CCAsat and Sparrow2011 when incorporated into these two solvers.

Keywords: SAT problem; the structural complexity; structural entropy; CNF formula; structural properties

check for updates

Citation: Zhang, Z.; Xu, D.; Zhou, J. A Structural Entropy Measurement Principle of Propositional Formulas in Conjunctive Normal Form. *Entropy* **2021**, *23*, 303. https://doi.org/10.3390/e23030303

Received: 8 January 2021
Accepted: 27 February 2021
Published: 4 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Given a propositional formula in conjunctive normal form with variables $\{x_1, x_2, \ldots, x_n\}$ and clauses $\{c_1, c_2, \ldots, c_m\}$, where c_i is the disjunction of some literals, a literal refers to the variable x or its negation $\neg x$. The satisfiability problem consists of finding an assignment for the variables so that all clauses are satisfied. The satisfiability (SAT) problem is a core problem in computer science, and many real-world applications, such as hardware and software verification, planning, cryptography, scheduling, among others, can be directly or indirectly encoded as a SAT problem. Therefore, the development of high-performance SAT solvers has always been a hot research issue.

However, the SAT problem is NP-hard (non-deterministic polynomial time hard) [1], which means that there is no polynomial time algorithm to solve it. Many efforts have been made to design high-performance SAT solvers. It is recognized that solvers based on a complete search technique can achieve good results in solving industrial SAT instances, and solutions based on the stochastic local search technique show effectiveness in solving random SAT instances. The performance of these solvers shows obvious differences in different formula classes. It is speculated that this may be related to the complex structure of a SAT formula [2–5]. Different conjunctive normal form (CNF) formula classes have different organizational structures, which leads to the differences in performance of SAT solvers. In recent years, researchers have tried to clarify the structures of CNF formulas in order to improve or design high-performance SAT solvers.

Ansótegui et al. studied the community structure of industrial SAT instances from the perspective of complex networks [4,5]. They found that the community structure of industrial SAT instances were obvious while the random SAT instances were not. They verified by experiments that the SAT solvers based on conflict-driven clause learning (CDCL) destroyed the community structure of formulas during a search.

It is useful to study the community structure of CNF formulas. Newsham et al. [6,7] found that the solving time of CDCL solvers was related to the community structure of a SAT instance and used the community structure of the SAT instance to improve the performance of several SAT solvers. Martins [8] used the community structure of formulas to partition soft clauses and then used it to solve the MaxSAT problem. Sonobe et al. [9] improved the performance of a parallel SAT solver by using community structured partitioning. Giráldez-Cru and Levy [10,11] used the formula community to generate some highly modular pseudo-random industrial instances.

Although research on the complex structure of formulas has made some progress, it is still a challenging problem to explain the relationship between the difficulty solving a formula and its structure. In recent years, Li and Pan [12] proposed the structural information theory. They used random walks to capture information interaction between the nodes in graphs and defined the structural entropy of graphs. They defined the structural entropy of a graph to be the minimum overall number of bits required to determine the code of the node that is accessible from a random walk in the graph. In their definition, the structural entropy of graphs is essentially the metric that allows us to fully or maximally detect the K-dimensional structure consisting of the rules, regulations, and orders of the graphs against the random variations occurring in the graphs. It supports the full analysis of networking data and unstructured big data. They used the structural entropy of graphs to study the dynamic evolution of networks, information interaction between nodes, and natural clustering of nodes and made some new progress [13–15].

Inspired by structural information theory, in this work, we studied the structural properties of formulas by using the structural entropy of graphs. Combined with the previous research, this paper attempts to give an explanation of the relationship between the difficulty in solving and the complex structure of a formula. The goal is to explore the success rules of SAT solution techniques and to possibly improve them.

The first contribution of this work is to improve an algorithm to solve the two-dimensional structural entropy of graphs. We added a random perturbation to the algorithm $\mathcal{E}$ proposed in [12,16] and made it more diversified for the selection of maximum entropy increment. In addition, we also introduced a λ-approximation strategy to approximate the structural entropy of large formulas.

The second contribution of this work is to analyze the relationship between the structural entropy of a formula and the difficulty of its solution and to give an explanation of the difficulty solving the formula from the perspective of structural information theory. First, we represented a CNF formula as a variable graph and then calculated the one-dimensional and two-dimensional structural entropy, community information, and compressing information of the variable graph of the formula. The experimental results showed that the compressing information of a formula is inversely proportional to the difficulty solving the formula. In other words, the smaller the compressing information of a formula, the more complex the essential structure of the formula and, thus, the more difficult the formula is to solve. On the contrary, the bigger the compressing information of a formula, the simpler the essential structure of the formula and the easier it is to solve.

The third contribution of this work is to analyze structural properties of formulas in the process of solving. In particular, We focus on whether the entropy of a variable affects its flipping frequency during solving and focus on the relationship between the selected variables when the local search algorithms encounter a local optimum and the communities to which they belong. The experimental results show the provability of the variable to be assigned or flipped. The variable with large entropy is more likely to be flipped repeatedly. Based on this conclusion, we proposed an initial solution candidate solution generation strategy, which further improved the performance of the solvers CCAsat [17] and Sparrow2011 [18] when integrating this strategy. In addition, the experimental results also showed that a local search solver tends to select variables in different communities for flipping when using it to solve SAT.

The rest of this paper is organized as follows. Some necessary concepts and basic notations are introduced in Section 2. In Section 3, we introduce the methods of graph representation of a formula and provide some definitions about the structural entropy of graphs. In Section 4, we introduce an algorithm to calculate the two-dimensional structural entropy of graphs. An analysis of the structural properties of formulas and the relationship between the structural entropy of a formula and its hardness are presented in Section 5. In Section 6, we present our conclusions.

2. Preliminaries

Given a set of Boolean variables $X = \{x_1, x_2, \ldots, x_n\}$, a literal refers to the variable x or its negation $\neg x$. A clause c of length s is a disjunction of s literals, i.e., $c = l_1 \vee l_2 \vee \cdots \vee l_s$; we note $s = |c|$. We say that a variable $x \in c$ means that c contains the literal x or $\neg x$. A CNF formula or a SAT instance of length t refers to the conjunction of t clauses, $F = c_1 \wedge c_2 \wedge \cdots \wedge c_t$. A k-CNF formula means that the length of each clause in a CNF formula is k.

Solving a CNF formula F refers to finding a truth assignment τ on a set of boolean variables X such that all clauses in F are true under τ.

For a CNF formula F, we usually use $V(F)$ and $C(F)$ to represent the set of all variables and the set of all clauses that appear in the formula F, respectively, and use $N(x)$ to represent the neighboring variable set of the variable x in F. Two variables x and y are neighbors if and only if x and y appear together in at least one clause.

An undirected and weighted graph G is an ordered pair $G = (V, w)$, where V is a set of vertexes and w is a weighted function $w : V \times V \rightarrow R^+$. For $x, y \in V$, we have $w(x, y) = w(y, x)$.

Definition 1. *Given an undirected and weighted graph $G = (V, w)$, let $N(u)$ denote the set of neighbor nodes of node u in G. Define the weighted degree of u to be $d_u = \sum_{v \in N(u)} w(u, v)$. For a subset $U \subseteq V$, define the volume of U to be $vol(U) = \sum_{v \in U} d_v$. Define $vol(G) = \sum_{v \in V} d_v$ to be the volume of G.*

3. Graphical Representation and Structural Entropy of Formulas

3.1. Graphical Representation of Formulas

To calculate the structural entropy of a formula, we need to represent a given SAT instance as a graph. Among the existing graph models, the variable graph is a common graph model. In variable graph model, a CNF formula F is represented as an undirected and weighted graph $G = (V, w)$. The vertices in G represent the variables in F, i.e., $V = V(F)$, and the edges in G represent the relationship between two neighboring variables. If two variables x and y are neighbors, we connect them with an edge. w is the weighted function of edges in G.

Definition 2. *Given a SAT instance F over the set of variables X, the variable graph of F is an undirected and weighted graph $G = (X, w)$, where X is the boolean variables set, and the weight function is as follows:*

$$w(x, y) = \sum_{x, y \in c \wedge c \in F} \frac{1}{\binom{|c|}{2}}. \tag{1}$$

3.2. Structural Entropy of Graphs

Representing a formula as a graph is convenient for us to study the structural characteristics of the formula. Here, we first introduce the definitions of structural entropy of graphs in [12].

Definition 3. *Let $G = (V, w)$ be an undirected and weighted graph with n nodes, and the weight function is w. The one-dimensional structural entropy of G is defined as follows:*

$$\mathcal{H}^1(G) = -\sum_{i=1}^{n} \frac{d_i}{vol(G)} \log_2 \frac{d_i}{vol(G)}. \tag{2}$$

We call the one-dimensional structural entropy of the variable graph of a formula the one-dimensional structural entropy of the formula or the original information of the formula. As we know, entropy is a measure of uncertainty of random variables or random systems. If the entropy of a random variable or random system is larger, then the greater the uncertainty of the variable or the more uncertainty is embedded in the system, the worse the stability of the variable or the system. By calculating the one-dimensional structural entropy of the variable graph of a formula, we can know how much uncertainty is embedded in the formula.

Definition 4. *Let $G = (V, w)$ be an undirected and weighted graph with n nodes, and w is the weight function. Suppose that $\mathcal{P} = \{X_1, X_2, \ldots, X_l\}$ is a partition of vertices and that $\mathcal{P}$ is disjoint; we define the structural entropy of graph G by partition $\mathcal{P}$ as follows:*

$$\mathcal{H}^{\mathcal{P}}(G) = -\sum_{j=1}^{l} \frac{V_j}{vol(G)} \sum_{i \in X_j} \frac{d_i^j}{V_j} \log_2 \frac{d_i^j}{V_j} - \sum_{j=1}^{l} \frac{g_j}{vol(G)} \log_2 \frac{V_j}{vol(G)} \tag{3}$$

where X_j is called a module or a community, V_j is the volume of module X_j, l is the number of modules, n_j is the number of nodes in X_j, d_i^j is the weight degree of the ith node of X_j, and g_j is the sum of the weights of the edges with exactly one endpoint in module X_j.

Definition 5. *Given a graph G, the two-dimensional structural entropy of G is defined as follows:*

$$\mathcal{H}^2(G) = \min_{\mathcal{P}}\{\mathcal{H}^{\mathcal{P}}(G)\} \tag{4}$$

where $\mathcal{P}$ runs over all the partitions of G.

The two-dimensional structural entropy of a graph G is defined by the partition of vertices in G. It contains two parts. The former part, $-\sum_{j=1}^{l} \frac{V_j}{vol(G)} \sum_{i \in X_j} \frac{d_i^j}{V_j} \log_2 \frac{d_i^j}{V_j}$, refers to the number of bits needed to determine the code of node v in its own module X, where v is the node accessible from a step of the random walk in G. The second part is $-\sum_{j=1}^{l} \frac{g_j}{vol(G)} \log_2 \frac{V_j}{vol(G)}$, which refers to the number of bits needed to determine the code of a module X, where X is the module accessible from a step of the random walk from nodes outside of X.

It can be seen from Definitions 4 and 5 that the two-dimensional structural entropy of a graph is a quantitative measure of the information interaction between nodes in the graph. It measures the minimum amount of information to position the two-dimensional code of the node that is accessible from random walk in G with stationary distribution [12]. From the perspective of information theory and coding, the two-dimensional structural entropy of a graph describes the minimum bits required to encode a graph. Therefore, the two-dimensional structural entropy of a graph is understood as an essential structure of the graph and describes how much intrinsic information is hidden in the graph, which cannot be encoded or decoded by any encoder or decoder. By calculating the two-dimensional structural entropy of the variable graph of a formula, we can obtain the essential structure of the formula, i.e., we know how much intrinsic information is hidden in the formula.

It can be seen from the above definition that, for a weighted graph $G = (V, w)$, if there is a partition $\mathcal{P}$ of vertices V of G such that $\mathcal{H}^2(G) = \min_{\mathcal{P}}\{\mathcal{H}^{\mathcal{P}}(G)\}$, then $\mathcal{P}$ is an optimal partition of V. We call $\mathcal{P}$ a community structure of G, in which most of the edges are within a community and few of them connect vertices of distinct communities. We can obtain the modularity of $\mathcal{P}$ by partition $\mathcal{P}$. The modularity of $\mathcal{P}$ is defined by Newman and Girvan [19] as follows:

$$Q = \sum_{X_k \in \mathcal{P}} \left[\frac{\sum_{i,j \in X_k} w(i,j)}{vol(G)} - \left(\frac{\sum_{i \in X_k} \sum_{j \in V} w(i,j)}{2vol(G)} \right)^2 \right]. \quad (5)$$

Note that the modularity defined by Newman and Girvan [19] measures the quality of community structure of a network. They define modularity as the fraction of edges connecting vertices of the same community minus the expected fraction of edges for a random graph with the same number of vertices and same degree. It is different from the definition of the structural entropy of graphs. The structural entropy of graphs is an information theoretical measure of the quality of community structures of graphs. It has many properties, but the definition of modularity has only some of them [16].

In the next section, we present a greedy algorithm to approximate the two-dimensional structural entropy of graphs. By calculating the two-dimensional structural entropy of a graph, we can obtain a partition of vertices of a graph. Here, we continue to present the definition of the compressing information of graphs.

Definition 6. *Given a weighted graph G, we define the compressing information of graph G as* $\mathcal{C}(G) = \mathcal{H}^1(G) - \mathcal{H}^2(G)$*. The compression ratio of graph G can be defined as* $\rho(G) = \frac{\mathcal{C}(G)}{\mathcal{H}^1(G)}$.

The compressing information of a graph G is a quantitative measure, which measures how much uncertainty that can be eliminated is embedded into G. Eliminating the amount of uncertainty embedded in G, we can obtain the essential structure of G. The essential structure of graph G is the intrinsic information hidden in G that cannot be eliminated by any lossless encoding of G.

For a CNF formula F, if the two-dimensional structural entropy of the variable graph of F is larger, then the essential structure of the formula is more complicated. We have reason to believe that this formula is very difficult to solve. In the following sections, we construct experiments to reveal this problem from the perspective of structural information theory.

4. An Algorithm for Calculating Two-Dimensional Structural Entropy

In this section, We describe a greedy algorithm to approximate the two-dimensional structural entropy of graphs.

4.1. Algorithm $\mathcal{E}$

First, we introduce the concept of structural entropy increment as follows [16].

Definition 7. *Given a graph G, suppose that* $\mathcal{P} = \{P_1, P_2, \ldots, P_l\}$ *is a partition of vertices of G and that* $\mathcal{P}' = \{P_1, P_2, \ldots, P_{i-1}, P_{i+1}, \ldots, P_{j-1}, P_{j+1}, \ldots, P_l, P_{ij}\}$ *is the new partition after merging modules* P_i*,* P_j *in P, where* $P_{ij} = P_i \cup P_j$*. We define the structural entropy increment* $\Delta^{P}_{i,j}$ *as*

$$\Delta^{\mathcal{P}}_{ij}(G) = \mathcal{H}^{\mathcal{P}}(G) - \mathcal{H}^{\mathcal{P}'}(G). \quad (6)$$

As can be seen from Definition 7, if $\Delta^{\mathcal{P}}_{ij}(G) > 0$, then the partition after merging is better than before. An approximate greedy algorithm $\mathcal{E}$ for calculating the two-dimensional structural entropy of graphs is described as follows:

(1) First, each node v of graph G is divided into a single set. That is, if $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ is the initial partition, then $P_i = \{v_i\}$.
(2) For any $1 \leq i, j \leq n$, compute $\Delta_{ij}^{\mathcal{P}}(G)$.
(3) If there is no $1 \leq i, j \leq n$, such that $\Delta_{ij}^{\mathcal{P}}(G) > 0$, then terminate and output the partition $\mathcal{P}$. Otherwise, find i, j such that $\Delta_{ij}^{\mathcal{P}}(G)$ is maximized among all $\Delta_{ij}^{\mathcal{P}}(G)$, then merge P_i, P_j. Let $P_{ij} = P_i \cup P_j$, $\mathcal{P} = \{P_1, P_2, \ldots, P_{i-1}, P_{i+1}, \ldots, P_{j-1}, P_{j+1}, \ldots, P_l, P_{ij}\}$, and go back to (2).
(4) Compute the two-dimensional structural entropy of graphs by $\mathcal{P}$ according to Equation (3).

The time complexity of algorithm $\mathcal{E}$ is $\mathcal{O}(n \log^{\mathcal{O}(1)} n)$, which means that we can quickly calculate the two-dimensional structural entropy of a formula with a certain scale.

In algorithm $\mathcal{E}$, each iteration always selects the modules with the largest $\Delta_{ij}^{\mathcal{P}}(G)$ to merge. We think such conditions are too strict. In practice, we found that merging slightly smaller $\Delta_{ij}^{\mathcal{P}}(G)$ produces better results. Therefore, we add a random perturbation to the algorithm $\mathcal{E}$, and step (3) is modified as follows:

(3) If there is no $1 \leq i, j \leq n$, such that $\Delta_{ij}^{\mathcal{P}}(G) > 0$, then terminate and output the partition $\mathcal{P}$. Otherwise, Among the first q largest $\Delta_{ij}^{\mathcal{P}}(G)$, randomly select a $\Delta_{ij}^{\mathcal{P}}(G)$, find the corresponding i, j, and merge P_i, P_j to obtain the partition $\mathcal{P}$. Finally go back to (2).

Note that there are some restrictions on the first q largest $\Delta_{ij}^{\mathcal{P}}(G)$. Small increments of $\Delta_{ij}^{\mathcal{P}}(G)$ are useless to us. Therefore, the difference between the values of the first q largest $\Delta_{ij}^{\mathcal{P}}(G)$ should be small and the variance should be as small as possible. In our experiment, the value of q is set as follows: (i) Select the first s structural entropy increments $\Delta_{ij}^{\mathcal{P}}(G)$ in turn, and calculate the average structural entropy increment Δ_{mean}. (ii) If the maximum entropy increment $\Delta_{max} \cdot 0.99 < \Delta_{mean}$ and $s \leq 10$, then set $q = s$; otherwise, set $q = 10$.

4.2. λ-Approximation Strategy

In addition, in order to quickly calculate the two-dimensional structural entropy of some large instances, we propose a λ-approximation strategy. This strategy is based on the singular value decomposition (SVD) of matrices. In other words, an $m \times n$ real matrix A can be factored into $A = U\Sigma V^T$, where U and V are unit orthogonal matrices, Σ is a diagonal matrix, and the elements on the main diagonal are the eigenvalues.

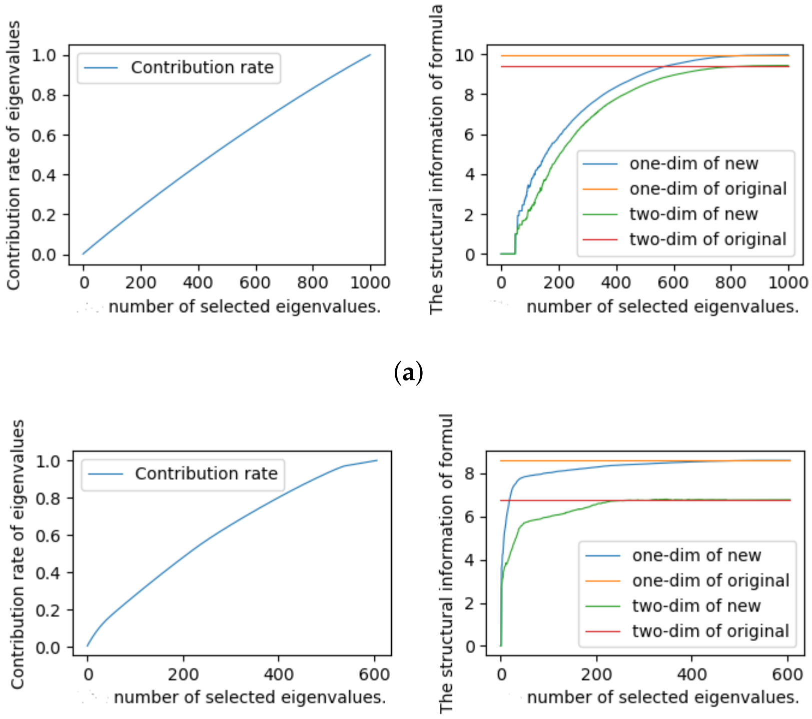
Given a CNF formula F with variables set X and clauses set C, our λ-approximation strategy algorithm is described as follows:

(1) Construct the formula matrix $M_{n\times m}$. Here, n is the number of variables and m is the number of clauses in F. If the variable x appears in clause c, then set $M_{x,c} = 1$ (positive occurrence) or $M_{x,c} = -1$ (negative occurrence). Otherwise, set $M_{x,c} = 0$.
(2) Compute the singular value of matrix M.
(3) Select the first λ eigenvalues in Σ to reconstruct the formula matrix $M^*_{n\times m}$, where $M^*_{n\times m} = U_{n\times\lambda}\Sigma^*_{\lambda\times\lambda}V^T_{\lambda\times m}$. Considering the matrix $M^*_{n\times m}$ as the adjacency matrix of a graph, we can calculate the one-dimensional and two-dimensional structural entropies of the graph by $M^*_{n\times m}$.

In the next subsection, we carry out experimental studies to illustrate the value of the parameter λ and verify the effectiveness of the strategy.

4.3. Experimental Evaluation

First, we designed some experiments to study the value of λ. We generated random 3-SAT instances (10 instances in each group) with different numbers of variables ($n = 20, 50, 100, 500, 1000$) and different numbers of clauses ($m = 91, 218, 403, 2130, 20,000$) for the experiments. At the same time, we also selected some instances from the SAT competition 2018 industrial benchmark for experiments. In the experiment, we calculated the eigenvalues of each formula matrix and sorted the eigenvalues from large to small. Then, we calculated the cumulative contribution rate of eigenvalues and the one-dimensional and two-dimensional structural entropy of formulas under the contribution rate. Some experimental results are shown in Figure 1. The experimental results of other instances are basically consistent with Figure 1. We will not show them here.



(**a**)

(**b**)

Figure 1. The contribution rate of eigenvalues and the approximate situation of structural information for instances A and B. The X-axis represents the number of selected eigenvalues in formula matrix, where the eigenvalues have been arranged from largest to smallest. (**a**) Random instance A (contains 1000 variables); (**b**) Industrial instance B (contains 600 variables).

As we can seen from Figure 1, we found that the structural entropy of the new formula can effectively approximate the structural entropy of the original formula when the cumulative contribution rate of selected eigenvalues exceeds 70% whether it is a random formula or an industrial formula. Therefore, in the subsequent experiments, we set the value of λ as the number of selected eigenvalues, where the cumulative contribution rate of selected eigenvalues was not less than 70%.

Second, we carried out experimental studies to evaluate the effectiveness of the λ-approximation strategy. We calculated the structural entropy and the running time of each group of instances on large random 5-SAT instances from the SAT competition 2009 benchmark by adopting the algorithm $\mathcal{E}$ and the λ-approximation strategy. These instances were relatively complex and larger than the random 3-SAT instances used in Figure 1. The experimental results are shown in Figure 2.

As can be seen from Figure 2, the average running time of our λ-approximation strategy is better than algorithm $\mathcal{E}$ on random 5-SAT instances. Compared with the original formula, the approximate ratio of one-dimensional and two-dimensional structural entropy of the new formula is more than 96%.

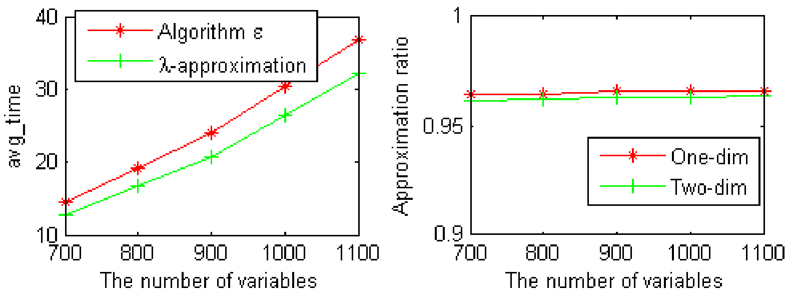


Figure 2. The average running time of two algorithms and the approximation ration.

At the same time, we also conducted experiments on some instances of the SAT competition 2018 industrial benchmark. The experimental results are shown in Figure 3. From Figure 3, we can see that the average running time of our λ-approximation strategy is slightly worse than that of algorithm $\mathcal{E}$ on classes *Jingchao* and *Biere*. Compared with the original formula, the approximate ratio of one-dimensional and two-dimensional structural entropies of the new formula is more than 92%.

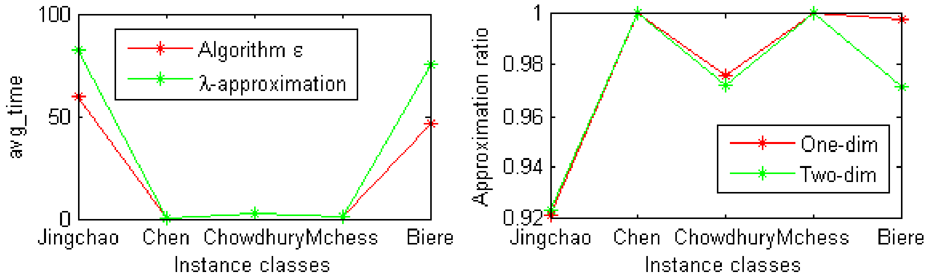


Figure 3. The average running time of two algorithms and the approximation ration.

5. Structural Characteristics of Formulas

The structure of a CNF formula is very complex, and the variables are related to each other by some clauses. In fact, the variable graph model we introduced is a good description of this complex structure. In general, a variable with more occurrences in a formula is subject to more constraints, and it should be connected with more edges. Our variable graph model just satisfies this condition. Therefore, analyzing the structural entropy of the variable graph of a formula may help us further understand the complex structure of the formula.

5.1. Structural Entropy of Formulas

To understand the complex structure of formulas, we carried out experiments to analyze the structural entropy of formulas. Our first experiment studied the structural entropy of random 3-SAT instances. We generated a number of random 3-SAT instances (10 instances in each group) with different clause to variable ratios $\alpha = m/n = 1, 2, \ldots, 7$, for a fixed number of variables $n = 10^4$. Simultaneously, we also generated a certain number of random 3-SAT instances (10 instances in each group) with a fixed constraint ratio $\alpha = 4.26$ and different number of variables $n = 10^2, 10^3, 10^4, 10^5$. Then, we reported the modularity Q of the partition returned by our algorithm as well as the one-dimensional *one-dim* and two-dimensional *two-dim* structural entropies of these formulas. We also reported the compressing information $\mathcal{C}$ and compression ratio ρ. Tables 1 and 2 show the results.

Table 1. The structural information of random 3-SAT instances with different α and $n = 10^4$.

n	$\alpha = m/n$	Q	ρ	*One-Dim*	*Two-Dim*	$\mathcal{C}$
10,000	1	0.419	0.321	13.027	8.841	4.186
10,000	2	0.228	0.172	13.162	10.901	2.261
10,000	3	0.161	0.118	13.206	11.652	1.554
10,000	4	0.130	0.090	13.227	12.033	1.194
10,000	4.25	0.125	0.085	13.231	12.103	1.128
10,000	4.5	0.121	0.081	13.234	12.161	1.072
10,000	5	0.113	0.074	13.239	12.260	0.979
10,000	6	0.101	0.064	13.247	12.402	0.845
10,000	7	0.091	0.057	13.253	12.498	0.756

Table 2. The structural information of random 3-SAT instances with different numbers of variables and fixed clause to variable ratios α.

n	$\alpha = m/n$	Q	ρ	*One-Dim*	*Two-Dim*	$\mathcal{C}$
100	4.26	0.164	0.132	6.587	5.718	0.868
1000	4.26	0.147	0.098	9.909	8.936	0.974
10,000	4.26	0.125	0.085	13.231	12.104	1.126
100,000	4.26	0.111	0.085	16.552	15.138	1.414

As we can see, the modularity Q and the compression ratio ρ of random 3-SAT instances with a fixed number of variables are very low (see Table 1). As the constraint ratio increases, the two-dimensional structural entropy of random 3-SAT instances gradually increases, which is very close to the one-dimensional structural entropy. We can also see that the compressing information $\mathcal{C}$ of these formulas decreases with the increase in the constraint ratio α. For random 3-SAT instances with a constraint ratio near the phase transition point (see Table 2), the modularity Q and compression ratio ρ did not change much with the increase in the number of variables n and their values are very low. The two-dimensional structural entropy of these formulas are very close to the one-dimensional structural entropy, and the compression ratio is less than 15%.

For random 3-SAT instances, the difficulty solving them gradually increases with the increase in the constraint ratio α and it is very difficult to solve when the constraint ratio is close to the phase transition point [20–22]. From the experiment, we observe that, as the constraint ratio increases, the two-dimensional structural entropy of this formulas gradually increases to be close to the one-dimensional structural entropy and the compressing information of formulas gradually decrease to less than 15% of the original information. We can conclude that the difficulty solving a formula is related to the compressing information of the formula. If the two-dimensional structural entropy of a formula is close to the one-dimensional structural entropy, that is, the compressing information of the formula is very little, then the intrinsic information hidden in the formula is bigger and the essential structure of the formula is more complex due to the large amount of uncertainty embedded in the formula, which makes the formula more difficult to solve.

In addition, we also observe that the community structure of random 3-SAT instances is not obvious except the constraint ratio $\alpha = 1$. This finding is consistent with the finding in [4,5]. Note that a network is considered to have a distinct community structure when the module value of the network is in the interval $[0.3, 0.7]$ [19].

In the second experiment, we analyzed the structural entropy of industrial SAT instances. As far as we know, it is relatively easy to solve industrial SAT instances. Now, we observe the structural entropy of industrial SAT instances by experiments and see if the compressing information of industrial SAT instances is relatively large. Our experiment was operated on the industrial benchmark from SAT competition 2018. We also report the one-dimensional and two-dimensional structural entropies, community structure, and compressing information of each instance class. The results are shown in Table 3.

Table 3. The structural information of industrial SAT instances.

Instance_Class	*#ins*	*Q*	ρ	*One-Dim*	*Two-Dim*	$\mathcal{C}$
biere	20	0.650	0.490	13.031	6.646	6.385
chen	20	0.159	0.103	8.319	7.464	0.855
chowdhury	19	0.534	0.332	9.249	6.179	3.069
deriendt	15	0.918	0.332	11.585	7.739	3.846
harder	11	0.803	0.366	15.930	10.093	5.837
heule	20	0.694	0.261	14.393	10.640	3.754
Heusser	17	0.600	0.413	16.737	9.827	6.910
jingchao	20	0.366	0.223	9.752	7.578	2.173
scheel	20	0.674	0.497	14.817	7.457	7.360
Porkhunov	9	0.670	0.423	10.674	6.159	4.514

In Table 3, the notations *instance_class* and *#ins* represent instance classes and the number of instances in each instance class, respectively. As you can see, the community structure of the industrial SAT instance is very obvious and it is consistent with the finding in [4,5]. The gap between two-dimensional structural entropy and one-dimensional structural entropy of industrial SAT instances is obvious, that is, the compressing information of industrial SAT instances is large. Meanwhile, we can also see that the compression ratio of industrial SAT instances is generally higher than 15%. This shows that a large amount of uncertainty embedded in the formula can be eliminated and that the essential structure of the formula is relatively simple. This may be one reason why industrial SAT instances are relatively easy to solve.

Through experiments, we can conclude that the structural entropy of a formula can approximately measure the complex structure of the formula. For a given CNF formula F, if the two-dimensional structural entropy of formula F is larger, then the essential structure of F is more complex. At the same time, we can see that the compressing information of a formula can approximately measure the difficulty solving the formula. If the compression information of a formula is larger, then a large amount of uncertainty embedded in formula can be eliminated, the essential structure of the formula is relatively simple, and the formula will be easier to solve. On the contrary, if the compressing information of a formula is smaller, then the uncertainty that can be eliminated in the formula is less, the essential structure of the formula is more complex, and the formula is more difficult to solve.

Now, a natural question is as follows: since the structural entropy of a formula can approximately measure the complex structure of the formula, does the structural entropy of the formula contribute to solving the formula? We continued to construct experiments to reveal this problem.

5.2. Structural Properties of Formulas during Solving

Given a formula F, G is the variable graph of F. The entropy of node u in G is defined as $H(u) = -\frac{d_u}{vol(G)} log_2 \frac{d_u}{vol(G)}$, which represents the uncertainty of random walks in G arriving at node u. The one-dimensional structural entropy of G is the sum of these uncertainties. Entropy describes the uncertainty of a random variable or a random system. In a variable graph, the greater the entropy of a node, the greater the degree of uncertainty freedom of the node. Therefore, we conjecture that the probability of the variable (node) with a large entropy to be flipped is high when solving the formula with a local search solver.

To verify our conjecture, we used the local search solver CCAsat to solve these random instances that were used in SAT competition 2009. At the same time, we also used the local search solver CCAsat to solve some industrial instances from SAT competition 2018. CCAsat is a competitive local search solver for random SAT instances, which was proposed by Cai in [17]. In the process of solving, we separately counted the number of times each variable was flipped and computed its entropy. Figure 4 shows the relationship between the frequency of a variable being flipped and its entropy for random instance unif-k3-r4.2-v2000-c8400-S1494472801-040 in the process of solving. Due to limited space, we do not show the statistical results for other instances, including industrial instances, but their results are highly consistent with the results shown in Figure 4.

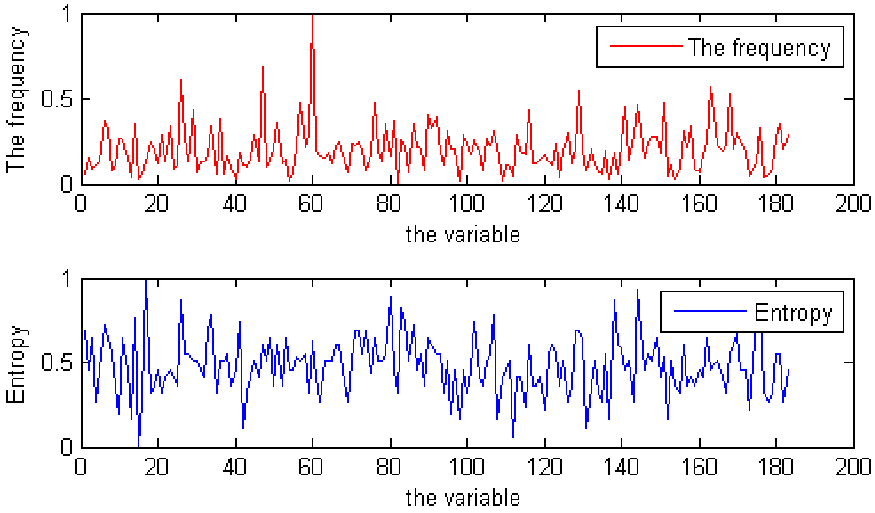


Figure 4. The flip times and entropy of variables (only some variables are shown) for the instance unif-k3-r4.2-v2000-c8400-S1494472801-040 during a search. These data have been normalized.

From the experiment, we can observe that, indeed, the greater the entropy of a variable, the greater the frequency of the variable being flipped. This conclusion is useful to us. On the one hand, it can further explain our previous conclusion, i.e., the difficulty solving a formula is related to the structural entropy of the formula. If there are many variables with large entropy in a formula, it will take more time to fix the values of these variables when solving the formula. Therefore, it is difficult to solve. On the other hand, we can use it to improve some local search solvers for SAT. For a given formula F, if the structural entropy of a variable is large, which means that the probability of this variable being repeatedly assigned is higher when solving F with a local search solver, then we can generate a better initial candidate solution according to the structural entropy of variables when solving F and use it to guide the algorithm to search.

Given a formula F, G is the variable graph of F. The process of generating an initial candidate solution with the structural entropy of variables is as follows.

(1) Calculate the structural entropy of nodes in G.
(2) Reorder variables according to the structural entropy (from small to large).
(3) Take out the sorted variables in turn and count their positive and negative occurrence times in the remaining unsatisfied clauses. If the times of positive occurrence are more than negative, then assign the value of the variable to 1 and to 0 otherwise.

This strategy can be effectively integrated into some other local search solvers for SAT. In our previous work, we integrated this method and clause weighting strategy into the solvers CCAsat and Sparrow2011 [18], resulting in solvers SICCAsat and SISparrow2011, which further improved their performance. Figure 5 shows the performance of these solvers on the SAT competition 2009 benchmark.

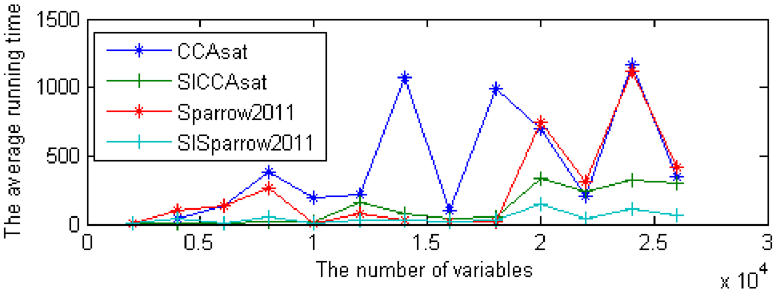


Figure 5. The performance comparison results of the solvers CCAsat, Sparrow2011, SICCAsat, and SISparrow2011 on the SAT competition 2009 benchmark.

We designed experiments to analyze the structure of a formula during a search. Ansótegui et al. [5] have verified by experiments that the community structure of industrial SAT instances is gradually destroyed with the increase in iteration times when using a complete solver CDCL to solve it. Here, we solved a SAT instance with the local search solver SICCAsat and observed whether the instance shows some structural characteristics during a search. In the experiment, we outputted each flipped variable and the community it belongs to when the solver encounters a local optimum. Figure 6 shows the results of the solver on a random instance. The numerical results are shown in Table 4. In Table 4, the notation *times* represents the number of times a algorithm encounters a local optimum, *var* represents the selected variables, and *com* represents the communities to which the selected variables belong. Since the results of other SAT instances are consistent with the one showed in Figure 6, we do not show more results here.

Table 4. The selected variables and the communities they belong to for the instance CBS_k3_n100_m403_b10_55 during a search.

Times	*var*	*com*	*Times*	*var*	*com*	*Times*	*var*	*com*	*Times*	*var*	*com*	*Times*	*var*	*com*
1	156	45	21	422	9	41	450	8	61	515	18	81	44	5
2	402	34	22	346	5	42	491	46	62	237	18	82	318	5
3	301	40	23	283	36	43	272	13	63	250	18	83	498	26
4	274	2	24	369	15	44	533	46	64	398	19	84	114	32
5	141	28	25	44	5	45	255	46	65	345	37	85	309	45
6	354	39	26	318	5	46	222	39	66	159	37	86	456	4
7	402	34	27	596	5	47	300	22	67	500	39	87	484	10
8	502	25	28	313	39	48	286	22	68	475	34	88	227	10
9	57	39	29	516	18	49	564	22	69	377	8	89	115	7
10	245	31	30	225	24	50	454	21	70	51	8	90	51	8
11	224	25	31	453	17	51	262	21	71	350	8	91	227	10
12	238	18	32	458	17	52	469	42	72	562	1	92	55	43
13	206	10	33	281	41	53	463	43	73	528	18	93	263	46
14	49	41	34	132	9	54	327	42	74	238	18	94	269	46
15	410	24	35	429	23	55	605	42	75	380	15	95	547	46
16	63	30	36	288	23	56	596	5	76	516	18	96	488	33
17	40	3	37	253	32	57	44	5	77	100	25	97	483	15
18	378	6	38	368	44	58	544	26	78	411	9	98	415	32
19	596	5	39	350	8	59	458	17	79	143	9	99	503	24
20	148	47	40	172	8	60	466	17	80	422	9	100	199	24

As shown in Figure 6, to jump out of the local optimum, the variables selected by the solver each time are different in most cases and the communities to which they belong are also different. This shows that a local search solver tends to select variables in different communities to flip on each iteration.

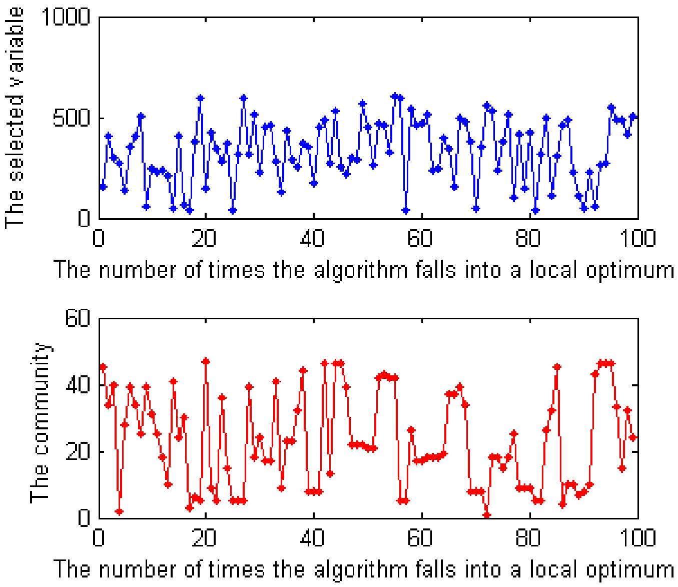


Figure 6. The selected variables and the communities to which they belong for the instance CBS_k3_n100_m403_b10_55 during a search. The horizontal axis represents the number of times that the algorithm encounters a local optimum when solving the instance with the local search solver SICCAsat, and the vertical axis represents the selected variables and the communities they belong to.

Combining the research of Ansótegui et al. [5], we can conclude that, when solving SAT instances with a solver, the solver tends to select variables among different communities to form a new learning clause (CDCL) or perform the next round of searches (local searches). This conclusion may be useful to us. After understanding the structural characteristics of a formula in the process of solving, we may be able to develop some high-performance SAT solvers.

6. Conclusions

Modern SAT solvers show obvious differences in solving different formula classes. The reason may be that different SAT formula classes have different structural characteristics. In this work, we studied the structural properties of CNF formulas and discussed the relationship between the structural properties of a formula and its solving difficulty.

Firstly, we represented a CNF formula as a graph, on which we studied the structural entropy and community structure of the formula. In the experiment, we found that the compressing information of a formula decreases with an increase in the constraint ratio for random 3-SAT instances. When the constraint ratio is close to the phase transition point, the compressing information of the formula gradually decreases to less than 15% of the original information. For industrial SAT instances, the compressing information is generally larger than 15% of the original information, and the gap between two-dimensional structural entropy and one-dimensional structural entropy is larger than random SAT instances. Therefore, we believe that the difficulty solving a formula is related to its structural entropy. Meanwhile, we gave an explanation of the difficulty solving a formula from the perspective of information theory. In addition, we also found that the community structure of industrial SAT instances are obvious while the random SAT instances are not.

Secondly, we proposed a λ-approximate strategy to approximate the two-dimensional structural entropy of formulas. The strategy is based on singular value decomposition of a matrix. The experimental results verified the effectiveness of the strategy on large random SAT instances.

Finally, we analyzed the structural properties of formulas in the process of solving. In particular, we experimentally verified that the structural entropy of a variable affects the frequency of the variable being flipped. By using this conclusion, we proposed an initial solution generation strategy for some stochastic local search solvers. Combined with the clause weight method, this strategy can effectively improve the performance of some local search solvers. In addition, we experimentally verified that, in solving SAT instances with a solver, the solver tends to select variables in different communities to form new learning clauses (complete solvers) or perform the next round of searches (local search solvers).

The study of the complex structure of formulas has significant implications for better understanding why some solvers perform better on industrial SAT instances and why others perform better on random SAT instances. Moreover, we can use these findings to develop or improve some existing solvers for SAT. Meanwhile, this analysis also serves as the basis for new random SAT generation models that produce more realistic pseudo-industrial random instances. This problem is distinguished as one of the ten challenge problems in SAT [23–25].

The findings in the present paper not only help us to design new SAT solvers but also help us to construct harder instances by the principle of structural entropy of CNF formulas.

Author Contributions: Conceptualization, Z.Z. and D.X.; methodology, Z.Z. and J.Z.; software, Z.Z., D.X., and J.Z.; validation, D.X.; formal analysis, Z.Z., J.Z., and D.X.; investigation, Z.Z. and J.Z.; resources, D.X.; writing—original draft preparation, Z.Z., D.X., and J.Z.; writing—review and editing, Z.Z., J.Z., and D.X.; supervision, D.X.; project administration, D.X.; funding acquisition, D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant numbers 61762019, 61862051, and 62062001), the Young Science and Technology Talent Growth Fund Project of Education Department of Guizhou Province of China (grant numbers Qian Jiao He KY Zi[2017]343, [2017]354, and [2018]411), the Science and Technology Foundation of Guizhou Province (grant number [2019]1299), the Top-notch Talent Program of of Guizhou province (grant number KY[2018]080), the Nature Science Foundation of Qiannan (grant number [2017]10), the Industrial Technology Foundation of Qiannan State of China (grant number Qiannan Ke He Gong Zi (2017) 16 Hao), and the General program of Qiannan Normal University for Nationalities of China (grant number qnsy2018016).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Some data benchmarks are available from http://satcompetition.org/ (accessed on 8 January 2021).

Acknowledgments: We would like to thank the anonymous referees for their helpful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cook, S.A. The Complexity of Theorem-Proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*; Association for Computing Machinery: New York, NY, USA, 1971; pp. 151–158. [CrossRef]
2. Williams, R.; Gomes, C.; Selman, B. Backdoors To Typical Case Complexity. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 9–15 August 2003.
3. Ansótegui, C.; Bonet, M.L.; Levy, J. Towards Industrial-Like Random SAT Instances. *Front. Artif. Intell. Appl.* **2008**, *184*, 109–117. [CrossRef]
4. Ansótegui, C.; Giráldez-Cru, J.; Levy, J. The community structure of SAT formulas. In *International Conference on Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 410–423. [CrossRef]
5. Ansótegui, C.; Bonet, M.L.; Giráldez-Cru, J.; Levy, J.; Simon, L. Community structure in industrial SAT instances. *J. Artif. Intell. Res.* **2019**, *66*, 443–472. [CrossRef]
6. Newsham, Z.; Ganesh, V.; Fischmeister, S.; Audemard, G.; Simon, L. Impact of community structure on SAT solver performance. In *International Conference on Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 252–268. [CrossRef]

7. Newsham, Z.; Lindsay, W.; Ganesh, V.; Liang, J.H.; Fischmeister, S.; Czarnecki, K. SATGraf: Visualizing the evolution of SAT formula structure in solvers. In *International Conference on Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9340, pp. 62–70. [CrossRef]
8. Martins, R.; Manquinho, V.; Lynce, I. Community-based partitioning for MaxSAT solving. In *International Conference on Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 182–191. [CrossRef]
9. Sonobe, T.; Kondoh, S.; Inaba, M. Community Branching for Parallel Portfolio SAT Solvers. In *International Conference on Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2014.
10. Giráldez-Cru, J.; Levy, J. Generating SAT instances with community structure. *Artif. Intell.* **2016**, *238*, 119–134. [CrossRef]
11. Giráldez-Cru, J.; Levy, J. Locality in random SAT instances. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 638–644. [CrossRef]
12. Li, A.; Pan, Y. Structural Information and Dynamical Complexity of Networks. *IEEE Trans. Inf. Theory* **2016**, *62*, 3290–3339. [CrossRef]
13. Li, A.; Zhang, X.; Pan, Y. Resistance maximization principle for defending networks against virus attack. *Phys. A Stat. Mech. Appl.* **2017**, *466*, 211–223. [CrossRef]
14. Li, A.; Yin, X.; Xu, B.; Wang, D.; Han, J.; Wei, Y.; Deng, Y.; Xiong, Y.; Zhang, Z. Decoding topologically associating domains with ultra-low resolution Hi-C data by graph structural entropy. *Nat. Commun.* **2018**, *9*, 3265. [CrossRef] [PubMed]
15. Li, A.; Yin, X.; Pan, Y. Three-Dimensional Gene Map of Cancer Cell Types: Structural Entropy Minimisation Principle for Defining Tumour Subtypes. *Sci. Rep.* **2016**, *6*, 20412. [CrossRef] [PubMed]
16. Li, A.; Li, J.; Pan, Y. Discovering natural communities in networks. *Phys. A Stat. Mech. Appl.* **2015**, *436*, 878–896. [CrossRef]
17. Cai, S.; Su, K. Local search for Boolean Satisfiability with configuration checking and subscore. *Artif. Intell.* **2013**, *204*, 75–98. [CrossRef]
18. Balint, A.; Frohlich, A. Improving Stochastic Local Search for SAT with a New Probability Distribution. In Proceedings of the 13th International Conference (SAT 2010), Edinburgh, UK, 11–14 July 2010.
19. Newman, M.; Girvan, M. Finding and Evaluating Community Structure in Networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef] [PubMed]
20. Kirkpatrick, S.; Selman, B.; Hill, M. Critical Behavior in the Satis ability of Random Boolean Formulae. *Science* **1994**, *264*, 1297–1301. [CrossRef] [PubMed]
21. Braunstein, A.; Mézard, M.; Zecchina, R. Survey propagation: An algorithm for satisfiability. *Random Struct. Algorithms* **2005**, *27*, 201–226. [CrossRef]
22. Mézard, M.; Zecchina, R. Random [formula presented]-satisfiability problem: From an analytic solution to an efficient algorithm. *Phys. Rev. E* **2002**, *66*, 056126. [CrossRef] [PubMed]
23. Selman, B.; Kautz, H.; McAllester, D. Ten challenges in propositional reasoning and search. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Nagoya, Japan, 23–29 August 1997; Volume 1, pp. 50–54.
24. Kautz, H.; Selman, B. Ten challenges redux: Recent progress in propositional reasoning and search. In *International Conference on Principles and Practice of Constraint Programming*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2833, pp. 1–18. [CrossRef]
25. Kautz, H.; Selman, B. The state of SAT. *Discret. Appl. Math.* **2007**, *155*, 1514–1524. [CrossRef]