An Improved Warning Propagation Algorithm for Solving Max-SAT Problems (Postprint)

Authors: Wu Yuxiang, Wang Xiaofeng, Ding Hongsheng, Yu Zhuo

Date: 2022-04-07T15:01:57Z

Abstract

Max-SAT is the optimization version of the SAT problem, with the objective of finding a variable assignment that maximizes the number of satisfied clauses in a given clause set. This problem is typically NP-hard. With the in-depth development of big data and artificial intelligence, existing algorithms are no longer adequate, and designing new solving algorithms or optimizing existing ones has become a current research hotspot. This paper addresses the limitations of warning propagation algorithms for solving random Max-3-SAT problems, proposes a warning propagation algorithm based on variable weight calculation, combines it with a random walk algorithm, and presents a novel algorithm called WWP+WalkSAT. By overcoming these solving limitations, it can better obtain a set of effective initial solutions, thereby enhancing the algorithm' s local search capability. Using benchmark instances from the 2016 Max-SAT International Competition, we conduct comparative experiments on accuracy between the WWP+WalkSAT algorithm and eight local search algorithms. The experimental results demonstrate that the WWP+WalkSAT algorithm exhibits favorable performance.

Full Text

Preamble

An Improved Warning Propagation Algorithm for Solving Max-SAT Problems

Wu Yuxiang[1], Wang Xiaofeng[1,2], Ding Hongsheng[1]†, Yu Zhuo[1]
(1. School of Computer Science & Engineering; 2. The Key Laboratory of Images & Graphics Intelligent Processing of State Ethnic Affairs Commission, North Minzu University, Yinchuan 750021, China)

Abstract: The Max-SAT problem is an optimized version of the SAT problem, where the goal is to find a variable assignment that satisfies the maximum number of clauses in a given set. This problem is a classic NP-hard problem. With the rapid advancement of big data and artificial intelligence, traditional algorithms are no longer adequate, making the design of new solution algorithms or the optimization of existing ones a current research focus. This paper addresses the limitations of the warning propagation algorithm in solving random Max-3-SAT problems by proposing a warning propagation algorithm based on variable weight calculation. Combined with a random walk algorithm, we present a novel algorithm called WWP+WalkSAT. By overcoming the limitations of the original approach, this method obtains a more effective set of initial solutions, thereby enhancing the local search capability of the algorithm. Using benchmark instances from the 2016 Max-SAT International Competition, we conducted comparative experiments on solution accuracy between WWP+WalkSAT and eight local search algorithms. Experimental results demonstrate that WWP+WalkSAT exhibits superior performance.

Keywords: satisfiability problem; maximum satisfiability problem; warning propagation algorithm; local search algorithm

0 Introduction

Combinatorial optimization problems play a crucial role in operations research, discrete mathematics, and computer science, with widespread applications in national defense, transportation, healthcare, and communications. Common combinatorial optimization problems include the knapsack problem, traveling salesman problem (TSP), vehicle routing problem (VRP), maximum clique problem (MCP), minimum vertex cover problem (MVC), and maximum satisfiability problem (Max-SAT). Among these, Max-SAT is a typical NP-hard problem and represents the optimization version of the satisfiability problem (SAT). Given a propositional formula in conjunctive normal form (CNF), which consists of a set of clauses combined by conjunction, where each clause comprises a disjunction of variables, the Max-SAT problem seeks to find a variable assignment that satisfies the maximum number of clauses. Max-SAT finds extensive applications in real-world scenarios such as combinatorial auctions, vehicle scheduling, and timetabling. Moreover, problems like maximum clique and vertex dominating sets in graph theory can be transformed into Max-SAT for solution.

Solution approaches for Max-SAT are primarily divided into complete algorithms and incomplete algorithms. Complete algorithms guarantee exact solutions but struggle with large-scale problems due to exponential time complexity. Recent research has focused on improving branch strategies, inference rules, and lower bound estimation, leading to effective algorithms such as WmaxSatz and MiniMaxSat. In contrast, incomplete algorithms can find optimal solutions for large-scale problems in relatively short time, improving efficiency but without guaranteeing solution accuracy. The main categories of incomplete algorithms include approximation algorithms, message passing algorithms, and local search algorithms.

Message passing algorithms are heuristic information transfer methods originating from statistical physics. Through marginal probability computation, these algorithms have been successfully applied across numerous domains. For propositional satisfiability problems, three primary message passing algorithms based on factor graphs exist: warning propagation (WP), belief propagation (BP), and survey propagation (SP). Currently, message passing algorithms represent the most effective approach for solving random SAT problems and have achieved promising results in various combinatorial optimization problems including graph coloring, maximum flow, and LDPC coding.

To address Max-SAT problems, researchers have proposed numerous effective algorithms. However, the convergence and effectiveness of message passing algorithms remain key research concerns. Convergence refers to the stabilization of iterative message values between successive iterations, while effectiveness refers to the algorithm' s ability to solve problems successfully. References [16–18] analyze the convergence and effectiveness of message passing algorithms, providing sufficient conditions for convergence. Reference [19] examines convergence based on specific instance generation models, offering probabilistic conditions for algorithm convergence.

Further research reveals that in random 3-SAT problems, the ratio of clause count m to variable count n, known as the constraint density α, significantly impacts both formula satisfiability and problem difficulty. As α increases, a phase transition occurs when $\alpha \approx 3.52$–4.48. Instances outside this phase transition region are typically easy to solve and highly likely to be satisfiable, while instances near the phase transition point are hard and highly likely to be unsatisfiable. Although message passing algorithms are highly effective for hard instances, they often fail to converge on easy instances outside the phase transition region.

To address this limitation, we propose a novel variable weight-based warning propagation algorithm called WWP+WalkSAT. By incorporating variable weight calculation into warning propagation, our approach obtains an effective set of initial solutions, which are then refined through random walk-based local search. This combination breaks the previous limitations of warning propagation for Max-SAT problems, enabling effective solution of instances across all regions while improving solution accuracy. Experimental results demonstrate that WWP+WalkSAT achieves superior performance compared to other local search algorithms across various instance types.

---

1 Basic Knowledge

The Max-SAT problem is formally defined as follows: Given a set of propositional variables $X = \{x_1, x_2, x_3, ..., x_n\}$, these variables form a set of clauses constituting a CNF formula. The objective is to maximize the number of satisfied clauses, or equivalently, minimize the number of unsatisfied clauses. The mathematical model for Max-SAT is expressed in equations (1) and (2):

Maximize: $\sum_{a=1}^{m} F_w(z_a \cdot \sum_{i \in S_a} x_i)$

Subject to: $x_i \in \{0,1\}, \forall i \in \{1,2,...,n\}$

Literal: Each Boolean variable $x_i \in X$ can appear as either a positive literal (x_i) or a negative literal $(\neg x_i)$.

Clause Set: A clause set $C = \{C_1, C_2, C_3, ..., C_m\}$ consists of m distinct clauses, where each clause contains one or more literals connected by disjunction. A clause is satisfied if at least one of its literals evaluates to 1; otherwise, it is unsatisfied. The number of literals in a clause is called its length. A clause containing only one literal is called a unit clause.

Conjunctive Normal Form (CNF Formula): A CNF formula $F = C_1 \wedge C_2 \wedge ... \wedge C_m$ is satisfiable if and only if every clause in the formula is satisfied.

Factor Graph: A factor graph is a bipartite graph containing two types of nodes: variable nodes (represented by circles, denoted $x_1, x_2, x_3, x_4, ...$) and function nodes (represented by squares, denoted $a, b, c, ...$). For a CNF formula $F = C_1 \wedge C_2 \wedge ... \wedge C_m$, all variables correspond to variable nodes, while clauses formed by disjunctions of variables correspond to function nodes. Edges in the graph represent connections between variables and clauses, with solid edges indicating positive literals and dashed edges indicating negative literals. For example, given the formula $F = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4)$, the corresponding factor graph is shown in Figure 1.

Notation Definitions: - $V(a)$ represents the set of variables appearing in clause a - $V^+(a)$ represents the set of variables appearing positively in clause a - $V^-(a)$ represents the set of variables appearing negatively in clause a - $V(a) \setminus \{i\}$ represents the set of variables in clause a excluding variable i - $V(i)$ represents the set of clauses containing variable i - $V^+(i)$ represents the set of clauses where variable i appears positively - $V^-(i)$ represents the set of clauses where variable i appears negatively - $V(i) \setminus \{a\}$ represents the set of clauses containing variable i excluding clause a

Consistency Sets: We define two consistency sets for variable i in clause a: - $U_a(i)$: The set of clauses (excluding a) containing variable i where the value assignment of i is inconsistent with its assignment in clause a - $S_a(i)$: The set of clauses (excluding a) containing variable i where the value assignment of i is consistent with its assignment in clause a

These sets are illustrated in the partial factor graph shown in Figure 2.

2 Variable Weight-Based Warning Propagation Algorithm

The WWP+WalkSAT algorithm improves upon the warning propagation algorithm [20]. Reference [16] analyzed the convergence properties of warning propagation by introducing parameters that transform message values from $\{0,1\}$ to $[0,1]$, utilizing contraction mapping properties in vector spaces to establish sufficient conditions for convergence and provide theoretical foundations for algorithm performance. Building upon this theoretical basis, WWP+WalkSAT incorporates variable weight calculation [21] to obtain an effective set of initial solutions. The goal is to select literals with the highest weights to maximize clause satisfaction, thereby reducing ineffective clause propagation and accelerating the search process. While traditional warning propagation is effective for hard instances, it often fails to converge on easy instances. To address this, our algorithm computes warning values with minimal local information change at the end of the iteration process, then performs variable weight calculation. Finally, the initial solution undergoes random walk-based local search to obtain improved solutions and enhance overall algorithm performance. WWP+WalkSAT breaks the previous limitations of warning propagation for Max-SAT problems, enabling effective solution of instances across all regions while improving solution accuracy.

2.1 Warning Propagation Algorithm

Message passing algorithms designed through information transfer demonstrate strong effectiveness for solving satisfiability problems. Warning propagation is an iterative algorithm where, in each iteration, every edge (a,i) in the factor graph receives a warning message representing a Boolean value transmitted from function node a to variable i, denoted as $u_{a\to i}$. The iterative update equation is shown in equation (4):

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

where t represents the iteration number and θ is a threshold function defined as:

$$\theta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \le 0 \end{cases}$$

This formulation indicates that when $u_{a\to i} = 1$, the satisfaction of clause a depends on variable i; when $u_{a\to i} = 0$, clause a cannot be satisfied by variable i alone, meaning its satisfaction depends on the values of other variables.

For each variable i, we compute a local cavity field h_i and a conflict field c_i as shown in equations (5) and (6):

$$h_i = \sum_{a \in V(i)} J_{a,i} \prod_{b \in V(i) \setminus \{a\}} u_{b \to i}$$

$$c_i = \sum_{a \in V(i)} \prod_{b \in V(i) \setminus \{a\}} u_{b \to i}$$

When $c_i > 0$, variable i receives conflicting constraints from two clauses, where one clause requires i to be 1 while another requires it to be 0. When $c_i = 0$, no conflict occurs for variable i, and its value can be determined by the local cavity field as shown in equation (7):

$$x_i = \begin{cases} 1 & \text{if } h_i > 0 \\ 0 & \text{if } h_i < 0 \\ \text{unassigned} & \text{if } h_i = 0 \end{cases}$$

2.2 Variable Weight Calculation

During iteration, basic warning propagation only assigns value 1 to variables with $c_i = 0$ and $h_i > 0$, assigns 0 when $h_i < 0$, and leaves variables with $c_i > 0$ unassigned. In solving Max-SAT problems, the quality of variable assignments directly impacts algorithm efficiency, as correct assignments enable more effective solutions.

We propose a novel heuristic variable weight calculation to determine variable assignments. Each unassigned variable possesses either a positive or negative weight, reflecting the difference between its positive and negative literal occurrences throughout the formula. The weight indicates the degree to which a variable should be positive or negative. A positive weight suggests the variable appears more frequently as a positive literal than a negative literal, and vice versa. The weight calculation is given by equation (8):

$$W_i = \frac{\text{NumberPosLit} - \text{NumberNegLit}}{\text{varNumberClause}}$$

where NumberPosLit counts positive literal occurrences, NumberNegLit counts negative literal occurrences, and varNumberClause represents the total number of clauses containing the variable.

For each unassigned variable i, if $W_i > 0$, we assign $x_i = 1$; if $W_i < 0$, we assign $x_i = 0$; and if $W_i = 0$, the variable remains unassigned temporarily.

2.3 Local Search Algorithm

Local search constitutes the majority of computational time in these algorithms. Conventional random search algorithms select initial solutions randomly, leading to high variability in solution quality. Good initial solutions enable efficient result finding, while poor ones waste time and degrade performance, often revisiting solutions and causing cycling behavior. Our variable weight-based warning propagation algorithm effectively generates a set of initial solutions that significantly benefit subsequent local search.

The WWP+WalkSAT algorithm first obtains variable assignments through variable weight calculation, constructing an initial solution. This initial solution may not be optimal in terms of satisfied clauses, requiring further refinement through local search. For Max-SAT problems, the objective is to find assignments maximizing satisfied clause count. We feed the initial solution into the WalkSAT algorithm for deeper search.

3 Experimental Results and Analysis

To thoroughly evaluate the effectiveness of our novel variable weight-based warning propagation algorithm for Max-SAT problems, we tested WWP+WalkSAT against WalkSAT, SA, GA, VNS-GA, CCLS2akms, CCEHC, Optirise6-in, and HS-Greedy. WalkSAT, SA, GA, and VNS-GA are classical heuristic algorithms, while CCLS2-akms, CCEHC, Optirise6-in, and HS-Greedy are solvers from the 2016 Max-SAT Evaluation competition. All experiments utilized random category datasets from the 2016 Max-SAT Evaluation.

Table 1 presents statistical comparisons among SA, WalkSAT, WWP+WalkSAT, GA, and VNS-GA on easy instances, using s3v70c1000 and s3v80c1000 datasets (3-SAT instances with 70 and 80 variables and 1000 clauses). Each dataset contains 10 instances, comparing the maximum number of satisfied clauses. Results show that for the 70-variable instances, WWP+WalkSAT achieved the best results in 7 out of 10 cases, with slightly inferior performance on the remaining 3. For 80-variable instances, it achieved the best results in 4 out of 10 cases, with marginally worse performance on the other 6. However, WWP+WalkSAT significantly outperformed WalkSAT in all cases, substantially increasing the number of satisfied clauses.

Tables 2 and 3 compare SA, WalkSAT, WWP+WalkSAT, CCEHC, CCLS2akms, Optirise6-in, and HS-Greedy. Table 2 uses 3-SAT instances with 70, 90, and 110 variables and clause counts ranging from 700 to 1100, with 10 instances per configuration. Results are reported as average precision (maximum satisfied clauses divided by total clauses, averaged over 10 instances). WWP+WalkSAT achieved 95%–99% precision, while CCEHC ranged from 95% to 97%. Among 15 datasets, WWP+WalkSAT ranked first in 12 cases, showing comparable performance to CCEHC and HS-Greedy but significantly outperforming SA, WalkSAT, CCLS2akms, and Optirise6-in. This demonstrates WWP+WalkSAT' s effectiveness for easy instances, surpassing conventional local search algorithms.

Table 3 presents results on hard instances using HG-3SAT-V300-C1000 and HG-3SAT-V250-C1200 datasets (3-SAT instances with 300 variables and 1000/1200 clauses). CCLS2akms failed to solve these problems, while WWP+WalkSAT achieved approximately 98-99% precision, slightly below CCEHC' s stable 99% but far superior to SA, WalkSAT, Optirise6-in, and HS-Greedy. On hard instances, WWP+WalkSAT performs slightly worse than CCEHC but excels on easy instances, confirming its effectiveness for random Max-3-SAT problems.

Additionally, we compared iteration counts between WWP+WalkSAT and WalkSAT across five datasets (Figures 3-7). Both algorithms ran for 1000 iterations. Except for dataset S3v110c1000 where WWP+WalkSAT' s initial result was slightly worse, WWP+WalkSAT outperformed WalkSAT from the start in all other cases. This occurs because WalkSAT uses random initial assignments that may occasionally be superior, while WWP+WalkSAT' s initial solutions derive from variable weight calculation, yielding better starting points. As iterations progress, WWP+WalkSAT' s satisfied clause count far exceeds WalkSAT' s. Although WWP+WalkSAT demonstrates excellent precision, its runtime is longer than WalkSAT' s, representing a direction for future improvement.

---

4 Conclusion

This paper presents the WWP+WalkSAT algorithm based on variable weight calculation for warning propagation. Experimental analysis demonstrates its advantages for both easy and hard Max-3-SAT instances, breaking the limitations of traditional message passing algorithms. These results provide valuable reference for future theoretical research. The algorithm also shows promise for practical applications in combinatorial auctions, vehicle scheduling, robot path planning, resource allocation, and social network analysis. However, the variable weight calculation increases computational complexity and time consumption. Future research will focus on variable pruning strategies and other methods to improve time efficiency.

References

[1] Wang Xizhao, He Yichao. Evolutionary algorithms for knapsack problems [J]. Journal of Software, 2017, 28 (1): 1-16.

[2] Gunduz M, Aslan M. DJAYA: A discrete Jaya algorithm for solving traveling salesman problem [J]. Applied Soft Computing, 2021, 105: 107282.

[3] Mousaei A, Taghaddos H, Nekouvaght Tak A, et al. Optimized Mobile Crane Path Planning in Discretized Polar Space [J]. Journal of Construction Engineering and Management, 2021, 147 (5): 04021036.

[4] Wang Xiaofeng, Yu Zhuo, Zhao Jian, et al. Algorithm analysis for maximum clique problem based on large-scale graphs [J/OL]. Computer Engineering: 1-15

[2022-03-11]. DOI: 10.19678/j.issn.1000-3428.005844.

[5] Masato, Suzuki, Yoshiyuki, et al. Statistical mechanics of the minimum vertex cover problem in stochastic block models [J]. Physical review. E, 2019, 100 (6-1): 62101-62101.

[6] He Kun, Zheng Jiongzhi. Survey on algorithms for maximum satisfiability problem [J]. Journal of Huazhong University of Science and Technology: Natural Science Edition, 2022, 50 (02): 82-95. DOI: 10.13245/j.hust.210966.

[7] Aiello A, Burattini E, Massarotti A. The Complexity of Theorem Proving Procedures [J]. Rairo Informat Théor, 1977, 11 (1): 75-82.

[8] Liu Yanli, Huang Fei, Zhang Ting. MaxSAT complete algorithm based on cyclic extension inference rules [J]. Journal of Nanjing University: Natural Sciences, 2015, 51 (04): 762-771. DOI: 10.13232/j.cnki.jnju.2015.04.014.

[9] Li Chumin, Xu Zhenxing, Coll J, et al. Combining Clause Learning and Branch-and-Bound for MaxSAT [C]// Proc of the International Conference on Principles and Practice of Constraint Programming. Montpellier: Springer, 2021: 38: 1-38: 18.

[10] Morgado A, F Heras, Liffiton M, et al. Iterative and core-guided MaxSAT solving: A survey and assessment [J]. Constraints, 2013, 18 (4): 478-534.

[11] Li Chumin, Felip Manyà, Nouredine Ould Mohamedou N O, et al. Resolution-based lower bounds in MaxSAT [J]. Constraints, 2010, 15 (4).

[12] Heras F, Larrosa J, Oliveras A. MiniMaxSat: a new weighted Max-SAT solver [C]// Springer Berlin Heidelberg. Springer Berlin Heidelberg, 2007: 1-12.

[13] Johnson D S. Approximation algorithms for combinatorial problems [J]. Journal of Computer & System Sciences, 1974.

[14] Park S, Shin J. Convergence and correctness of max-product belief propagation for linear programming [J]. Siam Journal on Discrete Mathematics, 2017, 31 (3): 2228-2246.

[15] Chu Yi, Luo Chuan, Cai Shaowei, et al. Empirical investigation of stochastic local search for maximum satisfiability [J]. Frontiers of Computer Science, 2019, 13 (1): 13.

[16] Wang Xiaofeng, Xu Daoyun. Sufficient condition for convergence of warning propagation algorithm [J]. Journal of Software, 2016, 27 (12): 3003-3013. DOI: 10.13328/j.cnki.jos.004940.

[17] Wang Xiaofeng, Xu Daoyun, Yang Deren, et al. Convergence analysis of belief propagation algorithm for satisfiability problem [J]. Journal of Software, 2021, 32 (05): 1360-1372. DOI: 10.13328/j.cnki.jos.005844.

[18] Shi X, Schonfeld D, Tuninetti D. Message error analysis of loopy belief propagation [C]// 2010 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2010: 2078-2081.

[19] Wang Xiaofeng, Xu Daoyun, Wei Li. Convergence of warning propagation algorithm on random satisfiable instances [J]. Journal of Software, 2013, 24 (01): 1-11.

[20] Braunstein A, Mezard M, Zecchina R. Survey propagation: an algorithm for satisfiability [J]. Random Structures & Algorithms, 2010, 27 (2): 201-226.

[21] Layeb A. A new greedy randomized adaptive search procedure for solving the maximum satisfiability [J]. International Journal of Operational Research, 2013, 17 (3): 1-17.

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