16. Navarro, G.: A guided tour to approximate string matching. ACM Comput. Surv. **33**, 31–88 (2001)

Thresholds of Random k-SAT

2002; Kaporis, Kirousis, Lalas

ALEXIS KAPORIS, LEFTERIS KIROUSIS
Department of Computer Engineering and Informatics,
University of Patras, Patras, Greece

Keywords and Synonyms

Phase transitions; Probabilistic analysis of a Davis–Putnam heuristic

Problem Definition

Consider n Boolean variables $V = \{x_1, \ldots, x_n\}$ and the corresponding set of $2n$ literals $L = \{x_1, \overline{x}_1 \ldots, x_n, \overline{x}_n\}$. A k-clause is a disjunction of k literals of distinct underlying variables. A random formula $\phi_{n,m}$ in k Conjunctive Normal Form (k-CNF) is the conjunction of m clauses, each selected in a uniformly random and independent way amongst the $2^k\binom{n}{k}$ possible k-clauses on n variables in V. The density r_k of a k-CNF formula $\phi_{n,m}$ is the clauses-to-variables ratio m/n.

It was conjectured that for each $k \geq 2$ there exists a critical density r_k^* such that asymptotically almost all (a.a.a.) k-CNF formulas with density $r < r_k^*$ ($r > r_k^*$) are satisfiable (unsatisfiable, respectively). So far, the conjecture has been proved only for $k = 2$ [3,11]. For $k \geq 3$, the conjecture still remains open but is supported by experimental evidence [14] as well as by theoretical, but nonrigorous, work based on Statistical Physics [15]. The value of the putative threshold r_3^* is estimated to be around 4.27. Approximate values of the putative threshold for larger values of k have also been computed.

As far as rigorous results are concerned, Friedgut [10] proved that for each $k \geq 3$ there exists a sequence $r_k^*(n)$ such that for any $\epsilon > 0$, a.a.a. k-CNF formulas $\phi_{n,\lfloor(r_k^*(n)-\epsilon)n\rfloor}$ ($\phi_{n,\lceil(r_k^*(n)+\epsilon)n\rceil}$) are satisfiable (unsatisfiable, respectively). The convergence of the sequence $r_k^*(n)$, $n = 0, 1, \ldots$ for $k \geq 3$ remains open.

Let now

$$\begin{aligned} r_k^{*-} &= \underline{\lim}_{n\to\infty} r_k^*(n) \\ &= \sup\{r_k : \Pr[\phi_{n,\lfloor r_k n\rfloor} \text{ is satisfiable } \to 1]\} \end{aligned}$$

and

$$\begin{aligned} r_k^{*+} &= \overline{\lim}_{n\to\infty} r_k^*(n) \\ &= \inf\{r_k : \Pr[\phi_{n,\lceil r_k n\rceil} \text{is satisfiable} \to 0]\}\,. \end{aligned}$$

Obviously, $r_k^{*-} \leq r_k^{*+}$. Bounding from below (from above) r_k^{*-} (r_k^{*+}, respectively) with an as large as possible (as small as possible, respectively) bound has been the subject of intense research work in the past decade.

Upper bounds to r_k^{*+} are computed by counting arguments. To be specific, the standard technique is to compute the expected number of satisfying truth assignments of a random formula with density r_k and find an as small as possible value of r_k for which this expected value approaches zero. Then, by Markov's inequality, it follows that for such a value of r_k, a random formula $\phi_{n,\lceil r_k n\rceil}$ is unsatisfiable asymptotically almost always. This argument has been refined in two directions: First, considering not all satisfying truth assignments but a subclass of them with the property that a satisfiable formula always has a satisfying truth assignment in the subclass considered. The restriction to a judiciously chosen such subclass forces the expected value of the number of satisfying truth assignments to get closer to the probability of satisfiability, and thus leads to a better (smaller) upper bound r_k. However, it is important that the subclass should be such that the expected value of the number of satisfying truth assignments can be computable by the available probabilistic techniques.

Second, make use in the computation of the expected number of satisfying truth assignments of *typical* characteristics of the random formula, i. e. characteristics shared by a.a.a. formulas. Again this often leads to an expected number of satisfying truth assignments that is closer to the probability of satisfiability (non-typical formulas may contribute to the increase of the expected number). Increasingly better upper bounds to r_3^{*+} have been computed using counting arguments as above (see the surveys [6,13]). Dubois, Boufkhad and Mandler [7] proved $r_3^{*+} < 4.506$. The latter remains the best upper bound to date.

On the other hand, for fixed and small values of k (especially for $k = 3$) lower bounds to r_k^{*-} are usually computed by algorithmic methods. To be specific, one designs an algorithm that for an as large as possible r_k it returns a satisfying truth assignment for a.a.a. formulas $\phi_{n,\lfloor r_k n\rfloor}$. Such an r_k is obviously a lower bound to r_k^{*-}. The simpler the algorithm, the easier to perform the probabilistic analysis of returning a satisfying truth assignment for a given r_k, but the smaller the r_k's for which a satisfying truth assignment is returned asymptotically almost always. In this context, backtrack-free DPLL algorithms [4,5] of increasing sophistication were rigorously analyzed (see the surveys [2,9]). At each step of such an algorithm, a literal is set to TRUE and then a *reduced* formula is obtained by (i) deleting clauses where this literal appears and by (ii) deleting the negation of this literal from the clauses it appears. At steps at which 1-clauses exist (known as forced steps), the selection of the literal to be set to TRUE is made so as a 1-clause becomes satisfied. At the remaining steps (known as free steps), the selection of the literal to be set to TRUE is made according to a heuristic that characterizes the particular DPLL algorithm. A free step is followed by a round of consecutive forced steps. To facilitate the probabilistic analysis of DPLL algorithms, it is assumed that they never backtrack: if the algorithm ever hits a contradiction, i. e. a 0-clause is generated, it stops and reports failure, otherwise it returns a satisfying truth assignment. The previously best lower bound for the satisfiability threshold obtained by such an analysis was $3.26 < r_3^{*-}$ (Achlioptas and Sorkin [1]).

The previously analyzed such algorithms (with the exception of the `Pure Literal` algorithm [8]) at a free step take into account only the clause size where the selected literal appears. Due to this limited information exploited on selecting the literal to be set, the reduced formula in each step remains random conditional only on the current numbers of 3- and 2-clauses and the number of yet unassigned variables. This retention of "strong" randomness permits a successful probabilistic analysis of the algorithm in a not very complicated way. However, for $k = 3$ it succeeds to show satisfiability only for densities up to a number slightly larger than 3.26. In particular, in [1] it is shown that this is the optimal value that can be attained by such algorithms.

Key Results

In [12], a DPLL algorithm is described (and then probabilistically analyzed) such that each free step selects the literal to be set to TRUE taking into account its *degree* (i. e. its number of occurrences) in the current formula.

Algorithm `Greedy` [Section 4.A in 12]

The first variant of the algorithm is very simple: At each free step, a literal with the maximum number of occurrences is selected and set to TRUE. Notice that in this greedy variant, a literal is selected irrespectively of the number of occurrences of its negation. This algorithm successfully returns a satisfying truth assignment for a.a.a. formulas with density up to a number slightly larger than 3.42, establishing that $r_3^{*-} > 3.42$. Its simplicity, contrasted with the improvement over the previously obtained lower bounds, suggests the importance of analyzing heuristics that take into account degree information of the current formula.

Algorithm `CL` [Section 5.A in 12]

In the second variant, at each free step t, the degree of the negation $\overline{\tau}$ of the literal τ that is set to TRUE is also taken into account. Specifically, the literal to be set to TRUE is selected so as upon the completion of the round of forced steps that follow the free step t, the marginal expected increase of the flow from 2-clauses to 1-clauses per unit of expected decrease of the flow from 3-clauses to 2-clauses is minimized. The marginal expectation corresponding to each literal can be computed from the numbers of its positive and negative occurrences. More specifically, if m_i, $i = 2, 3$ equals the expected flow of i-clauses to $(i-1)$-clauses at each step of a round, and τ is the literal set to TRUE at the beginning of the round, then τ is chosen so as to minimize the ratio $|\frac{\triangle m_2}{\triangle m_3}|$ of the differences $\triangle m_2$ and $\triangle m_3$ between the beginning and the end of the round. This has as effect the bounding of the rate of generation of 1-clauses by the smallest possible number throughout the algorithm. For the probabilistic analysis to go through, we need to know for each i, j the number of literals with degree i whose negation has degree j. This heuristic succeeds in returning a satisfying truth assignment for a.a.a. formulas with density up to a number slightly larger than 3.52, establishing that $r_3^{*-} > 3.52$.

Applications

Some applications of SAT solvers include Sequential Circuit Verification, Artificial Intelligence, Automated deduction and Planning, VLSI, CAD, Model-checking and other type of formal verification. Recently, automatic SAT-based model checking techniques were used to effectively find attacks on security protocols.

Open Problems

The main open problem in the area is to formally show the existence of the threshold r_k^* for all (or at least some) $k \geq 3$. To rigorously compute upper and lower bounds better than the ones mentioned here still attracts some interest. Related results and problems arise in the framework of variants of the satisfiability problem and also the problem of colorability.

Cross References

- ► Backtracking Based k-SAT Algorithms
- ► Local Search Algorithms for kSAT
- ► Maximum Two-Satisfiability
- ► Tail Bounds for Occupancy Problems

Recommended Reading

1. Achioptas, D., Sorkin, G.B.: Optimal myopic algorithms for random 3-sat. In: 41st Annual Symposium on Foundations of Computer Science, pp. 590–600. IEEE Computer Society, Washington (2000)
2. Achlioptas, D.: Lower bounds for random 3-sat via differential equations. Theor. Comput. Sci. **265**(1–2), 159–185 (2001)
3. Chvátal, V., Reed, B.: Mick gets some (the odds are on his side). In: 33rd Annual Symposium on Foundations of Computer Science, pp. 620–627. IEEE Computer Society, Pittsburgh (1992)
4. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Commun. ACM **5**, 394–397 (1962)
5. Davis, M., Putnam, H.: A computing procedure for quantification theory. J. Assoc. Comput. Mach. **7**(4), 201–215 (1960)
6. Dubois, O.: Upper bounds on the satisfiability threshold. Theor. Comput. Sci. **265**, 187–197 (2001)
7. Dubois, O., Boufkhad, Y., Mandler, J.: Typical random 3-sat formulae and the satisfiability threshold. In: 11th ACM-SIAM symposium on Discrete algorithms, pp. 126–127. Society for Industrial and Applied Mathematics, San Francisco (2000)
8. Franco, J.: Probabilistic analysis of the pure literal heuristic for the satisfiability problem. Annal. Oper. Res. **1**, 273–289 (1984)
9. Franco, J.: Results related to threshold phenomena research in satisfiability: Lower bounds. Theor. Comput. Sci. **265**, 147–157 (2001)
10. Friedgut, E.: Sharp thresholds of graph properties, and the k-sat problem. J. AMS **12**, 1017–1054 (1997)
11. Goerdt, A.: A threshold for unsatisfiability. J. Comput. Syst. Sci. **33**, 469–486 (1996)
12. Kaporis, A.C., Kirousis, L.M., Lalas, E.G.: The probabilistic analysis of a greedy satisfiability algorithm. Random Struct. Algorithms **28**(4), 444–480 (2006)
13. Kirousis, L., Stamatiou, Y., Zito, M.: The unsatisfiability threshold conjecture: the techniques behind upper bound improvements. In: A. Percus, G. Istrate, C. Moore (eds.) Computational Complexity and Statistical Physics, Santa Fe Institute Studies in the Sciences of Complexity, pp. 159–178. Oxford University Press, New York (2006)
14. Mitchell, D., Selman, B., Levesque, H.: Hard and easy distribution of sat problems. In: 10th National Conference on Artificial Intelligence, pp. 459–465. AAAI Press, Menlo Park (1992)
15. Monasson, R., Zecchina, R.: Statistical mechanics of the random k-sat problem. Phys. Rev. E **56**, 1357–1361 (1997)

Topology Approach in Distributed Computing

1999; Herlihy Shavit

MAURICE HERLIHY
Department of Computer Science, Brown University,
Providence, RI, USA

Keywords and Synonyms

Wait-free renaming

Problem Definition

The application of techniques from Combinatorial and Algebraic Topology has been successful at solving a number of problems in distributed computing. In 1993, three independent teams [3,15,17], using different ways of generalizing the classical graph-theoretical model of distributed computing, were able to solve *set agreement* a long-standing open problem that had eluded the standard approaches. Later on, in 2004, journal articles by Herlihy and Shavit [15] and by Saks and Zaharoglou [17] were to win the prestigious Gödel prize. This paper describes the approach taken by the Herlihy/Shavit paper, which was the first draw the connection between Algebraic and Combinatorial Topology and Distributed Computing.

Pioneering work in this area, such as by Biran, Moran, and Zaks [2] used graph-theoretic notions to model uncertainty, and were able to express certain lower bounds in terms of graph connectivity. This approach, however, had limitations. In particular, it proved difficult to capture the effects of multiple failures or to analyze decision problems other then consensus.

Combinatorial topology generalizes the notion of a graph to the notion of a *simplicial complex*, a structure that has been well-studied in mainstream mathematics for over a century. One property of central interest to topologists is whether a simplicial complex has no "holes" below a certain dimension k, a property known as *k-connectivity*. Lower bounds previously expressed in terms of connectivity of graphs can be generalized by recasting them in terms of k-connectivity of simplicial complexes. By exploiting this insight, it was possible to solve some open problems (k-set agreement, renaming), to pose and solve some new problems ([13]), and to unify a number of disparate results and models [14].

Key Results

A *vertex* $\vec{v}$ is a point in a high-dimensional Euclidean space. Vertexes $\vec{v}_0, \ldots, \vec{v}_n$ are *affinely independent* if $\vec{v}_1 - \vec{v}_0, \ldots, \vec{v}_n - \vec{v}_0$ are linearly independent. An *n-dimensional simplex* (or *n-simplex*) $S^n = (\vec{s}_0, \ldots, \vec{s}_n)$ is the convex hull of a set of $n + 1$ affinely-independent vertexes. For example, a 0-simplex is a vertex, a 1-simplex a line segment, a 2-simplex a solid triangle, and a 3-simplex a solid tetrahedron. Where convenient, superscripts indicate dimensions of simplexes. The $\vec{s}_0, \ldots, \vec{s}_n$ are said to *span* S^n. By convention, a simplex of dimension $d < 0$ is an empty simplex.

A *simplicial complex* (or complex) is a set of simplexes closed under containment and intersection. The *dimension* of a complex is the highest dimension of any of its