Maximum satisfiability problem

In computational complexity theory, the **maximum satisfiability problem** (**MAX-SAT**) is the problem of determining the maximum number of clauses, of a given Boolean formula in conjunctive normal form, that can be made true by an assignment of truth values to the variables of the formula. It is a generalization of the Boolean satisfiability problem, which asks whether there exists a truth assignment that makes all clauses true.

Example

The conjunctive normal form formula

$$(x_0 \lor x_1) \land (x_0 \lor \neg x_1) \land (\neg x_0 \lor x_1) \land (\neg x_0 \lor \neg x_1)$$

is not satisfiable: no matter which truth values are assigned to its two variables, at least one of its four clauses will be false. However, it is possible to assign truth values in such a way as to make three out of four clauses true; indeed, every truth assignment will do this. Therefore, if this formula is given as an instance of the MAX-SAT problem, the solution to the problem is the number three.

Hardness

The MAX-SAT problem is OptP-complete,[1] and thus NP-hard, since its solution easily leads to the solution of the boolean satisfiability problem, which is NP-complete.

It is also difficult to find an approximate solution of the problem, that satisfies a number of clauses within a guaranteed approximation ratio of the optimal solution. More precisely, the problem is APX-complete, and thus does not admit a polynomial-time approximation scheme unless P = NP.[2][3][4]

Weighted MAX-SAT

More generally, one can define a weighted version of MAX-SAT as follows: given a conjunctive normal form formula with non-negative weights assigned to each clause, find truth values for its variables that maximize the combined weight of the satisfied clauses. The MAX-SAT problem is an instance of Weighted MAX-SAT where all weights are 1.[5][6][7]

Approximation algorithms

1/2-approximation

Randomly assigning each variable to be true with probability 1/2 gives an expected 2-approximation. More precisely, if each clause has at least k variables, then this yields a $(1 - 2^{-k})$-approximation.[8] This algorithm can be derandomized using the method of conditional probabilities.[9]

(1-1/*e*)-approximation

MAX-SAT can also be expressed using an integer linear program (ILP). Fix a conjunctive normal form formula F with variables $x_1, x_2, ..., x_n$, and let C denote the clauses of F. For each clause c in C, let S^+_c and S^-_c denote the sets of variables which are not negated in c, and those that are negated in c, respectively. The variables y_x of the ILP will correspond to the variables of the formula F, whereas the variables z_c will correspond to the clauses. The ILP is as follows:

maximize	$\sum_{c \in C} w_c \cdot z_c$		(maximize the weight of the satisfied clauses)
subject to	$z_c \leq \sum_{x \in S_c^+} y_x + \sum_{x \in S_c^-} (1 - y_x)$	for all $c \in C$	(clause is true iff it has a true, non-negated variable or a false, negated one)
	$z_c \in \{0, 1\}$	for all $c \in C$.	(every clause is either satisfied or not)
	$y_x \in \{0, 1\}$	for all $x \in F$.	(every variable is either true or false)

The above program can be relaxed to the following linear program L:

maximize	$\sum_{c \in C} w_c \cdot z_c$		(maximize the weight of the satisfied clauses)
subject to	$z_c \leq \sum_{x \in S_c^+} y_x + \sum_{x \in S_c^-} (1 - y_x)$	for all $c \in C$	(clause is true iff it has a true, non-negated variable or a false, negated one)
	$0 \leq z_c \leq 1$	for all $c \in C$.	
	$0 \leq y_x \leq 1$	for all $x \in F$.	

The following algorithm using that relaxation is an expected (1-1/e)-approximation:[10]

1. Solve the linear program L and obtain a solution O
2. Set variable x to be true with probability y_x where y_x is the value given in O.

This algorithm can also be derandomized using the method of conditional probabilities.

3/4-approximation

The 1/2-approximation algorithm does better when clauses are large whereas the (1-1/*e*)-approximation does better when clauses are small. They can be combined as follows:

1. Run the (derandomized) 1/2-approximation algorithm to get a truth assignment X.
2. Run the (derandomized) (1-1/e)-approximation to get a truth assignment Y.
3. Output whichever of X or Y maximizes the weight of the satisfied clauses.

This is a deterministic factor (3/4)-approximation.[11]

Example

On the formula

$$F = \underbrace{(x \lor y)}_{\text{weight } 1} \land \underbrace{(x \lor \neg y)}_{\text{weight } 1} \land \underbrace{(\neg x \lor z)}_{\text{weight } 2+\epsilon}$$

where $\epsilon > 0$, the (1-1/*e*)-approximation will set each variable to True with probability 1/2, and so will behave identically to the 1/2-approximation. Assuming that the assignment of *x* is chosen first during derandomization, the derandomized algorithms will pick a solution with total weight $3 + \epsilon$, whereas the optimal solution has weight $4 + \epsilon$.[12]

State of the art

The state-of-the-art algorithm is due to Avidor, Berkovitch and Zwick,[13][14] and its approximation ratio is 0.7968. They also give another algorithm whose approximation ratio is conjectured to be 0.8353.

Solvers

Many exact solvers for MAX-SAT have been developed during recent years, and many of them were presented in the well-known conference on the boolean satisfiability problem and related problems, the SAT Conference. In 2006 the SAT Conference hosted the first **MAX-SAT evaluation** comparing performance of practical solvers for MAX-SAT, as it has done in the past for the pseudo-boolean satisfiability problem and the quantified boolean formula problem. Because of its NP-hardness, large-size MAX-SAT instances cannot in general be solved exactly, and one must often resort to approximation algorithms and heuristics[15]

There are several solvers submitted to the last Max-SAT Evaluations:

- Branch and Bound based: Clone, MaxSatz (based on Satz), IncMaxSatz, IUT_MaxSatz, WBO, GIDSHSat.
- Satisfiability based: SAT4J, QMaxSat.
- Unsatisfiability based: msuncore, WPM1, PM2.

Special cases

MAX-SAT is one of the optimization extensions of the boolean satisfiability problem, which is the problem of determining whether the variables of a given Boolean formula can be assigned in such a way as to make the formula evaluate to TRUE. If the clauses are restricted to have at most 2 literals, as in 2-satisfiability, we get the MAX-2SAT problem. If they are restricted to at most 3 literals per clause, as in 3-satisfiability, we get the MAX-3SAT problem.

Related problems

There are many problems related to the satisfiability of conjunctive normal form Boolean formulas.

- Decision problems:
  - 2SAT
  - 3SAT
- Optimization problems, where the goal is to maximize the number of clauses satisfied:
  - MAX-SAT, and the corresponded weighted version Weighted MAX-SAT
  - MAX-*k*SAT, where each clause has exactly *k* variables:
    - MAX-2SAT
    - MAX-3SAT
    - MAXEkSAT
  - The partial maximum satisfiability problem (PMAX-SAT) asks for the maximum number of clauses which can be satisfied by any assignment of a given subset of clauses. The rest of the clauses must be satisfied.
  - The soft satisfiability problem (soft-SAT), given a set of SAT problems, asks for the maximum number of those problems which can be satisfied by any assignment.[16]
  - The minimum satisfiability problem.
- The MAX-SAT problem can be extended to the case where the variables of the constraint satisfaction problem belong to the set of reals. The problem amounts to finding the smallest *q* such that the *q*-relaxed intersection of the constraints is not empty.[17]

See also

- Boolean Satisfiability Problem
- Constraint satisfaction
- Satisfiability modulo theories

External links

- http://www.satisfiability.org/
- https://web.archive.org/web/20060324162911/http://www.iiia.csic.es/~maxsat06/
- http://www.maxsat.udl.cat
- Weighted Max-2-SAT Benchmarks with Hidden Optimum Solutions (http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/max-sat-benchmarks.htm)
- Lecture Notes on MAX-SAT Approximation (http://www.cs.tau.ac.il/~azar/Methods-Class6.pdf)

References

1. M. Krentel (1988). "The complexity of optimization problems". *Journal of Computer and System Sciences*. **36** (3): 490–509. doi:10.1016/0022-0000(88)90039-6 (https://doi.org/10.1016%2F0022-0000%2888%2990039-6). hdl:1813/6559 (https://hdl.handle.net/1813%2F6559).

2. Mark Krentel. The Complexity of Optimization Problems (https://www.sciencedirect.com/science/article/pii/0022000088900396/pdf?md5=6fa18c741f2eb2d204433ebf681c0c70&pid=1-s2.0-0022000088900396-main.pdf). Proc. of STOC '86. 1986.
3. Christos Papadimitriou. Computational Complexity. Addison-Wesley, 1994.
4. Cohen, Cooper, Jeavons. A complete characterization of complexity for boolean constraint optimization problems (https://www.researchgate.net/profile/Martin_Cooper3/publication/221632891_Lecture_Notes_in_Computer_Science/links/02e7e5343108bf0ed3000000/Lecture-Notes-in-Computer-Science.pdf). CP 2004.
5. Vazirani 2001, p. 131.
6. Borchers, Brian; Furman, Judith (1998-12-01). "A Two-Phase Exact Algorithm for MAX-SAT and Weighted MAX-SAT Problems". *Journal of Combinatorial Optimization*. **2** (4): 299–306. doi:10.1023/A:1009725216438 (https://doi.org/10.1023%2FA%3A1009725216438). ISSN 1382-6905 (https://search.worldcat.org/issn/1382-6905). S2CID 6736614 (https://api.semanticscholar.org/CorpusID:6736614).
7. Du, Dingzhu; Gu, Jun; Pardalos, Panos M. (1997-01-01). *Satisfiability Problem: Theory and Applications : DIMACS Workshop, March 11-13, 1996* (https://books.google.com/books?id=_GOVQRL50kcC&dq=weighted+max+sat&pg=PA393). American Mathematical Soc. p. 393. ISBN 9780821870808.
8. Vazirani 2001, Lemma 16.2.
9. Vazirani 2001, Section 16.2.
10. Vazirani 2001, p. 136.
11. Vazirani 2001, Theorem 16.9.
12. Vazirani 2001, Example 16.11.
13. Avidor, Adi; Berkovitch, Ido; Zwick, Uri (2006). "Improved Approximation Algorithms for MAX NAE-SAT and MAX SAT". *Approximation and Online Algorithms*. Vol. 3879. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 27–40. doi:10.1007/11671411_3 (https://doi.org/10.1007%2F11671411_3). ISBN 978-3-540-32207-8.
14. Makarychev, Konstantin; Makarychev, Yury (2017). "Approximation Algorithms for CSPs" (https://doi.org/10.4230%2FDFU.VOL7.15301.287). *Drops-Idn/V2/Document/10.4230/Dfu.vol7.15301.287*: 39 pages, 753340 bytes. doi:10.4230/DFU.VOL7.15301.287 (https://doi.org/10.4230%2FDFU.VOL7.15301.287). ISSN 1868-8977 (https://search.worldcat.org/issn/1868-8977).
15. Battiti, Roberto; Protasi, Marco (1998). "Approximate Algorithms and Heuristics for MAX-SAT" (https://link.springer.com/chapter/10.1007/978-1-4613-0303-9_2). *Handbook of Combinatorial Optimization*. pp. 77–148. doi:10.1007/978-1-4613-0303-9_2 (https://doi.org/10.1007%2F978-1-4613-0303-9_2). ISBN 978-1-4613-7987-4.
16. Josep Argelich and Felip Manyà. Exact Max-SAT solvers for over-constrained problems (https://doi.org/10.1007%2Fs10732-006-7234-9). In Journal of Heuristics 12(4) pp. 375-392. Springer, 2006.
17. Jaulin, L.; Walter, E. (2002). "Guaranteed robust nonlinear minimax estimation" (http://www.ensta-bretagne.fr/jaulin/paper_qminimax.pdf) (PDF). *IEEE Transactions on Automatic Control*. **47** (11): 1857–1864. doi:10.1109/TAC.2002.804479 (https://doi.org/10.1109%2FTAC.2002.804479).

- Vazirani, Vijay V. (2001), *Approximation Algorithms* (https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Approximation%20Algorithms%20%5bVazirani%202010-12-01%5d.pdf) (PDF), Springer-Verlag, ISBN 978-3-540-65367-7
