



**HAL**  
open science

## A randomized, efficient algorithm for MAX2SAT

Cristian Dumitrescu

► **To cite this version:**

| Cristian Dumitrescu. A randomized, efficient algorithm for MAX2SAT. 2016. hal-01423189

**HAL Id: hal-01423189**

**<https://hal.science/hal-01423189>**

Preprint submitted on 7 Jan 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A randomized, efficient algorithm for MAX2SAT

Cristian Dumitrescu  
Independent Researcher  
cristiand43@gmail.com

**Abstract.** In this paper I present a MAX2SAT algorithm based on the randomized algorithm of Papadimitriou from 1991. We also show that this algorithm finds a solution (if it exists) for a MAX2SAT problem equivalent to a 3SAT problem with high probability in polynomial time.

**Keywords.** NP-complete problems, MAX2SAT

## 1 Introduction

In MAX2SAT, we are given a 2SAT expression involving a set of clauses, each with at most two literals in it, and an integer  $K$ . We are asked whether there is a truth assignment that satisfied at least  $K$  of the clauses. We call this the MAX2SAT problem. We note that if the 2SAT expression considered has  $m$  clauses, then the truth assignment mentioned above must leave at most  $m - K$  clauses unsatisfied.

**Definition 1.** We know that there is (at least) a truth assignment that satisfies a maximum number of clauses  $M$  and leaves unsatisfied a minimum number of clauses  $m - M$ , where  $m$  represents the number of clauses of the 2SAT expression under consideration. The truth assignment satisfying this condition will be called the maximal truth assignment (there is at least one).

**Proposition 1.** (see [Papadimitriou, 94] and [Hopcroft, 1979]). MAX2SAT is NP-complete.

**Proof.** For the proof, see the references listed above (see [Papadimitriou, 94], theorem 9.2, page 186).

## 2. The presentation of the algorithm

We will present first the well known randomized algorithm for 2SAT. This is the Papadimitriou algorithm from 1991 (see [Papadimitriou, 1991]). We also note that Schoning discussed a similar algorithm for 3SAT in 1991 (see [Schoning, 1991]).

*Papadimitriou's algorithm for 2SAT.*

*Input: a 2SAT expression in  $n$  variables.*

*Guess an initial truth assignment, uniformly at random.*

*Repeat  $C \cdot n^2$  times:*

*If the formula is satisfied by the actual assignment, stop and accept.*

*Let  $C$  be some clause not being satisfied by the actual assignment. Pick one of the (at most) two literals in the clause at random, and flip its truth value.*

*Update.*

*Stop and reject, the expression is not satisfiable.*

In the next section, I will prove that this algorithm performs well even for the MAX2SAT problem, so in the Papadimitriou algorithm, all we need to modify is the definition of "satisfies" (related to an assignment), since in the context of MAX2SAT, we will not need all the clauses to be satisfied.

*The MAX2SAT algorithm.*

*Input: a 2SAT expression with  $m$  clauses.*

*Guess an initial truth assignment, uniformly at random.*

*Repeat  $C \cdot m^2$  times:*

*If the actual truth assignment satisfies the MAX2SAT condition, stop and accept.*

*Let  $C$  be some clause not being satisfied by the actual assignment. Pick one of the (at most) two literals in the clause at random, and flip its truth value.*

*Update.*

*Stop and reject, the expression is not satisfiable.*

In other words, I will prove that if a MAX2SAT satisfying assignment exists, then the MAX2SAT algorithm will find it with high probability. Note that running the algorithm a quadratic number of times, in the number of clauses is stronger than running it a quadratic number of times in the number of variables. A given 2SAT expression with  $m$  clauses can have at most  $2m$  variables, and at least a number of variables of order of magnitude  $\sqrt{m}$  (up to a multiplicative constant).

### 3 Analysis of the algorithm

In this section, I will present the main theorem. We assume that the reader is also familiar with the 3SAT to MAX2SAT reduction. For reference, see [Papadimitriou, 94], theorem 9.2, page 186. A given MAX2SAT problem is called a reduction of a 3SAT problem if it is equivalent to a given 3SAT problem. Basically, given an instance of 3SAT, we construct an instance of MAX2SAT as follows. For each variable or literal  $x$ , we write  $x'$  for its negation. For each 3SAT clause  $(x \vee y \vee z)$ , we add to the MAX2SAT expression the following conjunction of ten clauses:

$$(x) \wedge (y) \wedge (z) \wedge (w) \wedge (x' \vee y') \wedge (y' \vee z') \wedge (z' \vee x') \wedge (x \vee w') \wedge (y \vee w') \wedge (z \vee w')$$

Here, the variable  $w$  is particular to the chosen 3SAT clause. If the 3SAT expression has  $m$  clauses, then the MAX2SAT expression thus constructed has  $10 \cdot m$  clauses. We set the MAX2SAT threshold at  $7 \cdot m$ . Then, there is a MAX2SAT maximal truth assignment that satisfies  $7 \cdot m$  clauses if and only if the original 3SAT expression is satisfiable.

**Theorem.** Given a MAX2SAT problem, that is a reduction of a given 3SAT problem with  $m$  clauses, then with high probability, after  $C \cdot m^2$  steps, the MAX2SAT algorithm will find a MAX2SAT satisfying assignment (where  $C$  is a constant and  $m$  is the number of clauses in the original 3SAT expression).

**Proof.** For the proof, we will use the principle of induction, following the number of clauses in the MAX2SAT expression (actually the number of clauses of the original 3SAT expression). We will consider the Markov chain approach considered in [Schoning, 1991] and [Papadimitriou, 1991]. We will also consider the expected travelled distance (by the Markov chain associated to the algorithm) towards the absorbing barrier 0 (note that the direction of travel is important).

We are given a 3SAT problem with  $m$  clauses. We construct the equivalent MAX2SAT problem. The MAX2SAT problem will have  $10 \cdot m$  clauses, each 3SAT clause will be replaced by a group of 10 clauses. The 3SAT problem is satisfiable if and only if its MAX2SAT reduction has a maximal truth assignment with  $7 \cdot m$  satisfied clauses.

We write  $P(N)$  for the statement : after  $C \cdot N^2$  steps (where  $10 \cdot N$  is the number of clauses in the MAX2SAT expression that is equivalent to a 3SAT expression with  $N$  clauses), the Markov chain associated to the MAX2SAT algorithm will have an expected travel distance (towards the absorbing state with Hamming distance 0) given by the square root of the number of flips performed by the algorithm (up to a multiplicative constant), on any MAX2SAT expression that is equivalent to a 3SAT expression with  $N$  clauses (or else the chain will hit the absorbing state). So if the algorithm performs  $C \cdot N^2$  steps, then the expected travel distance is  $C_1 \cdot \sqrt{C} \cdot N$  (where  $C_1$  is a constant greater than 1).

We start with  $N = 2$ . For  $N = 2$ , the original 3SAT expression has only two clauses.

We can consider all possible combinations, it is a simple case by case analysis. By suitably choosing the constant  $C$ , it is clear that the statement  $P(2)$  is true, .

We want to prove that if we assume that  $P(\alpha)$  is true for all integers  $\alpha < N$ , then  $P(N)$  is also true.

Let's consider a 2SAT expression  $\varphi$  with  $10 \cdot N$  clauses (the MAX2SAT equivalent to a 3SAT expression with  $N$  clauses). We consider the expression  $\psi$ , by eliminating a group of 10 clauses (associated to one of the clauses of the original 3SAT expression). We call this group, the special group of 10 clauses. The expression  $\psi$  will have  $10 \cdot (N - 1)$  clauses. When working on the expression  $\varphi$ , we assume that the algorithm will touch this special group of clauses a number of times  $K$ . That means that if the algorithm takes  $C \cdot N^2$  steps (where  $N$  is the number of clauses in the expression  $\varphi$ ), then  $C \cdot N^2 - K$  steps will be taken within the expression  $\psi$ , and  $K$  steps will involve the special group of clauses considered above.

We assume that the associated Markov chain will take  $n_1$  steps within the expression  $\psi$ , then it will touch the special group of clauses considered above, then it will take  $n_2$  steps within the expression  $\psi$ , then it will touch the special group of clauses considered above again, and so on, until it will take  $n_K$  steps within the expression  $\psi$ , then it will touch the special group of clauses considered above for the last time, and then it will take the last  $n_{K+1}$  steps within the expression  $\psi$ .

We are only interested in studying what happens if the original 3SAT expression is satisfiable.

Any group of 10 clauses (as described above) will have at most 6 unsatisfied clauses, and at least 3 unsatisfied clauses. The whole expression  $\varphi$  will have at least  $3 \cdot N$  unsatisfied clauses in the maximal truth assignment. As long as the algorithm does not find a maximal truth assignment, the probability of choosing an unsatisfied clause from the special group (at every step of the algorithm) is at most  $p = \frac{6}{3N} = \frac{2}{N}$ . That means that if the algorithm makes  $C \cdot N^2$  steps, then the expected number of hits of an unsatisfied clause from the special group is at most  $2 \cdot C \cdot N$ . In other words, the inequality  $K < 2 \cdot C \cdot N$  will be satisfied with high probability.

From the inductive hypothesis, when the algorithm works within  $\psi$ , it will drift towards zero with an expected speed, and when it touches an unsatisfied clause in the special group of clauses, the worst it can do is to backtrack one unit (away from the absorbing state 0).

That means that we have the following equation (this counts the number of steps within  $\psi$ , and when it touches the unsatisfied clause considered above).

$$n_1 + n_2 + n_3 + \dots + n_{K+1} = C \cdot N^2 - K \quad (1)$$

The overall, expected travel distance of the associated Markov chain (when the

algorithm works on  $\varphi$ ), will be at least:

$$D = C_1 \cdot ((n_1)^{\frac{1}{2}} + (n_2)^{\frac{1}{2}} + \dots + (n_{K+1})^{\frac{1}{2}}) - K \quad (2)$$

We also work under the conditions  $n_i \geq 1$ , for all  $i \in \{1, 2, 3, \dots, K+1\}$ , and we assume that  $K < 2 \cdot C \cdot N$ , as we discussed before.

We want to find the minimum value that  $D$  can take, under the constraint given by equation (1). We can apply the method of Lagrange multipliers, more precisely the Karush-Kuhn-Tucker minimization conditions, and we find that the minimum of  $D$  has the form  $C_2 \cdot \sqrt{C} \cdot N$ , where the constant  $C_2$  is greater than 1 (for a suitably chosen constant  $C$ ).

There is also a geometrical way to see this, but this is only an approximation. We consider the variables  $z_i = n_i^{\frac{1}{2}}$ , for all  $i \in \{1, 2, 3, \dots, K+1\}$ . We have to study the intersection of a hyperplane and a hypersphere (a  $K$ -sphere), in the region where all the coordinates are greater or equal to 1. The hyperplane is given by the equation  $C_1 \cdot (z_1 + z_2 + \dots + z_{K+1}) = \text{constant}$ , and the hypersphere is given by the equation  $z_1^2 + z_2^2 + z_3^2 + \dots + z_{K+1}^2 = C \cdot N^2 - K$ .

Let's study the KKT conditions (see [KKT, 2012]). We define the following functions:

In the following, we have  $x = (x_1, x_2, x_3, \dots, x_{K+1})$  and  $\lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{K+1})$

$$f(x) = C_1 \cdot (z_1 + z_2 + \dots + z_{K+1}) - K$$

$$h(x) = z_1^2 + z_2^2 + z_3^2 + \dots + z_{K+1}^2 - C \cdot N^2 + K$$

$$g_i(x) = 1 - x_i, \text{ for } i = 1, 2, 3, \dots, K+1$$

$$L(x, \mu, \lambda) = f(x) + \mu \cdot h(x) + \sum_{i=1}^{K+1} \lambda_i \cdot g_i(x)$$

$$L(x, \lambda) = f(x) + \sum_{i=1}^{K+1} \lambda_i \cdot g_i(x)$$

Then, the KKT result states that  $x^*$  is a local minimum if and only if there is a unique  $\lambda^*$ , such that the following conditions are satisfied:

$$\nabla_x L(x^*, \mu^*, \lambda^*) = 0 \quad (3)$$

$$\lambda_i^* \geq 0, \text{ for } i = 1, 2, 3, \dots, K+1 \quad (4)$$

$$\lambda_i^* \cdot g_i(x^*) = 0, \text{ for } i = 1, 2, 3, \dots, K+1 \quad (5)$$

$$g_i(x^*) \leq 0, \text{ for } i = 1, 2, 3, \dots, K+1 \quad (6)$$

$$h(x^*) = 0 \quad (7)$$

$$\text{The matrix } \nabla_{xx} L(x^*, \lambda^*) \text{ is positive semi-definite.} \quad (8)$$

We consider the general case, in which we can assume that

$\lambda_1 = \lambda_2 = \dots = \lambda_m = 0$  with  $1 \leq m \leq K$ , and also  $x_{m+1} = x_{m+2} = \dots = x_{K+1} = 1$ . This is the most general case for which relations (5) will be satisfied. We also emphasize that  $m$  must be greater or equal to 1, otherwise all  $x_i$  will be 1 and relation (7) will not be satisfied. We also note that condition (8) is automatically satisfied.

Under these conditions, it is easily verified that:

$$\min f(x) = f(x^*) = C_1 \cdot (\sqrt{m} \cdot \sqrt{C \cdot N^2 - 2K + m - 1} + K + 1 - m) - K \quad (9)$$

We are only interested in what is happening for large values of  $N$ , because these are the interesting problems for us (we can always take care of the few other cases by direct algorithm testing).

We assumed that  $C_1 > 1$ , and  $1 \leq m \leq K$ , and we also know that

$K < 2 \cdot C \cdot N$ . It can be easily verified that for any value of  $m$  (for any local minimum), the expression  $\min f(x)$  will take the form  $C_2 \cdot \sqrt{C} \cdot N$ , where the constant  $C_2$  is greater than 1.

As a result of the standard KKT calculations, we reach the conclusion that  $P(N)$  is true, and the inductive step is proved. QED.

The MAX2SAT algorithm always finds a maximal truth assignment, not only when the maximal truth assignment satisfies all the clauses

When a literal is flipped in an unsatisfied clause that will be satisfied in the maximal truth assignment, the Hamming distance to a solution decreases by 1 with probability at least  $\frac{1}{2}$ . When a literal is flipped in an unsatisfied clause that will remain unsatisfied in the maximal truth assignment, the Hamming distance indeed increases by 1, but enough clauses will become unsatisfied at this step, with this flip (and these clauses will be satisfied in the maximal truth assignment), such that the deviation will be corrected at later stages of the algorithm. In general, the closer we get to a solution (in terms of the Hamming distance), the more likely it is that the next chosen clause will remain unsatisfied in the maximal truth assignment. That is not always true though, especially if we have many long range variables (variables that appear in many clauses), that do not have the right truth assignment at this moment. A direct proof of the theorem above (without induction) would not be a simple matter.

The general procedure will look like this. We consider a 3SAT problem. We find the equivalent MAX2SAT problem, and we run the MAX2SAT algorithm in this

problem.

## 4 Discussion and conclusions

For general implications, related to efficiently solving NP – complete problems, see [Fortnow, 2013]. An interesting application is related to the problem of automated theorem proving using an efficient algorithm for NP – complete problems (see [A1]). The impact of this type of algorithm in mathematics, cryptography, science in general is hard to estimate.

### Acknowledgements.

I emphasize that Professor Fortnow’s book ([Fortnow, 2013]) was a great source of inspiration and motivation that led me to seriously consider this problem. Also, Professor Uwe Schoning , Professor Lance Fortnow, and Professor C. Papadimitriou corrected some errors that I had in many previous versions of this paper, and for that I am very grateful. I must emphasize though, that the final version of my paper did not receive feedback from these experts in the field, so I do not claim that I have their endorsement. If there are still flaws in my work, that is all my fault.

## Appendices

**A1.** Godel’s letter to John von Neumann. In his letter, Godel writes:

“One can obviously easily construct a Turing machine, which for every formula  $F$  in first order predicate logic and every natural number  $n$ , allows one to decide if there is a proof of  $F$  of length  $n$  (length = number of symbols). Let  $\psi(F, n)$  be the number of steps the machine requires for this and let  $\varphi(n) = \max_F \psi(F, n)$ . The question is how fast  $\varphi(n)$  grows for any optimal machine” (see [Godel, 1956]).

Now we consider this. In [Hopcroft, 1979], we have theorem 13.1, at page 325, where it is proved that for each Turing machine (deterministic or nondeterministic)  $M$  that is time bounded by a polynomial  $p(n)$ , a log-space algorithm exists, that takes as input a string  $x$  and produces a Boolean expression  $E_x$  that is satisfiable if and only if  $M$  accepts  $x$ .

This means that the process of seeking a proof (of reasonable length) of a mathematical statement can be completely automatized. With the algorithm presented in this paper, Godel’s vision can be made reality.

## References

- [Fortnow, 2013]. L. Fortnow, “ *The Golden Ticket, P, NP, and The Search For The Impossible* “, Princeton University Press, 2013.
- [Godel, 1956]. K. Godel, “ *Kurt Godel’s letter to John von Neumann*“, Princeton, 20 March 1956,  
<http://www.cs.cmu.edu/~aada/courses/15251s15/www/notes/godel-letter.pdf>
- [Hopcroft, 1979]. J. E Hopcroft, J. D. Ullman “ *Introduction to Automata Theory, Languages and Computation* “, Addison - Wesley Publishing Company, Inc., 1979.
- [KKT, 2012]. Lagrange Multipliers and the KKT conditions, 2012,  
<http://www.csc.kth.se/utbildning/kth/kurser/DD3364/Lectures/KKT.pdf>
- [Papadimitriou, 1991]. C.H. Papadimitriou, “ *On selecting a satisfying truth assignment* “, in FOCS, pages 163 – 169, 1991.
- [Papadimitriou, 1994]. C.H. Papadimitriou, “ *Computational Complexity* “, Addison - Wesley Publishing Company, Inc., 1994.
- [Schoning, 1991]. Uwe Schoning, “ *A probabilistic Algorithm for k-SAT and Constraint Satisfaction Problems* “, Research Supported by the ESPRIT Basic Research, 1991.

Cristian Dumitrescu,  
119 Young St., Ap. 11,  
Kitchener, Ontario N2H 4Z3,  
Canada.

Email: [cristiand43@gmail.com](mailto:cristiand43@gmail.com)  
[cristiand41@hotmail.com](mailto:cristiand41@hotmail.com)

Tel : (519) 574-7026

**Cristian Dumitrescu.** Biographical notes. I was born in 1964 in Romania. In 1988 I graduated the University of Bucharest, Faculty of Mathematics with a BSc. In Mathematics. I worked as a high school teacher of mathematics and computer programmer for various companies and software houses in Romania, UK and Canada. I have been a Canadian citizen since 1998. In the last 20 years I have been working in a different field of activity, but I have always kept close to my heart mathematics and physics.