

# Approximation Algorithms Lecture 2

## Contents

1	Recap	1
2	Content	1

## 1 Recap

### Definition 1

**3CNF SAT:** We are given a satisfiability formula in 3CNF form (boolean formula with  $\vee, \wedge, \neg$ ) and we are interested in knowing if there is an assignment of T/F values to variables which makes the formula true.

### Definition 2

**3CNF MAX-SAT:** Given a 3CNF formula, find an assignment of T/F values to variables which maximises the number of clauses satisfied.

### Example 1

$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$  is a formula in 3CNF form.

## 2 Content

The 3CNF MAX-SAT problem is NP-hard. We can however get a  $\frac{7}{8}$  approximation using a polynomial-time algorithm.

### Theorem 1

Given a formula  $\phi$  with  $m$  clauses,  $\exists$  a polynomial-time algorithm which finds a boolean assignment satisfying at least  $\frac{7}{8}m$  clauses.

*Proof.* Assign each variable independently a value of T/F with probability  $1/2$ . Consider a clause  $C = (x_i \vee x_j \vee x_k)$ . The probability that the clause is not satisfied is  $(\frac{1}{2})^3 = \frac{1}{8}$ . Then the probability that the clause is satisfied is  $\frac{7}{8}$ . So the expected number of clauses satisfied in a random assignment  $= \frac{7}{8}m$ .

However this is just in expectation. We shall derandomize this algorithm as follows.

Let's draw a binary tree. The number of levels = number of variables, and both children correspond to assigning T/F to the variable. There are potentially an exponential number of nodes in this binary tree (in particular,  $2^n$  leaves). The leaves correspond to unique assignments, each of which has some number of clauses that are satisfied, so write that on the corresponding leaf.

Average of numbers on the leaves is in fact  $\frac{7}{8}m$  (by the expectation result earlier). For a variable, one of the two assignments to that variable has average of leaves  $\geq \frac{7}{8}m$  (corresponds to one of the two subtrees).

This average value can be computed by adding the variable to the expectation and finding the corresponding conditional expectation of the number of satisfied clauses. That is, we get the following algorithm:

1. At the first step, we look at all assignments where  $x_1 = T$  and compute the average number of clauses satisfied in these.
2. Do the same computation for  $x_1 = F$ .

3. At least one of these has  $\geq \frac{7}{8}m$  clauses on average, and set  $x_1$  to the value corresponding to that one.
4. Then do this for the remaining variables as well.

So we go to the direction which always has a  $\geq$  number of satisfied clauses, and we get at least  $\frac{7}{8}m$  clauses that are satisfied.

Since the maximum number of clauses that can be satisfied is  $m$ , so  $OPT \leq m$ , so the number of clauses satisfied by our algorithm is  $\geq \frac{7}{8} \geq \frac{7}{8}OPT$ , whence we get that this algorithm is a  $\frac{7}{8}$ -approximation.  $\square$

**Note 1**

This is called the *method of conditional expectations*, and we choose an assignment which has a higher conditional expectation.