University of Algarve

Faculty of Sciences and technology

Masters in Informatics Engineering

Metaheuristics

Nome: Diogo Fonseca

Número: 79858

Assignment 1

4.

The only instance that satisfies the problem is: (ordered x_1, x_2, …, x_n)

11000

5.

There are 8 instances that solve this problem:

01110001111001101111

10000100000011101001

10000100100001101001

10000100100011101001

10010000010011101001

10010001010011101001

10010100000011101001
10010100010011101001

6.

Given that the naive algorithm goes through all the combinations of variables, and the possible value for each of them is either true or false, it evaluates to checking $\mathbf{2^n}$ different combinations, where **n** is the number of variables. Furthermore, for each combination, **p** clauses are checked, giving us a total of **2^np clause evaluations**. Going further, if all clauses have the same number of variables **v**, then we can conclude **2^npv variable evaluations**.

The uf100-430 has 100 variables, 430 clauses and each clause has exactly 3 variables, which means it should take $\mathbf{c \cdot 2^{100} \cdot 430 \cdot 3}$ amount of time, **c** being the time of one evaluation.

For uf20-91 which has 20 variables and 91 clauses, each one with 3 variables, the problem took about 146ms to solve. So we conclude that $\mathbf{c \cdot 2^{20} \cdot 91 \cdot 3 = 146}$ or $\mathbf{c \cdot 286261248 = 146}$. Rearranging, we find that **c=0.0000005100236271.**

Plugging c into the values from uf100-430, gives us an expected time of **8.340259666E26ms**, or about **2.6E16** years to compute.