

Metaheuristic Optimization

Lab (Zero Marks) - File format for Satisfiability Problems Week 1

The Propositional Satisfiability Problem (SAT) can be represented by a pair $F = (V, C)$ where, V indicates a set of Boolean variables and C a set of clauses in conjunctive normal form (CNF). The following formula shows a SAT example described by means of the CNF.

$$F = (x_1 \vee x_3 \vee -x_4) \wedge (x_4) \wedge (x_2 \vee -x_3)$$

Where \vee represents the **or** Boolean connective, \wedge represents **and**, and $-x_i$ is the negation of x_i . Given a set of clauses, C_1, C_2, \dots, C_m on the variables x_1, x_2, \dots, x_n , the satisfiability problem is to determine if the formula

$$C_1 \wedge C_2 \wedge \dots \wedge C_m$$

is satisfiable. That is, is there an assignment of values to the variables so that the above formula evaluates to *true*?

For instance, a potential solution for F would be $x_1=\text{True}$, $x_2=\text{False}$, $x_3=\text{False}$, $x_4=\text{True}$. x_1 satisfies the first clause, x_4 satisfies the second clause, and $-x_3$ satisfies the last clause.

The preamble. The preamble contains information about the instance. This information is contained in lines. Each line begins with a single character (followed by a space) that determines the type of line. These types are as follows:

- **Comments:** comment line give human-readable information about the file and are ignored by programs. Comment lines appear at the beginning of the preamble. Each line begins with a lower-case character *c*

c This is an example of a comment line

- **Problem line:** there is one problem line per input file. The problem line must appear before any node or arc descriptor lines. For cnf instances, the problem line has the following format.

p FORMAT VARIABLES CLAUSES

The lower-case character p means that this is the problem line. The FORMAL field allows programs to determine the format that will be expected, and should contain the word “cnf”. The VARIABLES field contains an integer value specifying n, the number of variables in the instance. The CLAUSES field contains an integer value specifying m, the number of clauses in the instance. This line must occur as the last line of the preamble.

- **The CLAUSES.** The clauses appear immediately after the problem line. The variables are assumed to be numbered from 1 up to n. It is not necessary that every variable appear in an instance. Each clause will be represented by a sequence of number, each separated by either a space, a tab, or a newline character. The non-negated version of a variable i is represented by i; the negated version is represented by -i.

Each clause is terminated by “0”. Unlike many formats that represent the end of a clause by a new-line symbol, this format allows clauses to be on multiple lines.

Example: $(x_1 \vee x_3 \vee -x_4) \wedge (x_4) \wedge (x_2 \vee -x_3)$

A possible input file would be

c Example CNF format file

c

p cnf 4 3

1 3 -4 0

4 0 2

-3 0

Output or solution files

Every algorithm or heuristic creates an output or solution file. This output file should consists of one or more of the following lines, depending on the type of algorithm and problem being solved.

- Comment. Comment line give human-readable information about the file and are ignored by programs. Comment lines can appear anywhere in the file. Each comment line begins with a lower-case character *c*. Note that comment lines can be used to provide solution information not otherwise available (i.e., computation time, number of calculations).

c This is an example of a comment line

- Variable Line

v *V*

The lower-case character *v* means that this is a variable line. The value *V* is either a positive value *i*, which means that *i* should be set to true or a negative value *-i*, implying it should be set to false.

Lab Work.

Write a program that reads two files. The first one contains a SAT instance *I* (.cnf format) and a solution for *I* and indicates whether the provided solution is a valid solution for *I* or not.

You need to run your code using the command:

Python SAT <cnf instance> <solution file>

For instance: “python SAT instance.cnf solution.txt” means the problem will use “instance.cnf” as a SAT instance and solution.txt as an solution for this instance.

A test dataset (data.zip) is available on Canvas. Inst includes 10 sat instances uf20-n.cnf and sol includes a set of potential solutions.