Sample problem

The Boolean satisfiability problem (SAT) is the problem of determining whether there exists an assignment of Boolean variables that satisfies a given formula. For example, the formula $(\neg a \vee b) \wedge \neg b$ is satisfiable by assigning FALSE to a and b, whereas the formula $(c \vee d) \wedge \neg c \wedge \neg d$ cannot be satisfied, no matter the truth value of its variables.

Here we assume that the formula is given in conjunctive normal form (CNF): a formula is a conjunction of clauses, and a clause is a disjunction of literals (a Boolean variable or its negation). The example formulas above are both in CNF.

1. Show how to solve SAT for any formula using an ILP.

Solution:

For every Boolean a variable occurring in the formula, we introduce one integer variable x_a. The variables should be interpreted as TRUE if their value is 1, and as FALSE if their value is 0. Therefore we have the constraints $0 \leq x_a$ and $x_a \leq 1$.

For each clause in the formula, we introduce one constraint in the ILP. Let P be the set of Boolean variables occurring positively in the clause and let N be the set of variables occurring negatively. We add the constraint:

$$\sum_{p \in P} x_p + \sum_{n \in N} (1 - x_n) \geq 1$$

This linear constraint is satisfied if and only if at least one of the positively occurring variables is set to 1, or at least one the negatively occurring variables is set to 0. The corresponding Boolean assignment satisfies the clause if and only if the linear constraint is satisfied.

If there exists a solution to the ILP that satisfies all the constraints, the corresponding Boolean assignment satisfies the conjunctions of clause, i.e. the formula

SAT is a decision problem (with a yes or no answer) rather than an optimization problem, so there is no natural objective function. We can simply pick a constant function.

For example for the formula:

$$(a \vee \neg b \vee \neg c \vee d) \wedge (\neg a \vee d) \wedge (b \vee e)$$

$$
\begin{aligned}
\max \quad & 0 \\
\text{s.t.} \quad & x_a + (1 - x_b) + (1 - x_c) + x_d \geq 1 \\
& (1 - x_a) + x_d \geq 1 \\
& x_b + x_e \geq 1 \\
& \vec{x} \geq \vec{0} \\
& \vec{x} \leq \vec{1}
\end{aligned}
$$

2. MAX-SAT is an extension of the SAT problem. Given a CNF formula, the problem is to compute the maximum number of clauses that can be satisfied by any assignment (in SAT, we only check whether all clauses can be satisfied). Modify your ILP to solve the MAX-SAT problem.

Solution: In addition to the variables described above, we also introduce one variable x_C for each clause C in the formula, and the constraints $0 \leq x_C$ and $x_C \leq 1$.

The constraint associated to the clause C is modified to read:

$$\sum_{p \in P} x_p + \sum_{n \in N} (1 - x_n) \geq x_C$$

The variable x_C is set to 1 in a feasible solution if and only if the clause C is satisfied by the Boolean assignment of N and P.

The objective function maximizes the number of clauses satisfied in the formula $\mathcal{F}$:

$$\max \sum_{C \in \mathcal{F}} x_C$$

The optimal value of the ILP is the maximum number of clauses of $\mathcal{F}$ that can be satisfied by any assignment.