Non-Interlaced SAT is in \( P \).

Dr Serge Burckel, S.I.G.L.E. (www.sigle.space).

Abstract. We investigate the NP-Complete problem SAT and the geometry of its instances. For a particular type that we call non-interlaced formulas, we propose a polynomial time algorithm for their resolution using graphs and matrices.

1. Introduction.

We investigate the decision problem SAT. Some reductions have been done in order to preserve the property to be NP-Complete ([1], [2]). Here, we propose a geometrical property that depends on the order of the clauses and that can lead to a polynomial time method. We consider formulas represented by finite lists \( F = [C_1, C_2, \ldots, C_k] \) where each clause \( C_i \) is a finite list \((x_1^i, \ldots, x_{n_i}^i)\) where \( n_i > 0 \) and each \( x_a^i \in Z^* \). This is an easy way to encode general CNF instances. For example \((a \lor b \lor \neg c) \land \neg(a \lor \neg b)\) will be encoded by \([(1, 2, -3), (-1, -2)]\). This is even more general because we allow clauses like \((1, 2, 1, -2, -1)\). The SAT problem is to decide whether there exists a good choice \([c_1, \ldots, c_k]\) where each \( c_i \in C_i \) and \( c_i \neq -c_j \) for every \( i, j \).

We will count the total number \( \Gamma(F) \) of these good choices since the initial formula \( F \) is satisfiable if and only if \( \Gamma(F) > 0 \). This will be possible in polynomial time for a certain type of instances.

2. Non-Interlaced Formulas.

Definition. Let \( F \) be a formula.

Let \( \Delta(F) \) be the set of pairs \([i, j]\) where \( i < j \) and \( \exists x_a^i, x_b^j \) with \( x_a^i = -x_b^j \). We say that \( F \) is interlaced if \( \exists \) \([i, j], [i', j']\) \( \in \Delta(F) \) with \( i < i' < j < j' \). Otherwise, \( F \) is non-interlaced.

For example \( F = [(1), (2), (-1), (-2)] \) is interlaced with \( \Delta(F) = \{(1, 3), [2, 4]\} \) and the permuted formula \( F' = [(1), (2), (-2), (-1)] \) is non-interlaced with \( \Delta(F') = \{(1, 4), [2, 3]\} \).

We call Non-Interlaced SAT the restriction of the SAT problem to non-interlaced formulas.

Theorem 1. Non-Interlaced SAT is in \( P \).

The proof consists of building an effective polynomial time algorithm that counts the number \( \Gamma(F) \) of good choices with a graph \( G \) and an associated matrix \( M \).

3. The Graph and the Matrix.

The graph \( G \) has \( N = 2 + n_1 + \ldots + n_k \) vertices: two special vertices \( s, t \) and a vertex \( v_a^i \) for each \( x_a^i \). The edges of \( G \) are directed and valued. Moreover, they are of different types in order to distinguish them. We do not indicate this type when it is not necessary. However, notice that two vertices can be linked with edges of different types. We begin to put the edges of type \( \theta \) in \( G \):

\[
\begin{align*}
s & \rightarrow v_a^1 \\
v_a^i & \rightarrow v_j^i \text{ for every } 1 \leq i < k \text{ and } j = i + 1 \\
v_a^k & \rightarrow t \text{ for every } x_a^k \in C_k \\
s & \rightarrow t \text{ when } k = 0
\end{align*}
\]

Observe that \( G \) has no cycle. This property will be preserved in the sequel.

In practice, we will not have to build \( G \). We will build simultaneously its adjacency matrix \( M \) of dimension \( N \times N \) indexed by the vertices of \( G \) and such that \( M[x, y] \) is the sum of the values of the edges \( x \rightarrow y \).
Definition. For two vertices \( x, y \), the number \( \pi(x, y) \) is the value of all the paths from \( x \) to \( y \) which is equal to the sum over all these paths of the products of the values of their edges.

In order to compute \( \pi(x, y) \), we perform the following operations on the matrix \( M \) since the longest path will always have at most \( k + 1 \) edges:

Fix \( M[y, y] := 1 \). Compute \( M' := M^{k+1} \). Fix \( M[y, y] := 0 \). Return the entry \( M'[x, y] \).

Observe that fixing \( M[y, y] := 1 \) enables us to take account of all the paths whatever their lengths. Observe that \( k + 1 < N \). Hence, this computation can obviously be performed in \( O(N^4) \) steps.

4. New edges.

We expect to have \( \pi(s, t) = \Gamma(F) \). We are going to “remove” step by step the contributions of pairs \([v^i_a, v^j_b]\) where \( x^i_a = -x^j_b \). We consider these pairs in the increasing order of \((j - i)\).

For \( \delta \) from 1 to \( k - 1 \) do

- For every pair \([i, j] \in \Delta(F)\) with \( j = i + \delta \)
  - For every pair \([v^i_a, v^j_b]\) with \( x^i_a = -x^j_b \)
    - Compute the number \( \alpha := \pi(v^i_a, v^j_b) \).
    - Add an edge \( v^i_a \rightarrow v^j_b \) of type 1.

Observe for example, that an edge of type 0 like \( v^i_a \rightarrow v^j_b \) with \( x^i_a = -x^j_b \) leads to the creation of an edge of type 1: \( v^i_a \rightarrow v^j_b \). Hence \( M[v^i_a, v^j_b] = 0 \). We have the complete method:

1. Given a non-interlaced formula \( F \), compute the set \( \Delta(F) \).
2. Build the matrix \( M \) according to edges of type 0.
3. Add the edges of type 1 in \( M \).
4. Conclude that \( F \) is satisfiable if and only if \( \pi(s, t) > 0 \).

We are able to prove Theorem 1.

Proof. We show that the previous method gives in polynomial time a correct answer.

- Complexity. The number \( N \) is smaller than the size of the instance \( F \). Since the number of pairs \([v^i_a, v^j_b]\) is bounded by \( N^2 \), the algorithm is polynomial in \( \text{DTIME}(N^6) \).

The computed numbers like \( \Gamma(F) \) are all bounded by the product \( n_1 \cdot n_2 \cdot \ldots \cdot n_k \) and representable in space \( \log(n_1) + \log(n_2) + \ldots + \log(n_k) < N \).

- Correctness. An edge \( e \) of type 1 like \( v^i_a \rightarrow v^j_b \) will set to zero the values of all the paths containing these two vertices. By induction on \( \delta = j - i \) that is sufficient to be correct. For \( \delta = 1 \), that is obvious and we have already made this observation. For \( \delta > 1 \), the possible edges of type 1 behind this edge \( e \) have made (by induction hypothesis) the good corrections for \( \pi(v^i_a, v^j_b) \) that will be the number of remaining good paths between \( v^i_a \) and \( v^j_b \). The edge \( e \) removes them with a value \(-\alpha\). For the others edges \( f \) of type 1 with values \(-\beta\) that are before of after \( e \), the inclusion-exclusion principle is automatically applied with the structure of the graph. The paths from \( s \) to \( t \) have 4 possibilities:
  - not use \( e \) and not use \( f \): the non corrected paths,
  - use \( e \) and not use \( f \): minus the ones corrected by \( e : -\alpha \),
  - not use \( e \) and use \( f \): minus the ones corrected by \( f : -\beta \),
  - use \( e \) and use \( f \): plus the ones corrected by \( e \) and by \( f \) (that where removed twice): \((-\alpha)(-\beta) = \alpha \beta \).

The same inclusion-exclusion principle applies for any number of successive edges of type 1.
5. Conclusion.

When the formula $F$ is non-interlaced, we always have $\pi(s, t) = \Gamma(F)$.

However, there exist interlaced formulas for which this is not the case.

For example, the computation for $F = [(1), (2), (-1), (-2)]$ will give $\pi(s, t) = -1$ whereas $\Gamma(F) = 0$.

We hope that this paper will bring some interests in the investigation of the geometry of formulas and lead to future improvements. One could adapt the presented method for interlaced formulas or find a tricky polynomial time transformation $\Phi$ such that an interlaced formula $F$ becomes a non-interlaced formula $\Phi(F)$ that is satisfiable if and only if $F$ is. That would imply $P = NP$.

[1] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, New York (1979).
[2] C.A. Tovey, A simplified satisfiability problem, Discrete Appl. Math., 8 (1984), pp. 85-89.