NP-completeness: Some Reductions

1. **SAT to 3SAT**

   SAT: Given a collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses, where each clause consists of a set of literals (representing the disjunction of those literals) over the finite set of Boolean variables $U = \{u_1, u_2, \ldots, u_n\}$, is there an assignment of truth values to U which makes every clause true ?
   Example: For the collection $c_1 = \{\overline{u_1}, u_2, u_4\} (= \overline{u_1} \vee u_2 \vee u_4), c_2 = \{\overline{u_2}, u_3\}, c_3 = \{u_1, \overline{u_2}, u_3, \overline{u_4}\}$, over the variables $\{u_1, u_2, u_3, u_4\}$ the answer is yes. (One assignment is u_1 false, u_2 false, u_3 false (irrelevant),u_4 false (irrelevant)).

   3-SAT: A version of the SAT problem in which every clause has 3 literals.

   We'll prove that 3-SAT is $\mathcal{NP}$-complete. Firstly, note that 3-SAT $\in \mathcal{NP}$ since a non-deterministic algorithm need only guess an assignment of values to U and check if it works in polynomial time ($\leq 3\times$ no. of clauses).

   We'll next find a polynomial reduction from SAT to 3-SAT.
   Let the given instance of SAT contain the collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses over the variables $U = \{u_1, u_2, \ldots, u_n\}$. We construct a collection C' of 3 literal clauses over the variables U' which consist of the original variables plus sets of additional variables as follows:

   Replace each clause $c_i \in C$ by a collection of 3 literal clauses over the variables which appear in c_i plus some additional variables which appear only in these 3 literal clauses. The exact procedure depends on the number of literals in c_i. Let $c_i \in C$ be given by $\{z_1, z_2, \ldots, z_k\}$ where the z_j's are literals over U. If

   $k = 1$ $c_i = \{z_1\}$. Use two additional variables $\{y_{i,1}, y_{i,2}\}$. Form the collection $C'_i = \{\{z_1, y_{i,1}, y_{i,2}\}, \{z_1, y_{i,1}, \overline{y_{i,2}}\}, \{z_1, \overline{y_{i,1}}, y_{i,2}\}, \{z_1, \overline{y_{i,1}}, \overline{y_{i,2}}\}\}$

   $k = 2$ $c_i = \{z_1, z_2\}$. Use one additional variable $\{y_{i,1}\}$.
   Form the collection $C'_i = \{\{z_1, z_2, y_{i,1}\}, \{z_1, z_2, \overline{y_{i,1}}\}\}$

   $k = 3$ $c_i = \{z_1, z_2, z_3\}$. No additional variables. $C'_i = c_i$.

   $k > 3$ Use the additional variables $\{y_{i,1}, y_{i,2}, \ldots, y_{i,k-3}\}$. Form the collection $C'_i = \{\{z_1, z_2, y_{i,1}\},$
   $\{\overline{y_{i,1}}, z_3, y_{i,2}\}, \{\overline{y_{i,2}}, z_4, y_{i,3}\}, \{\overline{y_{i,3}}, z_5, y_{i,4}\}, \ldots, \{\overline{y_{i,k-3}}, z_{k-1}, z_k\}\}$

   Example (cont'd): The transformation results in the collection of 3 literal clauses $C'_1 = \{\overline{u_1}, u_2, u_4\}, C'_2 = \{\{\overline{u_2}, u_3, y_{2,1}\}, \{\overline{u_2}, u_3, \overline{y_{2,1}}\}\}, C'_3 = \{\{u_1, \overline{u_2}, y_{3,1}\}, \{\overline{y_{3,1}}, u_3, \overline{u_4}\}\}$ over the variables $\{u_1, u_2, u_3, u_4, y_{2,1}, y_{3,1}\}$.

   To show that this procedure forms a polynomial reduction, firstly we observe that the number of 3 literal clauses in C' is bounded by a polynomial in nm which leads to the procedure being a polynomial time one. Secondly, whenever C is satisfiable, then the C' clauses formed in the cases $k \leq 3$ are automatically satisfied (for arbitrary assignment of any additional variables) while in the case $k > 3$

   (a) If z_1 or z_2 is true, assign all additional variables the truth value "false". In this case, the first literal in each relevant clause is true.

(b) If z_{k-1} or z_k is true, assign all additional variables the truth value "true". In this case, the third literal in each relevant clause is true.

(c) Otherwise, if z_l is true, assign $y_{i,j}$ the value "true" when $1 \leq j \leq l-2$ and the value "false" when $l-1 \leq j \leq k-3$. In this case, the third literal in each relevant clause preceding the one which includes z_l is true while the first literal in each relevant clause succeeding the one which includes z_l is true.

Thus all the clauses in C' are satisfied. Conversely, if all the clauses in C' are satisfied by an truth assignment to U' (which includes the original variables), then C is satisfied by the same truth assignment restricted to U.
Thus we conclude that C' is satisfied if and only if C is satisfied.

Example(cont'd): The collection of 3-literal clauses $\{\overline{u_1}, u_2, u_4\}$, $\{\overline{u_2}, u_3, y_{2,1}\}$, $\{\overline{u_2}, u_3, \overline{y_{2,1}}\}$, $\{u_1, \overline{u_2}, y_{3,1}\}$, $\{\overline{y_{3,1}}, u_3, \overline{u_4}\}$ is satisfied by u_1 false, u_2 false, u_3 false,u_4 false, $y_{2,1}$ false (irrelevant) and $y_{3,1}$ false (irrelevant).

2. **Graph Colouring**

Graph 3-colouring Problem: Given a graph $G = (V, E)$ is there a way to colour the vertices of G with 3 colours? A valid colouring of the graph requires all adjacent nodes to have different colours.
Graph 4-colouring Problem: Given a graph $G' = (V', E')$ is there a way to colour the vertices of G' with 4 colours? Again adjacent nodes must have different colours.

Since it is easy to verify a proposed 3-colouring in polynomial time, the problem is in NP. There is a reduction that reduces 3SAT to Graph 3-Colouring; thus showing that Graph 3-Colouring is NP-Complete.

We now show how to reduce 3-Colouring to 4-Colouring. Given a graph G, we create a new graph G' as follows. Add a new vertex v to G and add an edge from v to each vertex of V. $V' = V \cup \{v\}$. $E' = E \cup \{(v, u); u \in V\}$.

To prove the correctness of the reduction observe that if G is 3 colourable then G' is 4 colourable (use the new colour for v). If G' is 4 colourable, then assuming that v uses one colour, the remaining vertices of G' use only 3 colours and this shows that G is 3 colourable.

3. **HC to TSP**