

Hamiltonian Graphs

Recall the definition of **HAM**—the language of Hamiltonian graphs.

Given a graph $G = (V, E)$, a *Hamiltonian cycle* in G is a path in the graph, starting and ending at the same node, such that every node in V appears on the cycle *exactly once*.

A graph is called *Hamiltonian* if it contains a Hamiltonian cycle.

The language **HAM** is the set of encodings of Hamiltonian graphs.

Travelling Salesman

Recall the travelling salesman problem

Given

- V — a set of nodes.
- $c : V \times V \rightarrow \mathbb{N}$ — a cost matrix.

Find an ordering v_1, \dots, v_n of V for which the total cost:

$$c(v_n, v_1) + \sum_{i=1}^{n-1} c(v_i, v_{i+1})$$

is the smallest possible.

Hamiltonian Cycle

We can construct a reduction from **3SAT** to **HAM**

Essentially, this involves coding up a Boolean expression as a graph, so that every satisfying truth assignment to the expression corresponds to a Hamiltonian circuit of the graph.

This reduction is much more intricate than the one for **IND**.

Travelling Salesman

As with other optimisation problems, we can make a decision problem version of the Travelling Salesman problem.

The problem **TSP** consists of the set of triples

$$(V, c : V \times V \rightarrow \mathbb{N}, t)$$

such that there is a tour of the set of vertices V , which under the cost matrix c , has cost t or less.

Reduction

There is a simple reduction from **HAM** to **TSP**, mapping a graph (V, E) to the triple $(V, c : V \times V \rightarrow \mathbb{N}, n)$, where

$$c(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 2 & \text{otherwise} \end{cases}$$

and n is the size of V .

3D Matching

The decision problem of **3D Matching** is defined as:

Given three disjoint sets X, Y and Z , and a set of triples $M \subseteq X \times Y \times Z$, does M contain a matching?

I.e. is there a subset $M' \subseteq M$, such that each element of X, Y and Z appears in exactly one triple of M' ?

We can show that **3DM** is **NP**-complete by a reduction from **3SAT**.

Sets, Numbers and Scheduling

It is not just problems about formulas and graphs that turn out to be **NP**-complete.

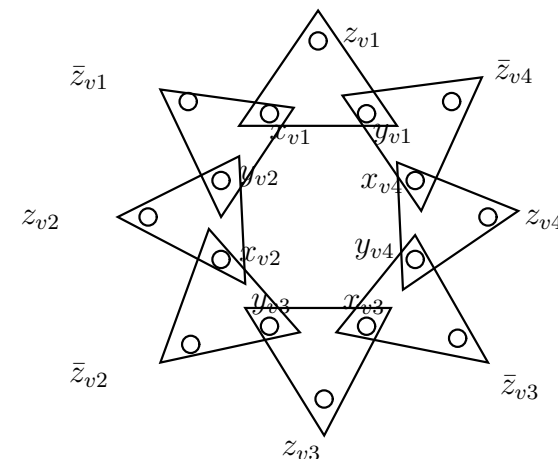
Literally hundreds of naturally arising problems have been proved **NP**-complete, in areas involving network design, scheduling, optimisation, data storage and retrieval, artificial intelligence and many others.

Such problems arise naturally whenever we have to construct a solution within constraints, and the most effective way appears to be an exhaustive search of an exponential solution space.

We now examine three more **NP**-complete problems, whose significance lies in that they have been used to prove a large number of other problems **NP**-complete, through reductions.

Reduction

If a Boolean expression ϕ in **3CNF** has n variables, and m clauses, we construct for each variable v the following gadget.



In addition, for every clause c , we have two elements x_c and y_c .

If the literal v occurs in c , we include the triple

$$(x_c, y_c, z_{vc})$$

in M .

Similarly, if $\neg v$ occurs in c , we include the triple

$$(x_c, y_c, \bar{z}_{vc})$$

in M .

Finally, we include extra dummy elements in X and Y to make the numbers match up.

Set Covering

More generally, we have the *Set Covering* problem:

Given a set U , a collection of $S = \{S_1, \dots, S_m\}$ subsets of U and an integer budget B , is there a collection of B sets in S whose union is U ?

Exact Set Covering

Two other well known problems are proved NP-complete by immediate reduction from 3DM.

Exact Cover by 3-Sets is defined by:

Given a set U with $3n$ elements, and a collection $S = \{S_1, \dots, S_m\}$ of three-element subsets of U , is there a sub collection containing exactly n of these sets whose union is all of U ?

The reduction from 3DM simply takes $U = X \cup Y \cup Z$, and S to be the collection of three-element subsets resulting from M .