1 Claim: ILP is NP-complete

Note: ILP ∈ NP because we can check if a sol satisfies all constraints and if obj ≥ K

Sol 1: 3SAT ≤p ILP.

3SAT instance has variables $x_1, \dots, x_n$

clauses $C_1, \dots, C_m$ (all length 3)

Reduce to ILP by solving

$$\max \sum_{j=1}^{k} b_j$$

$$\text{s.t.} \sum_{\substack{i \text{ s.t.} \\ x_i \in C_j}} a_i + \sum_{\substack{i \text{ s.t.} \\ \bar{x}_i \in C_j}} [1 - a_i] \geq b_j \qquad \forall\, j \in \{1, \dots, k\}$$

(each clause is satisfied)

$$a_i \in \{0,1\} \quad \forall\, i \in \{1, \dots, n\}$$

(each var x_i is set to $a_i = 1$ or $a_i = 0$)

$$b_j \in \{0,1\} \quad \forall\, j \in \{1, \dots, k\}$$

(each clause C_j is satisfied if $b_j = 1$)

If the objective is $\geq k$, then we have a sol to 3SAT.

Sol 2: 3SAT ≤p ILP. (alternate reduction)

Reduce to ILP by solving

$$\max \quad D$$

$$\text{s.t.} \sum_{\substack{i \text{ s.t.} \\ x_i \in C_j}} a_i + \sum_{\substack{i \text{ s.t.} \\ \bar{x}_i \in C_j}} [1 - a_i] \geq 1 \qquad \forall\, j \in \{1, \dots, k\}$$

(all clauses satisfied)

$$a_i \in \{0,1\} \quad \forall\, i \in \{1, \dots, n\}.$$

If this ILP has ANY feasible solution, then the 3SAT instance is satisfiable

Sol 3: Set Cover ≤p ILP.

Set cover instance has objects $\{a_1, \dots, a_n\}$ and $\{S_1, \dots, S_m\}$, number of sets to cover k

We will reduce set cover to ILP:

$$\min \sum_{i=1}^{n} x_i$$

(want as few x_i's = 1 if we choose set S_i)

$$\text{s.t.} \sum_{i=1}^{n} x_i \underbrace{\mathbb{1}_{[a_i \in S_j]}}_{\text{indicator var}} \geq 1 \qquad \forall\, j \in \{1, \dots, m\}$$

(want each a_j to be covered by some set S_i)

$$\mathbb{1}_{[a_i \in S_j]} = \begin{cases} 1 & \text{if } a_i \in S_j \\ 0 & \text{else} \end{cases}$$

$$x_i \in \{0,1\} \quad \forall\, i \in \{1, \dots, n\}$$

If this ILP has a sol $\leq k$, then the Set Cover problem can be solved with $\leq k$ sets.

Many other reductions possible.

2 Claim: Double Hamiltonian Cycle is NP-complete.

Sol: First notice Double-Hamiltonian ∈ NP since we can just check a solution by following it & making sure it visits every vertex twice.

Reduction: HamCycle ≤p Double HamCycle

Start w/ HamCycle instance $G = (V, E)$.

To get DoubleHamCycle instance, start by copying G to new graph G'.

Then $\forall\, v \in V$, create two new nodes in G'

v_1, v_2, and edges $v \to v_1$, $v_1 \to v_2$, $v_2 \to v$



(rest of network)

Correctness: Assume there is a Hamiltonian cycle C in G. Then we have a double Hamiltonian cycle in G' by augmenting C: when you visit v in C, etc. visit v_1, then v_2, then v, then the next in C. In this new path C', we then also easily note twice in G'.

If we have a double HamCycle in G', we can "short" the cycle anywhere, to start on a node $v \in V$.

Clearly, the two visits to v_1 & v_2 have to happen after the first visit to v but before the next visit to v, (or else you get stuck), so we can just remove these visits to get a single-Hamiltonian cycle in G

3 Strongly Ind Set is NP-complete.

Sol

Ind Set ≤p Strongly Ind Set.

Reduction: Start with IndSet instance $\{G = (V, E), k\}$

Create graph $G' = (V', E')$, relating by copies G

$\forall\, e = (v, y) \in E$, replace with vertex $v_e \neq$ edge (v, v_e), (v_e, y).



Also $\forall\, e_1, e_2 \in E$, add an edge $(v_{e_1}, v_{e_2}) \in E'$.

Take new defn our instance of Strongly Ind Set (G', k)

Correctness:

Notice, $\forall\, v, w \in V$, $d_{G'}(v, w) = \begin{cases} 2 & \text{if } d_G(v, w) = 1 \\ \geq 3 & \text{else.} \end{cases}$

Hence, $U \subseteq V$ is an IndSet in G iff U is a Strongly Ind Set in G'.

On the other hand, if U is a Strongly Ind Set, then either $U \subseteq V_E$ for some $e \in E$ or it contains only $v \in V$. (since we're distance ≥ 3 away from every node in V)

But then U is also an Ind Set in G.

Thus, $\forall\, k \geq 2$, if we can solve Strongly Ind Set, we can solve Ind Set. (and $k = 1$ is trivial)

Also, Strongly Ind Set ∈ NP since we can easily check a potential cert in poly time.

Thus, Strongly Ind Set is NP-complete.

4 Subgraph-Isomorphism is NP-complete

Sol The problem is clearly in NP because given a function f, we can check the mapping

We now show Ind Set ≤p Subgraph Isomorphism

For any IndSet instance $\{G = (V, E), k\}$, create a SubgraphIsomorphism instance with $G_1 = \{V_1, \emptyset\}$, and $G_2 = G$,

where $|V_1| = k$.

In other words, G_1 is a disconnected graph with no edges but k nodes.

G_1 is a subgraph of G_2 iff G has an independent set of size k.

Thus, Ind Set ≤p Subgraph Isomorphism

$\Rightarrow$ Subgraph Isomorphism is NP-complete.

5 Required-Path-Subgraph is NP-complete.

1. NP: If is easy to check such a subgraph if given, so is NP.

Sol 1: Hamiltonian Cycle ≤p Required Path Subgraph

Let $G = (V, E)$ be an instance of HamCycle.

For Required Path Subgraph, use same graph $G = (V, E)$ and cost $R_{ij} = 2 \quad \forall\, i \neq j$ and $R_{ii} = 0$, and $k = |V|$.

If G has a HamCycle C, set $S = \{e_{ij} = C\}_{ij}$, i.e. $|S| = |V| = k$, and then we at least the disjoint graphs along the cycle & opposite direction for each pair $i \neq j$.

Conversely, if $\exists\, S$ with $|S| \leq k$ edges that satisfy $R_{ij} = 2\ \forall\, i \neq j$, then the path from i to j from a cycle C containing $i \neq j$ with no repetition of nodes

Let C be the longest such cycle in S, and let $i \neq j$ be subsequent along C.

Suppose $\exists\, k \notin C$. (i.e. $k \neq i, k \neq j$, in short)

Clearly, there must be a node $C_{jk} \neq C$ that includes nodes j & k.

If $|C \cap C_{jk}| > 1$, then we could expand C without repetition of nodes by merging C and C_{jk}, so $C \cap C_{jk} = \{j\}$.

But this must also be true for C_{ik} if $C_{ik} \neq C$, $C \cap C_{ik} = \{i\}$.

But then we could expand C by going from $i \to k \to j \to$ rest of C.

Thus, $|C| = |V|$.

So there is exactly one cycle containing all nodes without repetition of nodes which is exactly a HamCycle.

Thus, Required Path Subgraph is NP-hard.

$\Rightarrow$ It is NP-complete.

Incorrect Sol: Trying to directly show HamPath ≤p ReqPath Subgraph by setting $R_{ij} = 1$ and $k = |V| - 1$, because that just shows a tree, and not a HamPath.