Example of NP-Hard and NP-Complete problem

Hamiltonian Cycle Problem

- A directed Hamiltonian cycle in a directed graph $G = (V, E)$ is a directed cycle of length $n = |V|$. So, the cycle goes through every vertex (node) exactly once and then returns back to the starting vertex.

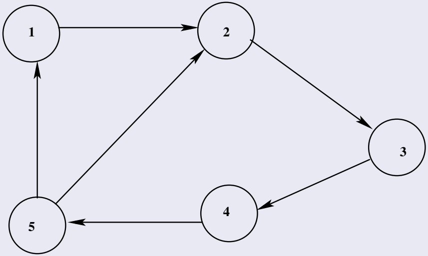


Figure: A directed graph G with a Hamiltonian cycle 1, 2, 3, 4, 5, 1.

Hamiltonian Cycle Problem

- Consider the following problem formally as *HAMCYCLE* $:= \{\langle G \rangle |$ there is a Hamiltonian cycle in the directed graph $G\}$.

Theorem

HAMPATH is NP-complete.

Hamiltonian Cycle Problem

- Consider the following problem formally as $HAMCYCLE := \{\langle G\rangle|$ there is a Hamiltonian cycle in the directed graph $G\}$.

Theorem

Using the fact that if HAMPATH is NP-complete, HAMCYCLE is also NP-complete.

Proof: To show HAMCYCLE is NP-complete, we must demonstrate two things:
(1) that HAMCYCLE is in NP; and
(2) that every language $A \in NP$ is poly-time reducible to HAMCYCLE.
That is, to prove HAMCYCLE is NP-hard, we take a poly-time reduction HAMPATH $\leq_p$ HAMCYCLE.

Hamiltonian Cycle Problem

Part 1. *HAMCYCLE* $\in$ *NP*

- We need to construct a poly-time NTM for HAMCYCLE.
- **NTM for HAMCYCLE:**
  Input: $\langle G \rangle$.
  Output: Accept, if there is a Hamiltonian cycle; reject, otherwise.

Stages:

1. Non-deterministically generate a sequence of $n+1$ nodes, where $n = |V|$ is the number of nodes in G, say $p_1, p_2, p_3, \ldots, p_n, p_{n+1}$ from the set $\{1, 2, 3, \ldots, n\}$.
2. If $p_1 \neq p_{n+1}$, then "reject".
3. If there is a repetition in $p_1, p_2, \ldots, p_n, p_{n+1}$, then "reject".

Hamiltonian Cycle Problem

Part 1. $HAMCYCLE \in NP$ (Continued...)

4. If for some $i = 1, 2, \ldots, n-1$, the edge (p_i, p_{i+1}) is not an edge of G, then "reject".
5. If (p_n, p_{n+1}) is not an edge of G, then "reject".
6. "Accept".

Obviously, HAMCYCLE runs in poly-time by the NTM. Hence, $HAMCYCLE \in NP$.

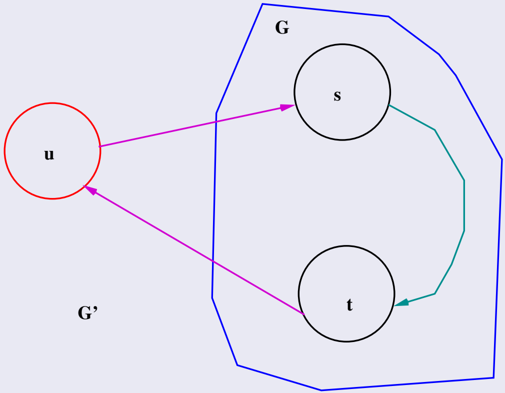
Hamiltonian Cycle Problem

Part 2. $HAMPATH \leq_p HAMCYCLE$

- We have, $HAMPATH := \{\langle G, s, t\rangle |$ there is a (directed) Hamiltonian path from node s to node t in the directed graph G $\}$.
- Let $G = (V, E)$ be a directed graph with two vertices s and t.
- We plan to convert $\langle G, s, t\rangle$ to another directed graph $G' = (V', E')$ such that G has an (s, t)-Hamiltonian path if and only if G' has a Hamiltonian cycle.
- **Construction of** $G' = (V', E')$:
  1. $V' := V \cup \{u\}$.
  2. $E' := E \cup \{(u, s), (t, u)\}$.

Hamiltonian Cycle Problem

Part 2. *HAMPATH* $\leq_p$ *HAMCYCLE*
Construction of $G' = (V', E')$ (Continued...):



Hamiltonian Cycle Problem

Part 2. *HAMPATH* $\leq_p$ *HAMCYCLE*
Construction of $G' = (V', E')$ (Continued...):

- Suppose that G has an s, t-Hamiltonian path $s, u_1, u_2, \ldots, u_m, t$. Then, $u, s, u_1, u_2, \ldots, u_m, t, u$ becomes a Hamiltonian cycle in G'.
- Conversely, let G' have a Hamiltonian cycle. If we traverse around the cycle starting from u, we must first reach s after leaving u. In order to complete the cycle we must take the edge (t, u). Between s and t the cycle visits every other node of G exactly once, that is, this cycle must be of the form $u, s, v_1, v_2, \ldots, v_m, t, u$. But then $s, v_1, v_2, \ldots, v_m, t$ is an s, t-Hamiltonian path in G.
- Clearly, this reduction runs in poly-time.

P, NP, NP-Hard and NP-Completeness

The Traveling Salesperson Problem

- Assume that a salesperson wishes to travel (visit) m cities with each city visited exactly once.
- Associated with each pair of cities a (positive) cost representing the overhead for inter-city travel (assumed symmetric with respect to the two cities).
- The objective of the salesperson is to reduce the total cost for the travel.
- Consider an undirected graph (complete) on m vertices each vertex representing a city and with each edge labeled by the cost of the corresponding inter-city travel.
- The traveling salesperson problem can be reformulated as finding an (undirected) Hamiltonian cycle in the graph with the minimum sum of labels on the edges of the cycle.

P, NP, NP-Hard and NP-Completeness

The Traveling Salesperson Problem (Continued...)

- Consider the following decision problem:
  $TSP := \{\langle G, k\rangle | G$ has a Hamiltonian cycle of (total) cost $\leq k\}$.

Theorem

TSP is NP-Complete.

Proof. To show that TSP is NP-Complete, we must demonstrate two things:

1. that TSP is in *NP*, and
2. that TSP is NP-hard, that is, $A \leq_p TSP$, for all $A \in NP$. In this case, we take a known NP-Complete problem, known as UHAMCYCLE as follows:
   $UHAMCYCLE := \{\langle G\rangle|$ there is a Hamiltonian cycle in the undirected graph G $\}$. Thus, we need to do the poly-time reduction as $UHAMCYCLE \leq_p TSP$.

The Traveling Salesperson Problem (Continued...)

Part 1: TSP is in *NP*

NTM (algorithm) for TSP:

- Let $m = |V|$ be the number of vertices in an undirected graph $G = (V, E)$ and $c(u, v)$ represent the cost of the edge $(u, v) \in E$.
- Input: $\langle G, k \rangle$
- Output: Accept/Reject.

Stages:

1. Let $cost := 0$ (initially).
2. Non-deterministically generate a sequence of $m + 1$ nodes (cities), say $C_1, c_2, \dots, C_m, C_{m+1}$; each C_i is chosen from the set $\{1, 2, \dots, m\}$.
3. If $C_1 \neq C_{m+1}$, then "reject".

The Traveling Salesperson Problem (Continued...)

4. If there is a repetition in the path: $C_1, C_2, \ldots, C_m$, then "reject".
5. If, for $i = 1, 2, \ldots, m$, the edge (C_i, C_{i+1}) is not in E(G), then "reject". Otherwise, update $cost := cost + c(C_i, C_{i+1})$.
6. If $cost \leq k$, then "accept"; otherwise, "reject".

Obviously, this algorithm decides TSP in ploy-time. Hence, $TSP \in NP$.

The Traveling Salesperson Problem (Continued...)

Part 2: $UHAMCYCLE \leq_p TSP$

Construction of G' for TSP:

- Let G be an instance for UHAMCYCLE.
- Let m be the number of vertices in G.
- Consider the complete graph G' with $V(G') = V(G)$ and with the cost of edge (u, v) equal to

$$cost(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E(G) \\ m + 1 & \text{if } (u, v) \notin E(G) \end{cases}$$

- Then the converted instance for TSP will be $\langle G', m \rangle$.

The Traveling Salesperson Problem (Continued...)

Part 2: *UHAMCYCLE* $\leq_p$ *TSP*

- Clearly, a Hamiltonian cycle in G translates to a Hamiltonian cycle in G' with each edge cost equal to 1.
- Conversely, if G' has a Hamiltonian cycle of $cost \leq m$, that cycle can not use an edge of cost $m + 1$, that is, an edge not in $E(G)$. Thus, this cycle resides in G as well.
- Clearly, this reduction is done in poly-time.
- Thus, *UHAMCYCLE* $\leq_p$ *TSP*.
- As a result, $TSP \in NP$ and also *TSP* is NP-hard, and so, *TSP* is NP-Complete.

Problem: Two computational problems P_1 and P_2 are called *polynomial-time equivalent* if there exist polynomial-time reductions $P_1 \leq_p P_2$ and $P_2 \leq_p P_1$. Prove or disprove: Every two NP-Complete problems are polynomial-time equivalent.

- ?

Problem: Two computational problems P_1 and P_2 are called *polynomial-time equivalent* if there exist polynomial-time reductions $P_1 \leq_p P_2$ and $P_2 \leq_p P_1$. Prove or disprove: Every two NP-Complete problems are polynomial-time equivalent.

- TRUE.
- By definition, $P_1, P_2 \in NP$.
- Since P_2 is NP-Complete, there exists a polynomial-time reduction $P_1 \leq_p P_2$.
- Moreover, since P_1 is NP-Complete, there exists a polynomial-time reduction $P_2 \leq_p P_1$.
- Therefore, P_1 and P_2 are polynomial-time equivalent.