Classical NP-complete problems

Richard Karp: Many problems commonly encountered are NP-complete!

Classical NP-complete problems

Richard Karp: Many problems commonly encountered are NP-complete!

Definition

In the *Set Cover* problem, we are given sets $S_1, \cdots, S_n$ and an integer k, and we are asked if there exists $T \subseteq \{1, 2, \ldots, n\}$, with $|T| \leq k$, such that $\cup_{i \in T} S_i = \cup_i S_i$.

Classical NP-complete problems

Richard Karp: Many problems commonly encountered are NP-complete!

Definition

In the *Set Cover* problem, we are given sets $S_1, \cdots, S_n$ and an integer k, and we are asked if there exists $T \subseteq \{1, 2, \ldots, n\}$, with $|T| \leq k$, such that $\cup_{i \in T} S_i = \cup_i S_i$.

Proposition

Set Cover is NP*-complete.*

Classical NP-complete problems: Sequencing

Definition

In a directed graph, a *Hamiltonian path* is a path that visits each vertex exactly once. A *Hamiltonian cycle* is a cycle that visits each vertex exactly once. In the Hamiltonian cycle(path) problem, we are given a directed graph and need to decide whether there exists a Hamiltonian cycle(path).

Classical NP-complete problems: Sequencing

Definition

In a directed graph, a *Hamiltonian path* is a path that visits each vertex exactly once. A *Hamiltonian cycle* is a cycle that visits each vertex exactly once. In the Hamiltonian cycle(path) problem, we are given a directed graph and need to decide whether there exists a Hamiltonian cycle(path).

Proposition

Hamiltonian cycle problem $\leq_{\mathrm{P}}$ *Hamiltonian path problem.*

Classical NP-complete problems: Sequencing

Definition

In a directed graph, a *Hamiltonian path* is a path that visits each vertex exactly once. A *Hamiltonian cycle* is a cycle that visits each vertex exactly once. In the Hamiltonian cycle(path) problem, we are given a directed graph and need to decide whether there exists a Hamiltonian cycle(path).

Proposition

Hamiltonian cycle problem $\leq_{\mathrm{P}}$ *Hamiltonian path problem.*

Proposition

The Hamiltonian cycle problem is NP*-complete.*

Classical NP-complete problems

Definition

Given n nodes $v_1, \ldots, v_n$, a *tour* is a path that starts from v_1, visits every other node exactly once, and returns to v_1.

In the *Traveling Salesman Problem (TSP)*, we are given n nodes and a distance $d_{i,j}$ from each node v_i to another one v_j, and a bound D. We are asked to decide whether there is a tour of total distance at most D.

Classical NP-complete problems

Definition

Given n nodes $v_1, \ldots, v_n$, a *tour* is a path that starts from v_1, visits every other node exactly once, and returns to v_1.

In the *Traveling Salesman Problem (TSP)*, we are given n nodes and a distance $d_{i,j}$ from each node v_i to another one v_j, and a bound D. We are asked to decide whether there is a tour of total distance at most D.

Proposition

Hamiltonian Cycle $\leq_{\mathrm{P}}$ *TSP.* $\Rightarrow$ *TSP is* NP*-complete.*

Classical NP-complete problems

Definition

Given n nodes $v_1, \ldots, v_n$, a *tour* is a path that starts from v_1, visits every other node exactly once, and returns to v_1.
In the *Traveling Salesman Problem (TSP)*, we are given n nodes and a distance $d_{i,j}$ from each node v_i to another one v_j, and a bound D. We are asked to decide whether there is a tour of total distance at most D.

Proposition

Hamiltonian Cycle $\leq_P$ *TSP.* $\Rightarrow$ *TSP is* NP*-complete.*

Remark: When $d_{i,j} = d_{j,i}$, $\forall i, j$, the problem is called a *symmetric TSP* problem; otherwise it is said to be asymmetric.

Classical NP-complete problems

Definition

Given n nodes $v_1, \ldots, v_n$, a *tour* is a path that starts from v_1, visits every other node exactly once, and returns to v_1.
In the *Traveling Salesman Problem (TSP)*, we are given n nodes and a distance $d_{i,j}$ from each node v_i to another one v_j, and a bound D. We are asked to decide whether there is a tour of total distance at most D.

Proposition

Hamiltonian Cycle $\leq_P$ *TSP.* $\Rightarrow$ *TSP is* NP*-complete.*

Remark: When $d_{i,j} = d_{j,i}$, $\forall i, j$, the problem is called a *symmetric TSP* problem; otherwise it is said to be asymmetric.
When $d_{i,j} + d_{j,k} \geq d_{i,k}$, $\forall i, j, k$, the problem is called a *metric TSP* problem.

Classical NP-complete problems

Definition

Given n nodes $v_1, \ldots, v_n$, a *tour* is a path that starts from v_1, visits every other node exactly once, and returns to v_1.
In the *Traveling Salesman Problem (TSP)*, we are given n nodes and a distance $d_{i,j}$ from each node v_i to another one v_j, and a bound D. We are asked to decide whether there is a tour of total distance at most D.

Proposition

Hamiltonian Cycle $\leq_{\mathrm{P}}$ *TSP.* $\Rightarrow$ *TSP is* NP*-complete.*

Remark: When $d_{i,j} = d_{j,i}, \forall i, j$, the problem is called a *symmetric TSP* problem; otherwise it is said to be asymmetric.
When $d_{i,j} + d_{j,k} \geq d_{i,k}, \forall i, j, k$, the problem is called a *metric TSP* problem.
In fact we show that asymmetric, metric TSP is NP-complete.

Classical NP-complete problems: Partitioning

Definition

Given disjoint sets X, Y and Z, each of size n, and given a set $T \subseteq X \times Y \times Z$ of ordered triples, the *3-Dimensional Matching* problem asks whether there exist a set of n triples in T so that each element of $X \cup Y \cup Z$ is contained in exactly one of the n triples.

Classical NP-complete problems: Partitioning

Definition

Given disjoint sets X, Y and Z, each of size n, and given a set $T \subseteq X \times Y \times Z$ of ordered triples, the *3-Dimensional Matching* problem asks whether there exist a set of n triples in T so that each element of $X \cup Y \cup Z$ is contained in exactly one of the n triples.

Proposition

3-Dimensional Matching is NP*-complete.*

Classical NP-complete problems: Partitioning

Definition

A *coloring* of an undirected graph is an assignment of colors to vertices, so that no two adjacent vertices share the same color. In graph theory, the minimal number of colors for which such an assignment is possible is called the *chromatic number*, often denoted as $\chi(G)$ for graph G.
The prolbem of *3-Coloring* asks, given a graph G, whether $\chi(G) \leq 3$, i.e., whether G can be colored using three colors.

Classical NP-complete problems: Partitioning

Definition

A *coloring* of an undirected graph is an assignment of colors to vertices, so that no two adjacent vertices share the same color. In graph theory, the minimal number of colors for which such an assignment is possible is called the *chromatic number*, often denoted as $\chi(G)$ for graph G.
The prolbem of *3-Coloring* asks, given a graph G, whether $\chi(G) \leq 3$, i.e., whether G can be colored using three colors.

Proposition

$\chi(G) \leq 2$ if and only if G is bipartite.

One may check bipartiteness in linear time.

Classical NP-complete problems: Partitioning

Definition

A *coloring* of an undirected graph is an assignment of colors to vertices, so that no two adjacent vertices share the same color. In graph theory, the minimal number of colors for which such an assignment is possible is called the *chromatic number*, often denoted as $\chi(G)$ for graph G.
The prolbem of *3-Coloring* asks, given a graph G, whether $\chi(G) \leq 3$, i.e., whether G can be colored using three colors.

Proposition

$\chi(G) \leq 2$ *if and only if* G *is bipartite.*

One may check bipartiteness in linear time.

Proposition

3-Coloring is NP*-complete.*

Classical NP-complete problems: Numerical

Definition

Given a set of integers $a_1, \ldots, a_n$ and a target K, the problem of *Subset Sum* asks whether there exists $S \subseteq [n]$ so that $\sum_{i \in S} a_i = K$, where $[n]$ denotes $\{1, 2, \ldots, n\}$.

Classical NP-complete problems: Numerical

Definition

Given a set of integers $a_1, \ldots, a_n$ and a target K, the problem of *Subset Sum* asks whether there exists $S \subseteq [n]$ so that $\sum_{i \in S} a_i = K$, where $[n]$ denotes $\{1, 2, \ldots, n\}$.

Proposition

Subset Sum is NP*-complete.*

Definition

Given a set of n items with weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$, the size of a knapsack B and a target value K, the *Knapsack Problem* asks whether one can fill the knapsack with items of total value at least K; that is, whether there is $S \subseteq [n]$, so that $\sum_{i \in S} w_i \leq B$, and $\sum_{i \in S} v_i \geq K$.

Definition

Given a set of n items with weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$, the size of a knapsack B and a target value K, the *Knapsack Problem* asks whether one can fill the knapsack with items of total value at least K; that is, whether there is $S \subseteq [n]$, so that $\sum_{i \in S} w_i \leq B$, and $\sum_{i \in S} v_i \geq K$.

Proposition

The Knapsack problem is NP*-complete.*

Categories of basic NP-complete problems

- Packing problems: Independent Set
- Covering problems: Vertex cover
- Sequencing problems: Hamiltonian Cycle, Hamiltonian Path, Traveling Salesman Problem
- Partitioning problems: 3-Dimensional Matching, 3-Coloring
- Numerical problems: Subset Sum, Knapsack

Don't forget about 3-SAT, sometimes a very convenient starting point.

Exercises

Does the following problem admit a polynomial-time algorithm or is it NP-complete?

1. Given a set $A = \{a_1, \dots, a_n\}$, a collection $B_1, B_2, \cdots, B_m$ of subsets of A, and an integer $k > 0$. Is there a set $H \subseteq A$, $|H| \leq k$ such that $H \cap B_i \neq \emptyset$ for $i = 1, \dots, m$?

Exercises

Does the following problem admit a polynomial-time algorithm or is it NP-complete?

1. Given a set $A = \{a_1, \ldots, a_n\}$, a collection $B_1, B_2, \cdots, B_m$ of subsets of A, and an integer $k > 0$. Is there a set $H \subseteq A$, $|H| \leq k$ such that $H \cap B_i \neq \emptyset$ for $i = 1, \ldots, m$?
2. Given a directed graph $G = (V, E)$ with $s, t \in V$, and an integer $k > 0$.
    1. Does there exist at least k edge-disjoint paths from s to t?

Exercises

Does the following problem admit a polynomial-time algorithm or is it NP-complete?

1. Given a set $A = \{a_1, \ldots, a_n\}$, a collection $B_1, B_2, \cdots, B_m$ of subsets of A, and an integer $k > 0$. Is there a set $H \subseteq A$, $|H| \leq k$ such that $H \cap B_i \neq \emptyset$ for $i = 1, \ldots, m$?
2. Given a directed graph $G = (V, E)$ with $s, t \in V$, and an integer $k > 0$.
    1. Does there exist at least k edge-disjoint paths from s to t?
    2. Given m paths $P_1, \cdots, P_m$ from s to t, does there exist at least k paths among $P_1, \cdots, P_m$ that are edge-disjoint?

Exercises

1. Given an undirected graph and integer k, decide whether there is a spanning tree in which each node has degree at most k.

Exercises

1. Given an undirected graph and integer k, decide whether there is a spanning tree in which each node has degree at most k.
2. Given a directed graph $G = (V, E)$ with $s, t \in V$ and nonnegative integral edge weights, and an integer $k > 0$.
    1. Does there exist a simple path from s to t with total weight at most k?

Exercises

1. Given an undirected graph and integer k, decide whether there is a spanning tree in which each node has degree at most k.
2. Given a directed graph $G = (V, E)$ with $s, t \in V$ and nonnegative integral edge weights, and an integer $k > 0$.
   1. Does there exist a simple path from s to t with total weight at most k?
   2. Does there exist a simple path from s to t with total weight at least k?