LECTURE 13 – THE NP-COMPLETE CLASS PART 2

Overview

- More NP-complete problems
- Applications of NP-complete Problems

MORE NP-complete PROBLEMS

The partitioning (PARTITION) problem: given a set of n elements, each with a positive integer size, and such the sum of all elements sizes is an even number, can the set be partitioned into two subsets such that the sum of sizes of the elements is one subset is equal to that that of the other subset?

The SUBSET SUM problem: given a set of n elements, a positive integer size for each element, and a positive integer goal k, can a subset of elements be found such that the sum of the size of its elements is exactly equal to k?

The TRIPARTITE MATCHING problem: given three sets A, B and C each of size n, and a ternary relation $R \subseteq A \times B \times C$, is there a set of n triplets in R, where no two triplets have an element in common?

The EXACT COVER problem: given a set U of size n, and a set $F = \{U_1,\ldots,U_m\}$ of subsets of U, is there an exact cover of U? that is, a subset $C \subseteq F$ such that the sets in C are disjoint and their union makes the set U?

The Maximum Cut (MAXCUT) problem: given a graph G=(V,E) and a positive integer bound k, can the set of vertices V be partitioned into two subsets such that the number of edges with endpoints in both subsets is no smaller than k?

Graph 3-colouring (3-COLOURING) problem: given a graph G=(V,E) and a set of three colours, is there an assignment of colours to the vertices such that no two adjacent vertices have the same colour, i.e. no edge has identically coloured endpoints?

The INDEPENDENT SET problem: given a graph G=(V,E) and a positive integer bound k, is there an independent set (for every pair of vertices i,j in an independent set, there is no edge between i and j) of vertices $I \subseteq V$ with $|I|=k$?

The HAMILTON PATH problem: given a graph G=(V,E), does G have a Hamilton path? i.e. a path that visits each vertex exactly once?

The RURAL POSTMAN problem: given a graph G=(V,E), a subset of edges $E' \subseteq E$ and a positive integer bound k, is there a circuit in G that includes each edge in E' and has a total length $\leq k$?

The BIN PACKING problem: given a set of n elements, where each i^{th} element as a positive integer size s_i, given a positive integer bin size b and a positive integer bound k, can the set of elements be partitioned into no more than k subsets so that the sum of the sizes of the elements of any subset is no larger than b?

The KNAPSACK problem: given a set of n items, where each i^{th} item has a positive integer size s_i and a positive integer profit p_i, given a positive integer size bound k and a positive integer profit bound m. Can a subset of the elements be found such that the sum of the sizes of its elements is no larger than k and the sum of the profits of its elements is no smaller that m?

Exercise 13.1 For each of the NP-complete problems above, illustrate a problem instance and a solution to that instance.

There is a noticeable characteristic of **NP-complete** problems. This is that the problem statement allows some freedom in the choice of the solution structure.

That is, many NP-complete problems involve the selection of a loosely structured subset, i.e. a subset such that the inclusion of one element does not lead directly to the inclusion of another element.

The difficulty then resides in the subset search. The property obeyed by the subset need not be difficult to verify. In fact, the definition of the class **NP** guarantees that such property is easily verifiable.

NP-complete problems are considered the most difficult ones in the class **NP**. If a problem is shown to be **NP-complete**, this means that it is among the least unlike to be in **P**, those will be solved in polynomial time only if **P = NP**.

Knowing that a problem is **NP-complete** allows us to assume that no effective algorithm exists to solve the problem (unless **P = NP**) and therefore, search for good enough solution approaches.

Given and NP-complete problem Π_1, the logic to prove that a problem Π_2 is also NP-complete is as follows:

1. Since Π_1 is NP-complete, all problems in NP $\propto \Pi_1$
2. Show $\Pi_1 \propto \Pi_2$
3. Then, all problems in NP $\propto \Pi_2$
4. Therefore, Π_2 is also NP-complete

APPLICATIONS OF NP-complete PROBLEMS

Exercise 13.2 For each of the real-world problems described below, state which of the above NP-complete decision problems.

- A set of tasks with known and arbitrary execution times must be executed on a set of identical processors by a given deadline. The problem is to schedule all the tasks onto the least number of processors so that the deadline is met.

- Frequencies from the electromagnetic spectrum should be assigned to mobile radios and devices. Two devices that are sufficiently close must be assigned different frequencies. Devices can share frequencies if they are separated by a given minimum distance. The problem is to minimise the number of frequencies that should be used to service all the devices.

- There is a sports day in the University and students sign up to participate in events. After students have signed up, a schedule should be constructed. Two events can be scheduled simultaneously if no student has signed up for both events. There is no restriction on venues for holding simultaneous events. There is one particular timeslot in which the organisers want to have as many events as possible because sponsors will be on site.

A special case of the Knapsack problem

In the steel industry a common problem is that of inventory matching. In this scenario, production planning involves assigning a given set of orders to the production units in the inventory. Manufacturing consideration such as the compatibility of orders and production units in terms of quality, size, etc. impose additional assignment constraints. Because of this, in general, it is not possible to assign a given order to certain production units. As production operations involve more complex processes and larger product variety, the problem becomes more complex.

Multiple Knapsack Problems with Assignment Restrictions (MKAR):

A set of n items where each item j has a positive weight w_j

A set of m knapsacks where each knapsack i has a positive real capacity c_i

For each item j there is a set of knapsacks that can hold the item j

Let B_i be the set of items that can be assigned to knapsack i then the time j is admissible to knapsack i if $j \in B_i$

The problem is to choose a subset of items to be assigned to each knapsack such that: all the subsets are disjoint (each item is assigned to at most one knapsack) and the capacity of each knapsack is not exceeded.

More formally:

Maximise $\sum_{i \in M} \sum_{j \in B_i} w_j x_{ij}$

Subject to $\sum_{j \in B_i} w_j x_{ij} \leq c_i, i \in M$

$\sum_{i \in A_j} x_{ij} \leq 1, j \in N$

$x_{ij} \in \{0,1\}, i \in A_j, j \in N$

Reading: (Garey&Johnson, Appendix)