NP-Completeness and Approximation Algorithms

MA428, Dr Katerina Papadaki

I. Exercise 1

A clique in a graph is a subset S of vertices such that every vertex is adjacent to every other. A maximum clique is one with as many vertices as possible. A maximal clique is one that cannot be enlarged by adding a vertex.
The *max-clique problem*: Given a graph G, find the maximum clique.
Decision Version of the max-clique problem(DVMC): Given G and integer k, does G contain a clique of size $\geq k$?

An Independent Set in a graph is a set of nodes no two of which have an edge. e.g., in a 7-cycle, the largest independent set has size 3, and in the graph coloring problem, the set of nodes colored red is an independent set.

The *Independent Set problem* is: Given a graph G find a maximum independent set.
Decision version of the Independent Set problem (DVIS): Given G and an integer k, does G have an independent set of size $\geq k$?

A vertex cover in a graph is a set of nodes such that every edge is incident to at least one of them. We defined the vertex cover problem in lecture, which was a general version. In the literature, the term *vertex cover* usually refers to the unweighted or minimum cardinality vertex

K. Papadaki is with the Department of Management, Operational Research Group at London School of Economics, United Kingdom
E-mail: `k.p.papadaki@lse.ac.uk`

cover, which is the one we use here, and the decision version of that is:

Decision version of Vertex Cover problem (DVVC): Given a graph G and an integer k, does G have a vertex cover of size $\leq k$?

1) Show an example where a maximal clique is not a maximum clique.
2) Use the fact that the DVMC problem is NP-complete, to show that the DVIS problem is NP-complete.
3) Use the fact that the DVMC problem is NP-complete, to show that the DVVC problem is NP-complete.

II. Exercise 2

Suppose we have a set of elements $N = \{1, 2, 3, 4\}$, and the following subsets of the elements: $S_1 = \{2, 3, 4\}$, $S_2 = \{1, 3, 4\}$, $S_3 = \{1, 2, 4\}$, $S_4 = \{1, 2, 3\}$, with respective costs $w_j = 1$ for $j = 1, 2, 3, 4$. Suppose we want to pick a collection of subsets S_j whose union is N at minimum cost.

1) Write the IP formulation of the problem.
2) Write the LP relaxation of the IP formulation.
3) Solve the LP (using an LP solver, like the Excel solver) and use the optimal solution of the LP to produce an α-approximation algorithm for the above problem.
4) What is the value of the solution of the approximation algorithm?
5) What is the value of α?
6) Find the optimal value of the problem (by inspection) and compare it to the approximate value given by the algorithm and compare it also to the worst case given by α.

III. Exercise 3

Give a polynomial time algorithm with oracle access to the decision version of the knapsack problem, that takes a knapsack instance (x, C) and either FINDS a feasible solution of value at least C or says that none exists.

IV. Exercise 4

The *Constrained Shortest Path* (CSP) problem is defined as follows:

Given a directed graph with a length l_{ij} and traversal time t_{ij} associated with each arc, the Constrained Shortest Path problem asks if there is a path from vertex s to vertex t with length at most L and time at most T.

Consider the following general decision version of the knapsack:

Input: $x = (n, (v_1, \ldots, v_n), (s_1, \ldots, s_n))$, a value V and knapsack size S. So an instance is (x, V, S) and the question is "can we select items with total value $\geq V$ and size $\leq S$?"

Show that the CSP problem is NP-complete, by reducing the above decision version of the knapsack to the constrained shortest path problem:

Knapsack $\preceq$ CSP, i.e. if we could solve the CSP in poly-time then we could solve the Knapsack.