Advanced Analysis of Algorithms - Homework IV

K. Subramani
LCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Instructions

1. The homework is due on December 6, in class.
2. Each question is worth 4 points.
3. Attempt as many problems as you can. You will be given partial credit, as per the policy discussed in class.

2 Problems

1. (a) Let S denote a non-empty convex set of the space E^n. A point $\mathbf{u} \in S$ is called an extreme point of S, if it cannot be written as a strict convex combination of two other points in S. In other words, there do not exist points $\mathbf{r}, \mathbf{t} \in S$, such that $\mathbf{u} = \alpha \cdot \mathbf{r} + (1 - \alpha) \cdot \mathbf{t}$, where $0 < \alpha < 1$. Show that $\mathbf{u}$ is an extreme point of S, if and only if the set $S - \{\mathbf{u}\}$ is a convex set.

   (b) Let $S_1, S_2, \ldots, S_n$ denote n convex sets. Is the set $S' = \cup_{i=1}^{n} S_i$ convex? How about the set $S'' = \cap_{i=1}^{n} S_i$? An assertion of convexity requires a formal proof; an assertion of non-convexity requires a counterexample.

2. Consider a directed acylic graph $\mathbf{G} = \langle \mathbf{V}, \mathbf{E}, \mathbf{c} \rangle$. The function $\mathbf{c} : \mathbf{E} \to \mathcal{Z}_+$ associates a positive integer with edge $e_{ij} \in \mathbf{E}$ and it represents the capacity of that edge. Let $s, t \in \mathbf{V}$ denote two special vertices known as the *source vertex* and the *destination vertex* respectively. A product which is stored at s must be shipped to t, along the edges of the network. How would you determine the maximum amount of product that can be shipped from s to t, under the following assumptions:

   (a) the amount of commodity on an edge of the network may never exceed the capacity of that edge,

   (b) no vertex other than s and t may store the product.,

   (c) the shipping is lossless, i.e., there is no loss in the transportation.

3. The Branch-And-Bound enumeration technique solves integer programs by solving linear programs. A very nice exposition is available here:
   `http://compalg.inf.elte.hu/~tony/Oktatas/SecondExpert/Chapter24-Branch-6April.pdf`.

   Essentially, we use the linear programming relaxations to find good bounds, which permits the pruning of the search tree. Solve the following integer program using Branch-And-Bound enumeration:

$$\begin{array}{rcc}
\max z & = -x_1 + 4x_2 & \\
\text{subject to} & & \\
-10 \cdot x_1 + 20 \cdot x_2 & \leq & 22 \\
5 \cdot x_1 + 10 \cdot x_2 & \leq & 49 \\
8 \cdot x_1 - x_2 & \leq & 36 \\
x_1, x_2, x_3, x_4 & \geq & 0 \\
& x_1, x_2, x_3, x_4 \ \text{integer} &
\end{array}$$

To solve this problem, you do need to know how to solve 2-dimensional linear programs. This can be done either graphically or through the SIMPLEX algorithm.

4. The Maximum Independent Subset (MIS) problem is defined as follows: Given an undirected graph $\mathbf{G} = \langle V, E \rangle$ and a target K, is there a set V' of vertices, such that $V' \subseteq V$, $|V'| \geq K$ and two vertices in V' are connected by an edge in G. In class, we mentioned the fact that the MIS problem is is **NP-complete**. What can you say about the complexity of the problem, if the graph has no cycles? If you claim that the problem is in **P**, then you should provide a polynomial time algorithm. On the other hand, if claim that the problem is **NP-complete**, then you must provide a reduction from an **NP-hard** problem to the MIS problem on trees.

5. (a) Let $S = \{a_1, a_2, \ldots, a_n\}$ denote a finite set. Let C denote a collection of m subsets of S, i.e., $C = \{S_1, S_2, \ldots S_m\}$, where each $S_i \subseteq S$. We are interested in the following question: Is there a collection of subsets from C, such that the union of the sets in this collection is S and the cardinality of the collection is at most K? Is this query **NP-complete**?

   (b) Let $S = \{a_1, a_2, \ldots, a_n\}$ denote a finite set. Let C denote a collection of m subsets of S, i.e., $C = \{S_1, S_2, \ldots S_m\}$, where each $S_i \subseteq S$. We are interested in the following problem: Is there a partition of S into two subsets R and T, such that no set in C is completely contained in either R or T? Prove that this problem is **NP-complete**.