Assignment 1

Problems from Textbook 2 by Papadimitriou & Steiglitz (I am using the notation Chapter#.Problem#): **1.1(d), 1.3, 1.6, 1.9, 1.10, 2.8.**

1.1(d)

Formulate the following as an optimization problem instance $(F,\ c)$, giving the domain of feasible solutions F and the cost function c.

(d) Find a cylinder with a given surface area A that has the largest (maximize) volume V.

Domain of feasible solutions F:

Surface area $A\ =\ 2\pi rh\ +\ 2\pi r^2$

Feasible set F = {all pairs (h, r) that satisfy constraint $A\ =\ 2\pi rh\ +\ 2\pi r^2$ with height height $h\ >\ 0$ and radius $r\ >\ 0$.

F: $\{(h, r)$: $2\pi rh\ +\ 2\pi r^2\ =\ A,\ h\ >\ 0, r\ >\ 0\}$

Cost function c (to maximize): $c(f) \geq c(y)\ \ for\ all\ \ y \in F$

Find the (h, r) pair that makes the value of the function c at that (h, r) pair ≥ any other (h, r) pair in the feasible set F.

Volume $V\ =\ \pi r^2 h$

max $f(x)$ = min -$f(x)$:

To maximize the volume V, we need to minimize $c(h, r)$: $c(h, r)$ = -$\pi r^2 h$ or $c(h, r)$ = $\frac{1}{\pi r^2 h}$ = $\frac{1}{V}$

1.3

Show that the neighborhood defined in Example 1.5 for the MST is exact.

Example 1.5

In the MST, an important neighborhood is defined by

$N(f) = \{g : g \in F$ and g can be obtained from f as follows: add an edge e to the tree f, producing a cycle; then delete any edge on the cycle$\}$ □

The neighborhood is exact if local optimality implies global optimality.

A minimum spanning tree must be acyclic and edges (E) must equal Vertices $(V\ -\ 1)$

The goal is to show that if a spanning tree f is locally optimal (i.e., no tree g in its neighborhood $N(f)$ has a lower weight), then it must also be globally optimal, meaning no tree in the entire feasible set F has a lower weight.

<u>Local optimality</u>: The neighborhood $N(f)$ encompasses all possible spanning trees generated by adding an edge e to tree f to form a cycle, and then removing another edge in the cycle, effectively exploring every potential edge swap between different spanning trees.Therefore, if f is locally optimal in this neighborhood, there is no tree in $N(f)$ with a lower weight.

<u>Global Optimality</u>: We assume by contradiction that f is not globally optimal. Let's assume that the globally optimal tree is M, which has a lower total weight than f.

If f is not globally optimal, there must exist at least one edge e_1 in M that is not in f, and at least one edge e_2 in f that has a higher weight than e_1. These two edges, e_1 and e_2, form a cycle as they share one node in common. It is possible to add e_1 to e_2, creating a cycle, and then remove the higher-weight edge e_2, resulting in a new spanning tree with a lower total weight than f. This contradicts the assumption that f is locally optimal, since we have found a tree with lower weight within its neighborhood.

Because this contradiction shows that no such tree M can exist with a lower weight than f, the assumption that f is not globally optimal must be false. Therefore, f must be globally optimal and the neighborhood $N(f)$ **is exact.**

1.6

Suppose we are given a set S containing $2n$ integers, and we wish to partition it into two sets S_1 and S_2 so that $|S_1| = |S_2| = n$ and so that the sum of the numbers S_1 is as close as possible to the sum of those in S_2.

Let the neighborhood N be determined by all possible interchanges of two integers between $|S_1|$ and $|S_2|$.

Is *N* exact?

Neighborhood $N(S_1, S_2)$ = {all partitions (S_1, S_2) that can be obtained by swapping two integers between $|S_1|$ and $|S_2|$.

Local optimality: Assume S_1 is locally optimal (no single replacement results in a smaller difference between the sums of $|S_1|$ and $|S_2|$

Global optimality: Assume S_1' is the globally optimal partition, where the difference between the sums of $|S_1|$ and $|S_2|$ is the smallest.

Proof by contradiction:

Since S_1 is different from S_1', there must be an element $l \in S_1$' that is not in S_1, and an element $i \in S_1$ that is not in S_1'.

If $l > i$, swapping i with l would make the sum of S_1 larger, which contradicts the assumption that S_1 is locally optimal (because we would improve the balance by the swap).

There cannot be such an element i, so S_1 must be both locally and globally optimal. Therefore, the neighborhood $N(S_1, S_2)$ **is exact.**

1.9

Let $f(x)$ be convex in *R*ⁿ.

Fix $x_2, \ldots, x_n$ and consider the function $g(x_1) = f(x_1, \ldots, x_n)$.

Is g convex in R^1?

Yes

Fixing $x_2, x_3, \ldots x_n$ to constant values reduces the function $f(x_1, \ldots, x_n)$ to a new $g(x_1)$ one-dimensional (R^1) function of x_1.

Since f is convex in R^n, it means that f is convex along any direction in R^n, including along the line where $x_2, x_3, \ldots x_n$ are held fixed (constant). This implies that the function $g(x_1) = f(x_1, \ldots, x_n)$ remains convex in R^1.

Let $S \subseteq R^n$ be a convex set.

c: $S \rightarrow R^1$ is convex in S if for any two points $x, y \in S$ the following equality is satisfied:

$c(\lambda x + 1 - \lambda)y \leq \lambda c(x) + (1 - \lambda)c(y)$, $\lambda \in R$ and $0 \leq \lambda \leq 1$

If $S = R^n$, c is simply called convex.

If x and y are vectors in R^n, the same convexity condition holds. You apply the weighted average to each component of x and y.

$f(\lambda x_1 + (1 - \lambda)y_1, \ldots, \lambda x_n + (1 - \lambda)y_n) \leq \lambda f(\lambda x_1, \ldots, x_n) + (1 - \lambda)f(y_1, \ldots, y_n)$

This says that the function's convexity applies to each individual component of the vectors.

Convexity in R^1 means that the convexity property still holds for functions with just one variable, making it a simpler version of the multi-variable case.

1.10
Let $f(x_i)$ be a convex function of the single variable x_i.

Then $g(x) = f(x_i)$ can also be considered as a function of $x \in R^n$.

Is $g(x)$ convex in R^n?

Yes

Even though $g(x)$ depends on the vector $x = (x_1, x_2, ..., x_n) \in R^n$, it only varies with respect to the i^{th} component x_i. Therefore, The convexity of $f(x_i)$ in one dimension R^1 transfers directly to $g(x)$ in the higher-dimensional space R^n.

Since $g(x) = f(x_i)$ behaves just like the one-variable function $f(x_i)$ and only depends on x_i, $g(x)$ will be convex in R^n if $f(x_i)$ is convex in R^1.

Let be $x_i y_i \in R,\ 0 \leq \lambda \leq 1$
Definition of convexity for f: $f(x_i + (1 - \lambda)y_i) \leq \lambda f(x_i) + (1 - \lambda)f(y_i)$

But $g(x) = f(x_i)$: $g(\lambda x + (1 - \lambda)y) \leq \lambda g(x) + (1 - \lambda)g(y)$ = definition of convexity for g in R^n.

2.8
Show that the set of optimal points of an instance of LP is a convex set.

Let x and y be two optimal points (i.e., $cx = cy = c_o$, minimum cost).

A convex combination of two points $x, y \in R^n$ is a point z.
$z = \lambda x + (1 - \lambda)y, \lambda \in R$ and $0 \leq \lambda \leq 1$
Since x and y are part of the feasible space, any point z also lies within this space. This is because the feasible space forms a convex set, which means a combination of points in the space remains in the space.

Prove feasibility of z:
Feasibility means that the solution satisfies the constraints. We can show z is feasible because it satisfies the same constraints that apply to x and y:

$Ax = b$
$Ay = b$

which means that $Az = b$

Prove that z is optimal:
$c(z) = c(z) = c(\lambda x + (1 - \lambda)y) = c\lambda x + c(1 - \lambda)y = \lambda cx + (1 - \lambda)cy = c_0$
Thus, the cost of z is the same as the cost of x and y, meaning z is also optimal.

Therefore, the set of optimal points forms a convex set in linear programming, meaning any convex combination of optimal points is still optimal.