Name: Prasham Dhaneshbhai Sheth
UNI: pds2136

HW:3 — Theoretical part

1. **Solution to problem 1**

Flow network with demands is a directed capacitated graph with potentially multiple sources and sinks, which may have incoming and outgoing edges respectively. In particular, each node $v \in V$ has an integer demand d(v); if $d(v) > 0, v\ is\ a\ sink,\ while\ if\ d(v) < 0, it\ is\ a\ source.$ Let S be the set of source nodes and T the set of sink nodes.

A circulation with demands is a function $f : E- > R+$ that satisfies the given capacity and demand constraints.

(1) For a feasible circulation with demands to exist, we must have sum of all demands equal to sum of all the supplies (i.e.) total demand shall equal total supplies. Here supplies are treated as negative demands and hence the following mathematical equation arises:

$$\Sigma d(v) = -\Sigma d(u);\ where\ u \in S\ and\ v \in T$$

(2) We can reduce the problem "is there a circulation f with demands that meets both capacity and demand conditions" to a decision version of max flow where in we answer "Yes" if there exists a maximum flow in the equivalent flow network with value equal to $\Sigma d(v)$ where $v \in T$ and "No" otherwise.

For making an equivalent flow network we modify the given network by following the steps stated below:

1: Add a new source s' with an edge (s' , s) from s' to every node $s \in S$.

2: Add a new source t' with an edge (t' , t) from t' to every node $t \in T$.

3: The capacity of edges (s', s) = -d(s) (since d(s) ¡ 0, this is positive)

4: The capacity of edges (t, t') = d(t)

After modifying the network we can use the decision version of the max flow problem to find out whether the max flow of $\Sigma d(v)$ where $v \in T$ exists in the modified version of the graph with source s' and sink t'. If the max flow of that value exists we can say that the original flow network has a circulation f with demands that meets both capacity and demand constraints.

To prove the reduction we need to prove following 2 claims:

1: For any valid feasible circulation with demands f in flow network, we can have a max flow with same value in the equivalent flow network created by us.

Lets say we have a valid feasible circulation with demands f in the flow network. That means that the flow follows the given flow and demand constraints. The way in which we augment the flow network G to get G' is by adding a Source vertex s and a sink vertex t to which all the sources and sinks are connected respectively.

For the equivalent network G' the capacities of all the edges from source s to vertices with negative demand is the absolute value of the demands and for those from vertices with positive demands have demand value as their capacities. The circulation f hence can be converted to a max flow f' in the equivalent graph by utilizing the edges corresponding to G in G' to have an equivalent flow and use the newly introduced edges to meet the flow conservation constraints in the network G'.

The value of the max flow in G' would thus be equal to D which is the value of total demands of the sinks in G as it is the amount that would be sent through all the sources of G which in the sense would be connected to a source S in G'. Hence the total outgoing flow from source s in G' would be equal to D as it is required to constitute a valid circulation in G. The flow would be a max flow as there is no scope of increasing the flow from source s in G' and hence corresponds to the max flow for the network G'.

The resultant flow hence would be a max flow from newly introduces source to newly introduced sink.

2: Given a max flow f' in G', we can construct a feasible circulation with demands f in G, with value equal to the value of the max flow.

If we have a max flow f' in the network G' of value D we can easily find the valid circulation with demands in G. This can be done by sending the flow across the edges E in G with value equal to that in the equivalent edges in G' when having the max flow. The demand constraints of all the sources in G would be satisfied as it would be receiving the same flow as input from the source S in G' and for the sinks in G it would be sending the same amount to Sink T in G'. Hence if we have a max flow with value D in G' we can say that we have a valid circulation in G when same amount of flow is sent through the edges in G as through the equivalent edges in G'.

As we can see that we require to perform steps in polynomial time for converting the network with demands to an equivalent flow network and after that we call the algorithm to get the solution problem for max flow with value D in the reduced network once which means that this a polynomial reduction.

2. **Solution to problem 2**

We are given a flow network G = (V;E) with demands where every edge e has an integer capacity c_e, and an integer lower bound $l_e >= 0$. A circulation f must now satisfy$le <= f(e) <= c_e$ for every $e \in E$, as well as the demand constraints defined in the first question.

In the context of having a lower bounded constraint we can think of a network with minimum circulation equal to the value of lower bound for the capacity constraints. The lower bounded circulation hence would meet the capacity constraints but would fail to meet the demand constraints for the network as the minimum circulation would meet the demand requirements of the nodes only partially. Hence, we can modify the network to an equivalent network which would essentially require to follow the constraints as defined in the first question.

As, described earlier the flow with value equal to the lower bound (L0) such that it takes into consideration the lower bound for each edge would partially satisfy the demands of the nodes.The satisfies demand can be thought of as S_v which is defined as the difference between the flow out of the vertex and the flow inside the vertex when the flow is set to the lower bound. Hence to make an equivalent graph G' following the constraints as defined in the question one we do the following steps:

(1) For each vertex $v \in V$, we update the demand value by subtracting the already satisfies value from it (i.e) D'(v) = D(v) - S(v)

(2) For each edge $e \in E$, update the capacity value by subtracting the lower bound value for the edge from the upper bounded capacity value. (i.e) C'(e) = C(e) - l(e)

This graph G' can be further be solved to find whether a feasible circulation exists in it or not by reducing the same to a max flow problem as defined in the first question. To get the circulation for the original graph G with the lower bounded constraints we add the lower bound corresponding to each edge in the max flow returned by the series of 2 reductions.

To prove the equivalence of the demand network with given lower bound and reduced demand network without constraints we shall prove the following:

1: Existence of a valid circulation in the network G implies the existence of valid circulation in the network G'.

The network G' is formed after removing the lower bound constraints in G. Hence if there exists a valid circulation in G we can easily say that there must exist a valid circulation in G' as well because of the fact that it is specialized version of G in the terms that it just has been constructed after satisfying the lower bound constraints. As a valid circulation exists in G which satisfies the demand constraints considering the lower bounds we can say that the extraneous flow in G would be a valid circulation in G'. This is intuitive after the fact that lower bound only satisfy the flow constraints and hence the extraneous flow can be used to make a valid circulation in G'.

2: Existence of a valid circulation in the network G' implies the existence of valid circulation in the network G.

The existence of valid circulation in the network G' means that we a circulation which when considered with the already satisfied lower bounded circulation in G would make a valid circulation in G. This implies that in the valid circulation of G' if we utilize the capacity of equivalent edges by letting the flow equal to lower bounded value for that edge in addition to that in G' we can have a valid circulation that follows the demand as well as the flow constraints in G. This is intuitively also true as we had seen that allowing the flow equal to the lower bounded value for each edge satisfies the flow constraints but fails to satisfy the demand constraint which we satisfy using the extra flow we supply in the form of that obtained from valid circulation in G'

3. **Solution to problem 3**

(i): The max flow problem can be formulated as a min-cost flow problem. For this we need to reduce the problem of solving max flow to a min-cost flow problem as max flow problem can be thought of as a special case of min-cost flow problem.

We can hence reduce the given flow network for the Max-Flow problem to an equivalent graph for min cost flow problem and then using that equivalent graph and the algorithm to solve the min-cost flow problem we can get the required output for max flow of the original graph.

To reduce the problem of max flow to min-cost flow problem we follow the following steps for reduction:

1) Add an edge from sink(t) to source(s) and set its capacity to infinite (to allow max flow across the edge) and set is cost $a_{ts} = -1$ (We can also use the upper bound for the max flow which is obtained by (n*U) to set the flow for the edge).

2) Set the costs for each of the edges except the edge from t to s as 0.

3) For all the vertices v in V, we set the supply values for each of them as 0. This step ensures that we take into consideration the flow conservation constraint of Max Flow problem.

Use this equivalent graph G' to find the min-cost flow into the graph G'. The flow for the equivalent graph G' that has the minimum cost can be said to have the maximum flow in the original graph G. (The flow on the edge t-s needs to be ignored as it is not and edge in the original graph)

To prove th correctness of the reductions we need to prove: The graph G' has a flow f with minimum cost if and only if the graph G has flow f as the maximum flow from source(s) to (t).

The proof follows from these 2 (forward and backward implications):

- Forward: if the Graph G' has a flow f that corresponds to minimum cost then the same flow f is the max flow in graph G.

Graph G' has a flow f which corresponds to the minimum cost flow in the graph. We can draw inference that the flow f has the maximum value occupied on the edge t-s that it requires to send the flow from t-s. (This can be inferred as we know that the costs along each of the pother edges is zero and hence to minimize the cost of the graph maximum value needs to flow from the edge t-s as it has a negative cost.)

Also, we know that the supply values of all the vertices in the graph have been set to 0 and hence all the flow that enters s from t-s edge needs to exit s and hence the same flow will make to the vertex t through the flow network. As already known that the flow on t-s edges needs to be the maximum, we can say that the same flow when is flowing from s-t would be the max flow for the flow network from s-t which is nothing but the graph G.

- Backward: if f is the max flow in the graph G then f is also a flow with minimum cost in G'.

The graph G' would have minimum cost whenever the sink to source edge (i.e) the t-s edge carries the maximum flow and that is due to the fact that cost for that is -1 and would be minimum whenever the flow through that particular edge is maximum as we have set the cost for all the other edges to be 0. Hence the cost would only be minimized when maximum flow is flowing through the edge t-s.

If f is the max flow in Graph G, we know that f maximizes the flow from s to t. As the supply values of the nodes in G' are set to be 0 the flow entering one vertex always needs to make out from that vertex. Hence the flow into the edge t-s is always the flow that enters into the t from s and hence if f is the max flow in the graph G then f is also a flow with minimum cost in G'.

Hence, from the proof for both backward and forward direction we can say that the problem of max flow can be reduced to the problem of finding the min cost flow.

Also, as we follow a fixed number of steps for converting the graph and the time required for the same is of polynomial order and a single call is made to the procedure for finding the minimum cost flow for the equivalent flow network, we say that the given reduction is a polynomial reduction.

(ii): The linear program for min-cost problem can be shown as following: The problem is of finding a flow satisfying edge capacity constraints and node supplies that minimizes the total cost of the flow.

This can be shown as following:

$min\ a_{ij} * f_{ij}$ where each edge (i,j) $\in$ E

subject to "supply constraint": For every v $\in$ V is $f^{out}(v) - f^{in}(v) = s_v$.

"edge capacity constraint": $0 <= f_{ij} <= C_{ij}$ for each edge (i,j) $\in$ E

Note the problem is not feasible unless the supplies and demands are balanced

(i.e.) $\Sigma s(v) = 0$ For every v $\in$ V.

4. **Solution to problem 4**

(i) We are given a data set of m red and n white points on the plane. We want to separate the red from the white points by a line, if possible; that is, we are looking for a line such that all the red points are on one side, all the white points on the other side and no points lie on the line.

We need to form a linear program for the same which can be done as follows:

It is a case of binary classification of the points and we can define the class of m red points to be positive and that of n white points to be negative. The goal of the problem is to find a line y = a*x + b such that for all the red points $y > 0$ and for the white points $y < 0$.

Both the constraints defined above are strict inequalities and as strict inequalities are not allowed in the linear programs we need to introduce an extra decision variable that can modify the inequalities to not be strict and hence can be sued in the linear program. We use "d" which represents the margin by which the line would separate the set of points from each other.

Hence, we modify the inequalities listed above and can formalize the linear program to maximize the margin d.

Thus, the linear program is to

$maximize\ d$

subject to the following 2 constraints:

$a * x + b - d >= 0$ for all points of red color

$a * x + b - d <= 0$ for all points of white color

The value of optimal value need to be strictly positive in order to be a feasible solution for the problem and in the other case we say that there does not exist a feasible line that can separate the given set of points by their color. In the case when feasible solution exists we say that value of a and b characterize the required line that separates the 2 sets of points.

(ii) Now suppose that our linear program above is infeasible, so no line can separate the red from the white points. We are now wondering whether a quadratic function (parabola) of the form $ax^2 + bx + c$ could separate the points.

As we know the points are not linearly separable in the given dimensions and need to separated by a non linear function. We can thus, use a transformation function (kernel operation in terms of using Support Vector Machines) to transform the given set of points into a set of points which are linearly separable. generally the transformed space would have higher dimensionality then the original one (in our case it would be 1 more than the original one) After finding a line that separated the transformed points we apply inverse transformation to get the points into the original space and when inverse transformation is applied over the line in transformed domain we can obtain a no-linear separating decision boundary.

If we denote the transformation function as f(x) which maps the given point x into (x,x^2) then we can formalize the linear program as the one described in question one just with a modification into constraints as,

$a * f(x) + b - d >= 0$ for all the red points

$a * f(x) + b - d <= 0$ for all the white points