CS 514

Assignment 8 – Linear Programming and Network Flow

report8.txt

Q1.

Solution --

Provided in the hw8.py file submitted.

Q2.

Solution –

We will use a flow network to design the architecture of this problem.

For the nodes, we will have a source node (S), paper nodes (P1, P2, P3, … , P40), reviewer nodes (R1, R2, R3, … , R12) and a sink node(T).

The edges and the capacities of the network can be described as follows –

1. We connect the source node (S) to each paper node (P1, P2, ..., P40) with an edge representing the number of reviewers needed for each paper. We set the capacity of each of these edges to 3 as each paper has to be reviewed by 3 reviewers.
2. Next, we connect each paper node (P1, P2, ..., P40) to the corresponding reviewer nodes (R1, R2, ..., R12) with an edge representing the assignment of a paper to a reviewer. We set the capacity of each of these edges to 1 as this represents the assignment of a paper to a reviewer.
3. Lastly, we connect each reviewer node (R1, R2, ..., R12) to the sink node (T) with an edge representing the maximum number of papers a reviewer can review (which is 11). Hence, we set the capacity of each of these edges to 11.

This network ensures that each paper is reviewed by three reviewers and that no reviewer is assigned more than 11 papers.

To find the maximum flow of the network, we need to maximize the flow from the source (S) to the sink (T), which represents maximizing the number of papers reviewed while respecting the constraints.

Now, we can use Ford-Fulkerson algorithm to find the maximum flow in this network. The solution below is using the concept of matching in bipartite graphs.

The given problem can be modeled as a bipartite graph, where one set of nodes represents papers (P1, P2, ..., P40), another set represents reviewers (R1, R2, ..., R12), and edges connect papers to reviewers. We know each paper must be connected to exactly three reviewers.

As we know finding the maximum flow in a bipartite graph is equivalent to finding a maximum cardinality matching in the graph.

We have 40 papers and 12 reviewers, and each paper needs 3 reviewers. Therefore, we need 40 * 3 = 120 edges in our matching.

Since each reviewer can review a maximum of 11 papers, we have enough capacity for each paper to be reviewed by 3 reviewers.

Thus, the maximum flow (maximum number of papers that can be assigned to reviewers while meeting the given constraints) is 40 (papers) * 3 (reviewers per paper) = 120.

So, the maximum flow in this network is 120.

Q3.

Solution –

Theorem:
A perfect matching in a bipartite graph of N boys and N girls exists if and only if every subset S of boys is connected to at least |S| girls.

Proof:
To prove that there is a perfect match if and only if every subset S of boys is connected to at least |S| girls, we will use the Max-flow Min-cut theorem and Hall's theorem. The Max-flow Min-cut theorem states that the maximum flow in a network is equal to the minimum cut capacity.

Consider a bipartite graph G = (V, E) where V is partitioned into two sets P and Q representing boys and girls, respectively. The edges in E connect vertices from P to vertices in Q. Let's denote the capacities of the edges by c(u, v), where u is in P and v is in Q.

Now, let's create a flow network from this bipartite graph. Add a source node s and connect it to every boy in P with an edge of capacity 1. Similarly, add a sink node t and connect every girl in Q to the sink with an edge of capacity 1.

The idea is to find a flow from the source to the sink such that the total flow leaving the source is equal to the total flow entering the sink, and the flow through each edge is either 0 or 1.

Now, let's define the capacity function c' for the edges in the flow network. If (u, v) is an edge in the original bipartite graph G, then c'(u, v) = 1 if there is an edge (u, v) in G, and c'(u, v) = 0 otherwise.

Apply the Max-flow Min-cut theorem, which states that the maximum flow in a network is equal to the minimum capacity of a cut in the network. Let F be the maximum flow in the network.

1. "If" direction:
- If there is a perfect matching in the original bipartite graph, then there is a flow of value N from the source to the sink in the flow network.
- By the Max-flow Min-cut theorem, the minimum capacity of a cut is also N.
- This implies that for any subset S of boys in P, the minimum cut separating S from the rest of P has capacity at least |S|.
- Since the capacity of each edge is 1, the minimum cut must include exactly one edge for each boy in S, ensuring that every subset S of boys is connected to at least |S| girls.

2. "Only if" direction:
- If every subset S of boys is connected to at least |S| girls, then there is a cut in the flow network with capacity at least N.
- By the Max-flow Min-cut theorem, the maximum flow F is equal to the minimum capacity of a cut.
- If F = N, then there is a perfect matching in the original bipartite graph.

The Max-flow Min-cut theorem allows us to establish the equivalence between the existence of a perfect matching in a bipartite graph and the condition that every subset of boys is connected to at least |S| girls.

Q4.

Solution –

Let's define decision variables and constraints:

Decision Variables:
Let X[u, v]) be a decision variable representing the number of times the edge (u, v) is used in the path from node s to node t.

Objective Function:
Minimize the total length of the path:

$$\text{Minimize } \sum_{(u,v)\in E} l[u,v]\cdot X[u,v]$$

Constraints:
1. Maintain flow conservation at each intermediate node v (except s and t):
$$\sum_{(u,l)\in E} X[v,l] - \sum_{(l,u)\in E.} X[l,v]=0. \quad \text{for all } v \in V \setminus \{s,t\}$$
2. Ensure a single unit of flow leaves the source node s:
$$\sum_{(u,t)\in E} X[u,t]=1$$
3. Ensure a single unit of flow enters the destination node t:
$$\sum_{(s,v)\in E} X[s,v]=1$$
4. Guarantee non-negativity and limit the flow on each edge to the edge length:
$X[u,v]=0$
$X[u,v] \leq l[u,v]$ *for all* (u,v) ▯ E

This linear program aims to minimize the overall path length from node s to t, ensuring adherence to flow conservation constraints at intermediate nodes and maintaining flow on each edge within prescribed bounds.

The dual variables associated with constraints 1 and 4 are denoted as Y[v] and Z[u,v] respectively. The dual program can be expressed as follows:

Dual Objective Function

Maximize Y[s]−Y[t]

Dual Constraints

1. For each intermediate node v:
   $Y[v] - Y[l] + Z[l,v] - Z[v,l] = 0$ *for all* $v \in V / \{s,t\}$

2. For the source node s,
   $Y[s] - Y[l] + Z[l,s] - Z[s,l] = 1$

3. For the destination node t
   $Y[t] - Y[l] + Z[l,t] - Z[t,l] = 0$

4. Dual feasibility conditions:
   $Y[v], Z[u,v] \geq 0$

The dual program seeks to maximize the disparity in dual variables between the source and destination nodes, all the while upholding conditions of feasibility. The dual variables Y[v] denote the "potential" at each node, while Z[u,v] signifies the flow along the respective edge. Through this, the dual program furnishes insights into the significance or cost attributed to each node within the network.