Minimum cost flow problem

The minimum cost flow problem involves determining the flow of goods through a network at the lowest possible cost. This network consists of nodes and arcs, where nodes represent points of supply, demand, or transit, and arcs represent the paths between these nodes along with their associated costs. In this problem, there are 8 nodes and 11 arcs. Each node has a certain amount of supply or demand, which is represented by the vector b. Positive values in this vector represent supply, while negative values represent demand. For example, node 1 has a supply of 2 units, while node 3 has a demand of 8 units.

The arcs between nodes are defined in the arcs matrix, where each row represents an arc with a start node and an end node. The corresponding costs for these arcs are listed in the costs vector. To solve this problem using YALMIP in MATLAB, we first define the decision variables for the flow on each arc using sdpvar. The objective function is to minimize the total cost, which is the dot product of the costs vector and the flow variables. The constraints for the problem ensure flow conservation at each node. For each node, the sum of the flow into the node minus the sum of the flow out of the node must equal the supply or demand at that node as specified in b. Additionally, the flow on each arc must be non-negative.

The problem is then solved using the Gurobi solver by specifying it in the sdpsettings. If a solution is found, the optimal flow values for each arc are displayed. If the problem is infeasible, a message indicating this is shown.In essence, this algorithm ensures that the flow through the network satisfies all supply and demand constraints while minimizing the total transportation cost.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b(i) \qquad i \in N \qquad (1) \\
& l_{ij} \le x_{ij} \le u_{ij} \qquad (i,j) \in A \qquad (2)
\end{aligned}
$$

```
Model fingerprint: 0xed341ceb
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [4e+00, 1e+01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+00, 8e+00]
Presolve removed 11 rows and 0 columns
Presolve time: 0.01s

Solved in 0 iterations and 0.01 seconds (0.00 work units)
Infeasible or unbounded model
Problem is infeasible.
```

The maximum flow problem

The maximum flow problem aims to determine the maximum flow possible from a source node to a sink node within a network, adhering to capacity constraints on the arcs. The provided MATLAB code utilizes YALMIP and Gurobi to solve this problem. Initially, the code adds the necessary paths for YALMIP and Gurobi to the MATLAB environment. It then defines the number of nodes and arcs in the network using `numNodes` and `numArcs`. The `arcs` matrix specifies the connections between nodes, adjusted to 1-based indexing, while the `capacities` vector outlines the maximum flow capacity for each arc. The `source` and `sink` variables denote the starting and ending nodes for the flow.

An incidence matrix `A` is constructed next, where each column corresponds to an arc and each row corresponds to a node. For each arc, the incidence matrix is populated to reflect the direction of flow, with +1 for the start node and -1 for the end node.The decision variables are then defined, where `x` represents the flow on each arc and `v` represents the total flow from the source to the sink, which is to be maximized. Non-negativity and capacity constraints are established to ensure the flow on each arc remains between 0 and the arc's capacity. Additionally, flow conservation constraints ensure that the flow entering each node equals the flow leaving it, except for the source, which has an outflow of `v`, and the sink, which has an inflow of `v`.

The objective is set to maximize the flow `v` by minimizing `-v`. The problem is solved using the Gurobi solver with specified options. If the solver finds a solution, it extracts and displays the maximum flow value `v` and the flow distribution across each arc. If there is an issue, the solver provides information about the encountered problem. Overall, this code effectively models and solves the maximum flow problem by defining the network structure, setting up the necessary constraints, and optimizing the flow using advanced solvers.

```
Solved in 0 iterations and 0.02 seconds (0.00 work units)
Optimal objective  0.000000000e+00
Maximum Flow:
     0

Flow Distribution:
     Arc       Flow
    _______    ____

    2    1      0
    1    4      0
    2    3      0
    2    6      0
    3    4      0
    3    5      0
    4    7      0
    5    6      0
    5    7      0
    6    8      0
    7    8      0
```