Engineering Optimization ISyE 3133
LP Model for Question 1

One of the key modeling issues in this question is how to represent flows of electric power, which are directed from one node to another, when a conduit is given as an indirected link between pairs of nodes.

There are two main approaches to addressing this issue.

One is to define a directed arc in each direction between any two nodes connected by a conduit. Whenever nodes i and j are connected by a conduit, the set of arcs should include the arcs (i, j) and (j, i), both of which have an upper bound on flow given by the capacity of the conduit and a lower bound of zero. For example, row 10 of the network data Excel file for the DS9 instance has entries 13, 28, 11, so both $(13, 28)$ and $(28, 13)$ should be included as arcs that permit flow within the domain $[0, 11]$.

An alternative approach is to choose an arbitrary order for two nodes joined by a conduit, and define a directed arc in that direction only. Whenever nodes i and j are connected by a conduit, the set of arcs should include exactly one of the arcs (i, j) and (j, i) with an upper bound on flow given by the capacity of the conduit and a lower bound given by the negative of the capacity of the conduit. For example, row 10 of the network data Excel file for the DS9 instance has entries 13, 28, 11, so either $(13, 28)$ or $(28, 13)$, but not both, should both be included as an arc that permits flow within the domain $[-11, 11]$.

We first given an LP using this second approach.

<u>Parameters</u>

- N = set of nodes in network given by data (i.e. $\{1, 2, \ldots, n\}$),
- A = set of directed arcs in network, one for each conduit given by data,
- c_a = maximum capacity of conduit a, one parameter for each $a \in A$,
- d_i = maximum demand of node i, one parameter for each $i \in N$.

<u>Variables</u>

- x_a = flow on arc a, one variable for each $a \in A$,
- y_i = demand satisfied for each node i, one variable for each $i \in N$.

LP Model

$$\begin{aligned} \max \quad & \sum_{i \in N} y_i \quad \text{or} \quad y_1 + \sum_{a \in \delta^+(1)} x_a, & \\ \text{s.t.} \quad & \sum_{a \in \delta^-(i)} x_a - \sum_{a \in \delta^+(i)} x_a = y_i & \forall i \in N \setminus \{1\}, \\ & -c_a \le x_a \le c_a, & \forall a \in A, \\ & 0 \le y_i \le d_i & \forall i \in N, \end{aligned}$$

where $\delta^+(i)$ and $\delta^-(i)$ are defined with respect to the network (N, A).

Comment Many teams took this approach but defined the variable x_{ij} for all $i, j = 1, \ldots, n$, where $n = 30$ for the DS9 data. If this approach is taken, then it is important that $c_{i,j}$ is defined to equal zero for any i and j not connected by a conduit, and for any $i = j$. It is also helpful if whenever i and j are connected by a conduit, one of the pair c_{ij} and c_{ji} is defined to be zero and the other is defined to be the capacity of the conduit. In other words, the capacity parameter c_{ij} is set to zero for each $i, j = 1, \ldots, 30$ with $(i, j) \notin A$. Alternatively, a constraint should be added to the model, stating $x_{ij} = 0$ for each $i, j = 1, \ldots, 30$ with $(i, j) \notin A$.

Comment Some teams fixed $y_1 = d_1$ explicitly. This is unnecessary and could cause problems in Question 2, since increasing fairness might require the demand met at node 1 to fall short.

It is possible to also have a Max Flow Problem formulation by using the first approach to modeling conduits, adding a dummy source node, 0, an arc $(0, 1)$ having effectively infinite capacity, a dummy sink node, $n + 1$, and an arc $(i, n + 1)$ having capacity d_i, for each $i \in \{1, 2 \ldots, n\}$. The Max Flow Problem parameters are thus defined as follows.

Parameters

- $\hat{N}$ = set of demand nodes from data plus dummy source and sink (i.e. $\{0, 1, 2, \ldots, n, n+1\}$),
- $\hat{A}$ = an arc in each direction for each conduit plus the arc $(0, 1)$ plus an arc from each demand node to the dummy sink,
- $c_{(i,j)}$ = maximum capacity of conduit between demand nodes i and j that are connected by a conduit, two parameters for each conduit,
- $c_{(0,1)}$ = an effectively infinite value, e.g. $\sum_{i=1}^{n} d_i$,
- $c_{(i,n+1)} = d_i$ is the capacity of the arc from demand node i to the dummy sink node, for each $i = 1, 2, \ldots, n$,
- source node is 1, and
- sink node is $n + 1$.

Variables

- x_a = flow on arc $a \in \hat{A}$, one variable for each $a \in \hat{A}$.

LP Model of Max Flow Problem

$$\begin{aligned}
\max \quad & \sum_{a \in \delta^+(0)} x_a \quad \text{or} \quad \sum_{a \in \delta^-(n+1)} x_a, & \\
\text{s.t.} \quad & \sum_{a \in \delta^-(i)} x_a - \sum_{a \in \delta^+(i)} x_a = 0 & \forall i \in \hat{N} \setminus \{0, n+1\}, \\
& 0 \leq x_a \leq c_a, & \forall a \in \hat{A},
\end{aligned}$$

where $\delta^+(i)$ and $\delta^-(i)$ are defined with respect to the network $(\hat{N}, \hat{A})$.

Comment The Python code solution, “project v4.py”, took this approach. A glossary showing how LP model elements are implemented in the Python code is given below.

LP Model	Python Code
n	`len(nodes)`
0	`source`
$n+1$	`sink`
$\hat{N} \setminus \{0, n+1\}$	`nodes`
d_i	`demands[i]`
$\hat{A}$	`arcs`
$c_{(i,j)}$	`capacity[i,j]`
$x_{(i,j)}$	`flow[i,j]`