Mathematical Programming
Programming Exercise Report 1

Mathias Hofer (01226806)

April 2018

1 Single-Commodity Flow

The model is based on the SCF MSTP formulation from the slides. A new binary variable y_i was added which basically indicates if a node is one of the chosen k nodes. This variable is then used to control the flow.

1.1 Model

Objective function

$$\min \Sigma_{e \in E} w_e x_e$$

Send out k commodities from the artificial root node

$$\text{s.t. } \Sigma_{(0,j) \in \delta^+(0)} f_{0j} = k$$

Each node in k-MST consumes 1 commodity and sends the rest out, hence the difference is -1, for all nodes not in k-MST the difference is 0.

$$\Sigma_{(i,j) \in \delta^+(i)} f_{ij} - \Sigma_{(j,i) \in \delta^-(i)} f_{ji} = -y_i \qquad \forall i \in V \setminus \{0\}$$

There is only a flow on selected edges

$$f_{ij} \leq k x_e \qquad \forall e = \{i,j\} \in E$$

As above, but since there is no backflow to the root, we can use $(k-1)$

$$f_{ji} \leq (k-1) x_e \qquad \forall e = \{i,j\} \in E$$

Select k edges, and therefore $k+1$ nodes, but with the artificial root node

$$\Sigma_{e \in E} x_e = k$$

If we select some edge (i,j) then y_i and y_j has to be 1

$$x_e \leq y_i \qquad \forall e = \{i,j\} \in E$$

$$x_e \leq y_j \qquad \forall e = \{i, j\} \in E$$

Select only one of the artificial 0-weight edges

$$\Sigma_{e \in \delta^+(0)} x_e \leq 1$$

Directed flow variable

$$f_{ij} \geq 0 \qquad \forall (i, j) \in A$$

Decision variable for the selected edges

$$x_e \in \{0, 1\} \qquad \forall e \in E$$

Decision variable for the selected nodes

$$y_i \in \{0, 1\} \qquad \forall i \in V$$

1.2 Results

	|V|	**K**	**Objective**	**Runtime in Seconds**	**B&B Nodes**
g01	10	2	46	0.00	0
		5	447	0.02	0
g02	20	4	373	0.05	0
		10	1390	0.08	0
g03	50	10	725	0.06	0
		25	3074	0.13	0
g04	70	14	909	0.19	0
		35	3292	0.41	0
g05	100	20	1235	0.28	0
		50	4898	1.25	0
g06	200	40	2068	23.67	7,960
		100	6705	70.67	49,206
g07	300	60	1335	29.83	4,810
		150	4534	984.41	640,579
g08	400	80	1620	165.20	64,825
		200	5787	4631.23	3,596,750

Table 1: Single-Commodity Flow Results on a Intel Core i7-5500U 2.4GHz

2 Multi-Commodity Flow

The model is based on the MCF MSTP formulation from the slides. A new binary variable y_i was added which basically indicates if a node is one of the chosen k nodes. This variable is then used to control the flow.

2.1 Model

Objective function

$$\min \Sigma_{e \in E} w_e x_e$$

Send out the commodities for the selected nodes

$$\text{s.t. } \Sigma_{(0,j) \in \delta^+(0)} f_{0j}^k = y_k \qquad \forall k \in V \setminus \{0\}$$

Each node in the k-MST receives its commodity, for all other nodes the difference is 0

$$\Sigma_{(i,j) \in \delta^+(i)} f_{ij}^k - \Sigma_{(j,i) \in \delta^-(i)} f_{ji}^k = \begin{cases} -y_k & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in V \setminus \{0\}, \forall i \in V \setminus \{0\}$$

There is only a flow on selected edges

$$f_{ij}^k \leq x_e \qquad \forall k \in V \setminus \{0\}, \forall e = \{i,j\} \in E$$

$$f_{ji}^k \leq x_e \qquad \forall k \in V \setminus \{0\}, \forall e = \{i,j\} \in E$$

Select k edges, and therefore $k+1$ nodes, but with the artificial root node

$$\Sigma_{e \in E} x_e = k$$

If we select some edge (i,j) then y_i and y_j has to be 1

$$x_e \leq y_i \qquad \forall e = \{i,j\} \in E$$

$$x_e \leq y_j \qquad \forall e = \{i,j\} \in E$$

Select only one of the artificial 0-weight edges

$$\Sigma_{e \in \delta^+(0)} x_e \leq 1$$

Directed flow variable

$$f_{ij}^k \geq 0 \qquad \forall k \in V \setminus \{0\}, \forall (i,j) \in A$$

Decision variable for the selected edges

$$x_e \in \{0,1\} \qquad \forall e \in E$$

Decision variable for the selected nodes

$$y_i \in \{0,1\} \qquad \forall i \in V$$

2.2 Results

	|V|	K	**Objective**	**Runtime in Seconds**	**B&B Nodes**
g01	10	2	46	0.01	0
		5	477	0.03	0
g02	20	4	373	0.06	0
		10	1390	0.16	0
g03	50	10	725	0.13	0
		25	3074	1.31	0
g04	70	14	909	1.55	0
		35	3292	6.78	0
g05	100	20	1226	1.42	0
		50	4839	31.50	103
g06	200	40	1927	10.14	0
		100	6529	1692.78	525
g07	300	60	-	-	-
		150	-	-	-
g08	400	80	-	-	-
		200	-	-	-

Table 2: Multi-Commodity Flow Results on a Intel Core i7-5500U 2.4GHz

3 Interpretation of Results

The SCF formulation outperforms the MCF one quite significantly. This is probably because it also generates significantly less columns. If we compare the g06 instances with k=100, it takes cplex only a fraction of the runtime to solve the SCF formulation compared to the MCF one. But comparing the Branch-and-Bound nodes, quite the opposite is the case. However, since the MCF formulation seems to be a bit buggy, a comparison between the two results have to be taken with care. We get the right results for MCF and instances g01-g04, but then get a wrong result for g05 and g06. Since the runtime increased drastically for the bigger instances, we did not get results for g07 and g08 for the MCF formulation.

If we put the focus on the bigger graphs and the SCF formulation, we see that the number of Branch-and-Bound nodes seems to explode. This is probably because we have introduced an additional binary variable, i.e. y_i, which must not be fractional. Hence, a different formulation which only depends on x_e as integer variables, would decrease the Branch-and-Bound nodes.

Another important general observation, which holds for both formulations, is that we often get a “good enough” result, i.e. with a gap to optimality of far less than 1%, quite early during the optimization process. Hence, for SCF and g08, we had an objective value with a gap to the optimum of less than 1% in 2-3 minutes, however, it took an additional hour to prove that the found solution was indeed optimal.