Course on mathematical modelling: AMPL and CPLEX

teacher: Giacomo Lanza

Dipartimento di Informatica, Università di Pisa

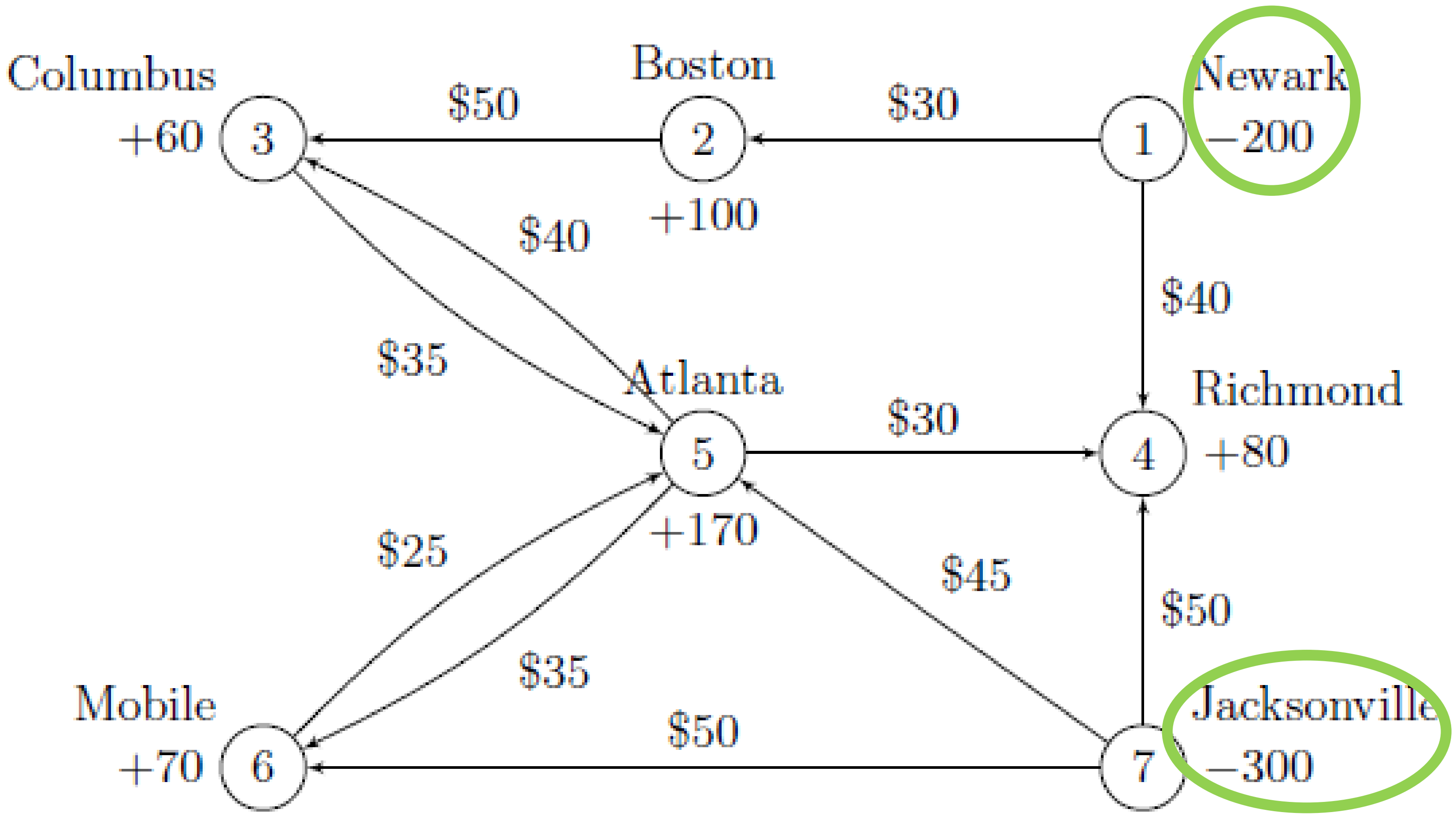
a.a. 2019-2020

An example of Minimum Cost Flow Problem

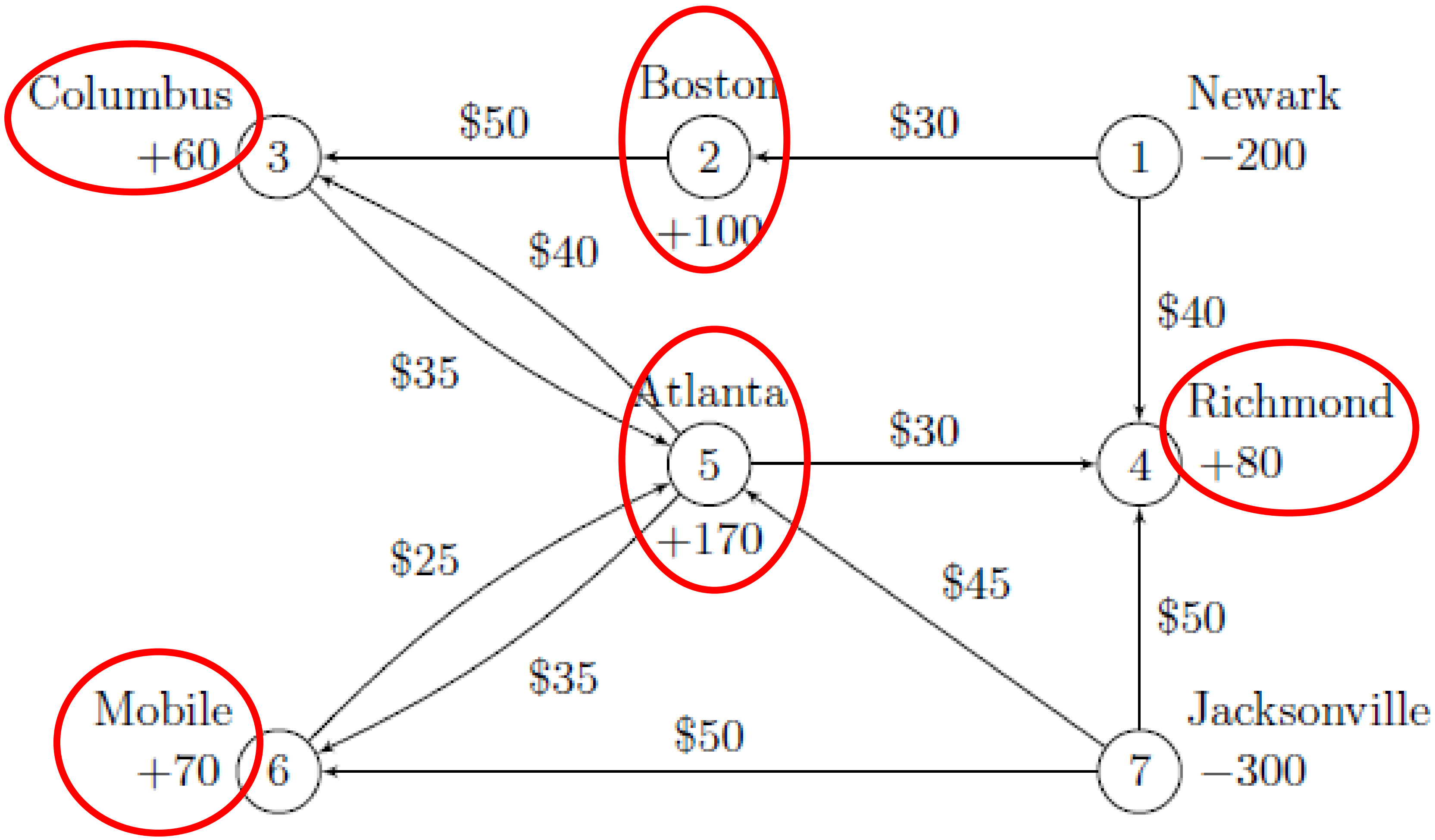
Bavarian Motor Company (BMC) manufactures luxury cars in Germany and exports them in the U.S.; they are currently holding 200 cars available at the port in Newark and 300 cars available at the port in Jacksonville. From there, the cars are transported (by rail or truck) to five distributors having a specific requirement of cars (see the figure). In the network, Newark and Jacksonville are supply nodes (or origins): negative numbers (e.g. -200) represent their supply; Boston, Columbus, Atlanta, Richmond and Mobile are demand nodes (or destinations): positive numbers (e.g. +100) represent their demand.

The problem is to determine how to transport (flowing) cars along the arcs of the network to satisfy the demands at a minimum cost.

An example of Minimum Cost Flow Problem



An example of Minimum Cost Flow Problem



An example of Minimum Cost Flow Problem

$$
\begin{aligned}
&\min \; 30x_{12} + 40x_{14} + 50x_{23} + 35x_{35} + 40x_{53} + 30x_{54} + \\
&\quad + 35x_{56} + 25x_{65} + 50x_{74} + 45x_{75} + 50x_{76} \\
&-x_{12} - x_{14} \geq -200 \\
&x_{12} - x_{23} = 100 \\
&x_{23} + x_{53} - x_{35} = 60 \\
&x_{14} + x_{54} + x_{74} = 80 \\
&x_{35} + x_{65} + x_{75} - x_{53} - x_{54} - x_{56} = 170 \\
&x_{56} + x_{76} - x_{65} = 70 \\
&-x_{74} - x_{75} - x_{76} \geq -300 \\
&x_{12}, x_{14}, \ldots \geq 0
\end{aligned}
$$

An example of Minimum Cost Flow Problem: Set Definition

- Set of **Cities**

```
# model
set Cities;
```

```
# data
set Cities:= Newark Jacksonville Boston Columbus Atlanta
Richmond Mobile;
```

An example of Minimum Cost Flow Problem: Set Definition

- Set of **Cities**
  - Subset of Origins and subset of Destinations

```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);
```

```
# data
set Cities:= Newark Jacksonville Boston Columbus Atlanta
Richmond Mobile;
set Origins:= Newark Jacksonville;
set Destinations:= Boston Columbus Atlanta Richmond Mobile;
```

Set operator **within** tests the membership of sets.
The expression "*A* **within** (*B*)" means that all the elements of set *A* are also elements of set *B*.
That is, set *A* is a subset of (or is the same set as) set *B.*

An example of Minimum Cost Flow Problem: Set Definition

- Set of **Link**

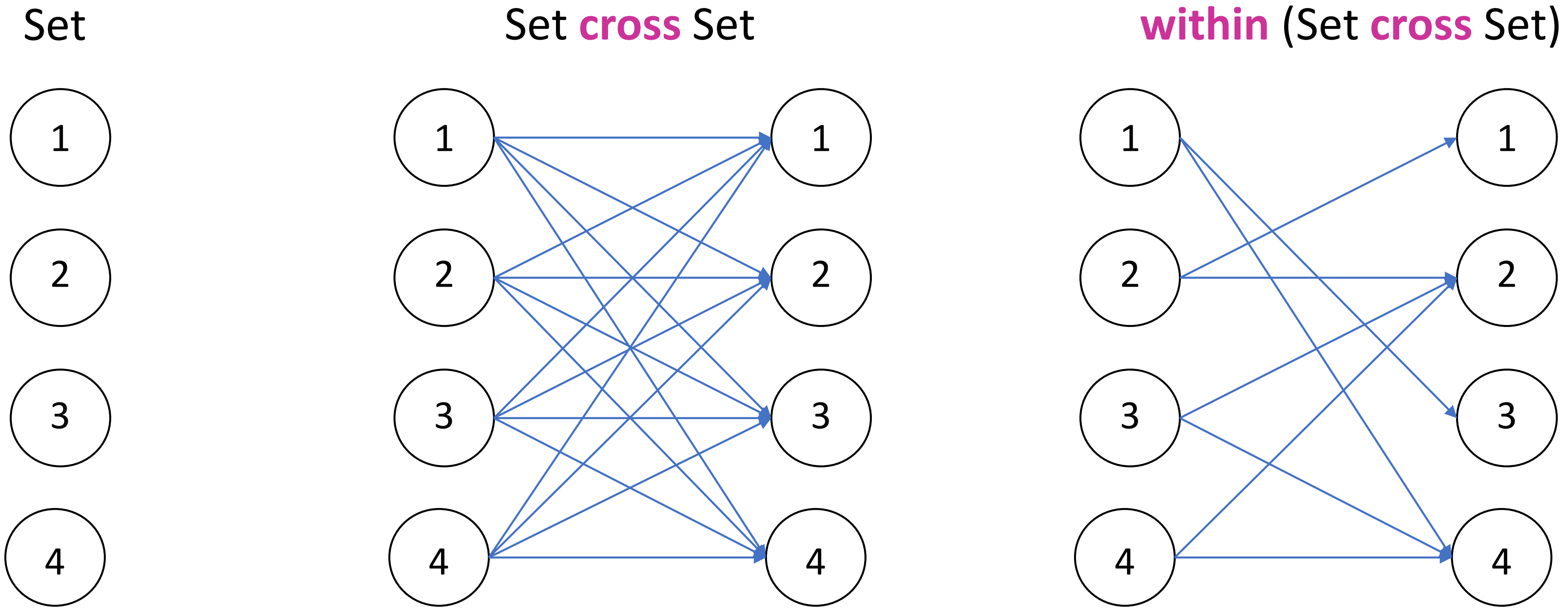
```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);

set Link  within (Cities cross Cities);
```

```
# data
set Cities:= Newark Jacksonville Boston Columbus Atlanta
Richmond Mobile;
set Origins:= Newark Jacksonville;
set Destinations:= Boston Columbus Atlanta Richmond Mobile;
```

An example of Minimum Cost Flow Problem: Set Definition

Set = {1, 2, 3, 4}



An example of Minimum Cost Flow Problem: Set Definition

- Set of **Link**

```
# data
set Link := (Jacksonville, Richmond)
(Jacksonville, Atlanta)
(Jacksonville, Mobile)
(Mobile, Atlanta)
(Atlanta, Mobile)
(Atlanta, Columbus)
(Atlanta, Richmond)
(Columbus, Atlanta)
(Boston, Columbus)
(Newark, Richmond)
(Newark, Boston);
```

```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);

set Link  within (Cities cross Cities);
```

Set operator **cross** is about the definition of sets.

The expression "*A* **cross** *B*" gives the set of all pairs (the cross or Cartesian product) of its arguments, which are both sets A and B. Thus the expression has the same meaning as the indexing expression A x B.

An example of Minimum Cost Flow Problem: Parameters Definition

```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);
set Link  within (Cities cross Cities);

param Costs {Link};
param DemSup {Cities};
```

```
# data
param Costs :=
Jacksonville, Richmond 50
Jacksonville, Atlanta 45
Jacksonville, Mobile 50
[ … ];

param DemSup :=
Jacksonville -300
Boston 100
[ … ];
```

An example of Minimum Cost Flow Problem: Variables Definition

The decision is how much product to ship from city *i* to city *j*

```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);
set Link  within (Cities cross Cities);

param Costs {Link};
param DemSup {Cities};

var Ship {Link} >= 0;
```

An example of Minimum Cost Flow Problem: Objective Definition

The objective is to *minimize the total cost* of transportation

```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);
set Link  within (Cities cross Cities);

param Costs {Link};
param DemSup {Cities};

var Ship {Link} >= 0;

minimize Total_Cost: sum {(i,j) in Link} Cost[i,j] * Ship[i,j];
```

An example of Minimum Cost Flow Problem: Constraint Definition

We only need to write flow conservation constraints

```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);
set Link  within (Cities cross Cities);
param Costs {Link};
param DemSup {Cities};
var Ship {Link} >= 0;

minimize Total_Cost: sum {(i,j) in Link} Costs[i,j] * Ship[i,j];
subject to Supply {i in Origins}: - sum {(i,k) in Link} Ship[i,k] >= DemSup[i];
subject to Demand {i in Destinations}:   sum {(j,i) in Link} Ship[j,i] - sum {(i,k) in Link} Ship[i,k] == DemSup[i];
```

An example of Minimum Cost Flow Problem: Model and Data Files

```
# model
set Cities;
set Origins  within (Cities);
set Destinations  within (Cities);
set Link  within (Cities cross Cities);
param Costs {Link};
param DemSup {Cities};
var Ship {Link} >= 0;

minimize Total_Cost: sum {(i,j) in Link} Costs[i,j] * Ship[i,j];
subject to Supply {i in Origins}: - sum {(i,k) in Link} Ship[i,k]
>= DemSup[i];

subject to Demand {i in Destinations}:   sum {(j,i) in Link}
Ship[j,i] - sum {(i,k) in Link} Ship[i,k] == DemSup[i];
```

```
data;
set Cities:= Newark Jacksonville [ … ] Mobile;
set Origins:= Newark Jacksonville;
set Destinations:= Boston Columbus [ … ] Mobile;
set Link := (Jacksonville, Richmond)
            [ … ]
            (Newark, Boston);

param Costs :=
Jacksonville, Richmond 50
[ … ];

param DemSup :=
Jacksonville -300
Boston 100
[ … ];
```

An example of Minimum Cost Flow Problem running scripts

```
# Console
ampl: include BMC.run;
CPLEX 12.6.1.0: optimal solution; objective 22350
7 dual simplex iterations (0 in phase I)
Ship :=
Atlanta      Columbus    40
Atlanta      Mobile       0
Atlanta      Richmond     0
Boston       Columbus    20
Columbus     Atlanta      0
Jacksonville Atlanta    210
Jacksonville Mobile      70
Jacksonville Richmond     0
Mobile       Atlanta      0
Newark       Boston     120
Newark       Richmond    80;
```

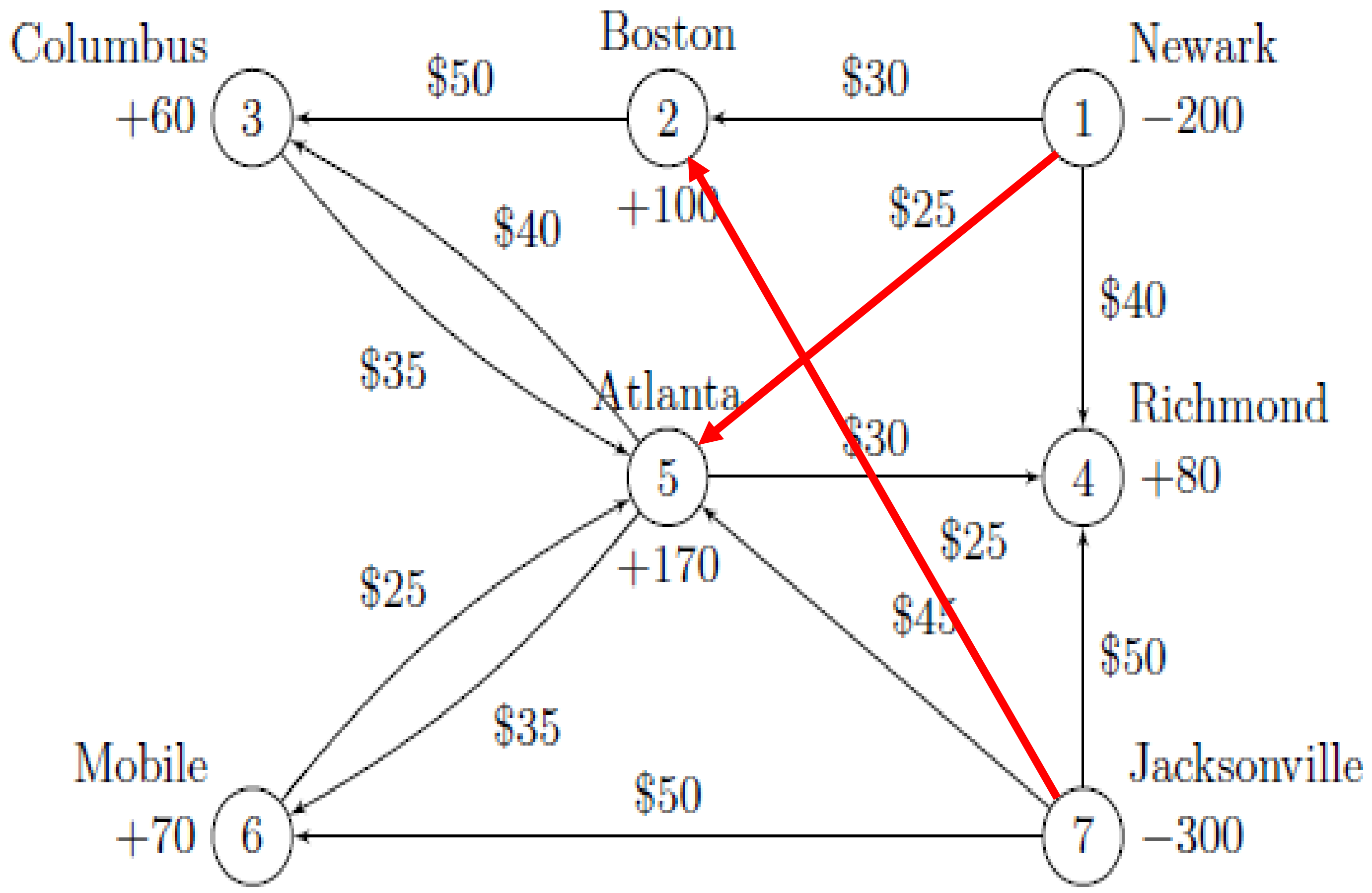
```
# ATM.run file
reset;
model BMC.mod;
data BMC.dat;
option solver cplexamp;
solve;
display Ship;
```

A constrained Minimum Cost Flow Problem

Considering the same problem as before, take into account the following additional constraints:

- Each link has a capacity, that cannot be exceeded (see the figure)
- Two additional links are available (Jacksonville-Boston and Newark-Atlanta), but BMC has to pay a (fixed) activation cost to use them (see the figure)
- By considering the demand of Boston fixed to 100, suppose that the number of cars allowed to pass through the city is limited to 30
- Columbus is allowed to increase its original demand (i.e. 60), but it has to pay 500 to BMC for any additional car that it will receive (with respect to the original demand)

A constrained Minimum Cost Flow Problem



Link	Capacity
Jacksonville, Richmond	90
Jacksonville, Atlanta	180
Jacksonville, Mobile	50
Newark, Boston	50
Newark, Richmond	90
Atlanta Columbus	50
All other links	100

Link	Activation Cost
Jacksonville, Boston	300
Newark, Atlanta	800

AMPL Main Commands:

- reset; # reset the environment
- model *modelfilename*.mod; # model upload
- data *datafilename*.dat; # data upload
- option solver *nameofsolver*; # optimizer selection
- solve; # solve
- display *nameofvariables*; # display variables