# Shortest route problem

Find the shortest route from the starting point (p1) to the ending point (p6).

![Graph of route problem]

## 1. Set optimization variables

For each "link" in the graph, we set one variable. For example, there’s a "link" from P1 to P2, so we set a variable x12. Notice that there are two variables associated with points P2 and P3, they are x23 and x32. If the value of a variable is 1, it means the "route" will pass through this link. Value 0 means this link will not be taken by the route.

```
In[1]:= vars = {x12, x13, x23, x32, x24, x25, x35, x54, x46, x56};
```

## 2. Set the objective function

We would like to minimize the total "distance" (cost) of the chosen route:

```
In[2]:= f = 15*x12 + 13*x13 + 9*x23 + 9*x32 + 11*x24 + 12*x25 + 16*x35 + 4*x54 + 17*x46 + 14*x56;
```

## 3. Set the constraints

To make sure the route is a connected path from the starting point to the ending point, we need to following constrains:

- (1) for each point other than the starting and the ending points, the total entering links (inflow)
should be equal to the total leaving links (outflow).

\[
g2 = x_{12} + x_{32} = x_{24} + x_{25} + x_{23};
g3 = x_{13} + x_{23} = x_{32} + x_{35};
g4 = x_{24} + x_{54} = x_{46};
g5 = x_{35} + x_{25} = x_{54} + x_{56};
\]

- (2) for the starting point, \( \text{out flow} - \text{inflow} = 1 \)

\[
g1 = x_{12} + x_{13} = 1;
\]

- (3) for the ending point, \( \text{inflow} - \text{outflow} = 1 \)

\[
g6 = x_{46} + x_{56} = 1;
\]

- (4) we also need to set all variables greater than or equal to 0

\[
\text{NonNegativeness} = \text{And} @@ \text{Thread}[vars \geq 0]
\]

\[
x_{12} \geq 0 \& \& x_{13} \geq 0 \& \& x_{23} \geq 0 \& \& x_{32} \geq 0 \& \& x_{24} \geq 0 \& \& x_{25} \geq 0 \& \& x_{35} \geq 0 \& \& x_{54} \geq 0 \& \& x_{46} \geq 0 \& \& x_{56} \geq 0
\]

4. Solve the problem

- (1) First, we can use the command "Minimize", which accepts equations/inequalities as inputs

\[
\text{Minimize}[[f, g1 \& \& g2 \& \& g3 \& \& g4 \& \& g5 \& \& g6 \& \& \text{NonNegativeness}], vars]
\]

\[
\{41, \{x_{12} \rightarrow 1, x_{13} \rightarrow 0, x_{23} \rightarrow 0, x_{32} \rightarrow 0, x_{24} \rightarrow 0, x_{25} \rightarrow 0, x_{35} \rightarrow 0, x_{54} \rightarrow 0, x_{46} \rightarrow 0, x_{56} \rightarrow 1\}\}
\]

The above solution tells us that the shortest route is \( P_1 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \), and the total cost is 41.

- (2) \textit{Mathematica} also provides a function "LinearProgramming", which is more efficient. However, it only accepts matrices/vectors in the standard LP form as inputs. Therefore, we first need to derive the coefficients in the standard form minimizing \( f = c^T x \), subject to constraints \( Ax \geq b, x \geq 0 \). Notice that \textit{Mathematica} uses \( Ax \geq b \), different from the \( Ax=b \) in the textbook. Hence when we feed the command with inputs, we have to specify for each RHS value \( b_i \) that it is an exact "equal". This can be done by setting a matrix of the form \( \{b_1, 0\}, \{b_2, 0\}, \ldots \{b_k, 0\} \).

\[
c = \text{Normal}[\text{CoefficientArrays}[f, \text{vars}]][[2]]
\]

\[
\{15, 13, 9, 9, 11, 12, 16, 4, 17, 14\}
\]
In[12]:= \[\text{A} = \text{Normal[\text{CoefficientArrays\{g1, g2, g3, g4, g5, g6\}, \text{vars}\} [[2]]];}
\text{MatrixForm[A]}\]

Out[13]/MatrixForm=
\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{pmatrix}
\]

In[14]:= \[\text{b} = \{\{1, 0\}, \{0, 0\}, \{0, 0\}, \{0, 0\}, \{0, 0\}, \{1, 0\}\}\]

Out[14]= \{\{1, 0\}, \{0, 0\}, \{0, 0\}, \{0, 0\}, \{0, 0\}, \{1, 0\}\}

In[15]:= \text{LPSol = LinearProgramming[c, A, b]}\]

Out[15]= \{1, 0, 0, 0, 0, 1, 0, 0, 0, 1\}

In[16]:= \text{LinearProgramming[c, A, b, Method \to \"Simplex\"]}\]

Out[16]= \{1, 0, 0, 0, 0, 1, 0, 0, 0, 1\}

In[17]:= \text{LinearProgramming[c, A, b, Method \to \"RevisedSimplex\"]}\]

Out[17]= \{1, 0, 0, 0, 0, 1, 0, 0, 0, 1\}

In[18]:= \text{LinearProgramming[c, A, b, Method \to \"InteriorPoint\"]}\]

\text{LinearProgramming::lpipp :}
\text{Warning: Method \to \text{InteriorPoint} specified for non-machine-precision problem. A machine-precision result will be given. If a non-machine-precision result is needed, set the option to Method \to \text{Simplex}.}


Again, the above results indicate that the Shortest route is \(x_{12}=1, x_{25}=1, x_{56}=1\). The cost is

In[19]:= \text{f /. Thread[\{x_{12}, x_{13}, x_{23}, x_{32}, x_{24}, x_{25}, x_{35}, x_{54}, x_{46}, x_{56}\} \to \text{LPSol}\]}\]

Out[19]= 41