Minimum Cost Flow Problems


Outline
- Formulation
- Examples
- Properties of the problem
- Network simplex algorithm
- Special cases of minimum cost flow problems

1. FORMULATION

Given a network \( G = (N, A) \) where \( N \) is the set of nodes \( i \) and \( A \) is the set of arcs \( (i,j) \). Let

Parameters
- \( c_{ij} \) = Per unit cost on arc \( (i,j) \)
- \( u_{ij} \) = Capacity on arc \( (i,j) \)
- \( b_i \) = Supply to node \( i \) (if \( b_i < 0 \), then there is a required demand)

Decision variables
- \( x_{ij} \) = Number of units of flow on the arc \( (i,j) \)

The minimum cost flow problem can be written as:

\[
\begin{align*}
\text{min} \quad & \sum_{j} c_{ij} x_{ij} \\
\text{s.t} \quad & \sum_{j} x_{ij} - \sum_{k} x_{jk} = b_i \\
& 0 \leq x_{ij} \leq u_{ij}
\end{align*}
\] (1)

The set of constraints (1) is called the set of flow conservation constraints, which is very standard for network flow problems. It says that the flow going into a node must be equal to the flow going out of a node. Consider the example below:

![Diagram of a network flow problem with nodes and arcs labeled with costs and capacities.](image-url)
We can write

\[
\begin{align*}
\min & \quad 3x_{12} + 2x_{13} + 2x_{23} + 3x_{24} + 2x_{25} + 4x_{35} + 3x_{46} + 5x_{54} + 4x_{56} \\
\text{s.t.} & \quad x_{12} + x_{13} - x_{12} + x_{23} - x_{23} - x_{24} + x_{25} + x_{35} - x_{35} - x_{25} - x_{24} + x_{46} - x_{46} - x_{46} - x_{54} + x_{54} + x_{54} + x_{56} = 9 \\
& \quad -x_{13} - x_{23} + x_{24} + x_{25} + x_{35} + x_{46} - x_{54} + x_{54} = 0 \\
& \quad x_{12} \leq 8, \quad x_{13} \leq 3, \quad x_{23} \leq 3, \quad x_{24} \leq 8, \quad x_{25} \leq 2, \quad x_{35} \leq 3, \quad x_{46} \leq 5, \quad x_{54} \leq 4, \quad x_{56} \leq 6
\end{align*}
\]

and all variables are non-negative.

There are 9 variables and 15 constraints.

Question: This is an LP with equality constraints. If we solve this problem by simplex method using simplex tableau, what is the size of a tableau?

The coefficient matrix of this linear program is called the **node-arc incidence matrix** where each row corresponds to a node and each column corresponds to an arc:

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
2 & -1 & 1 & 1 & 1 \\
3 & -1 & -1 & 1 \\
4 & -1 & 1 & -1 \\
5 & -1 & -1 & 1 & 1 \\
6 & -1 & -1 & -1
\end{bmatrix}
\]

In the matrix form, we can write the minimum cost flow problem as:

\[
\begin{align*}
\max & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \leq u \\
& \quad x \geq 0
\end{align*}
\]
2. An example

Optimal loading of a hopping airplane
A small commute airline uses a plane, with a capacity to carry at most p passengers, on a “hopping flight,” shown below. The hopping flight visits the cities 1, 2, 3, …, n, in a fixed sequence. The plane can pick up passengers at any node and drop them off at any other node. Let $b_{ij}$ denote the number of passengers available at node $i$ who want to go to node $j$, and let $f_{ij}$ denote the fare per passenger from node $i$ to node $j$. The airline would like to determine the number of passengers that the plane should carry between the various origins to destinations in order to maximize the total fare per trip while never exceeding the plane capacity. Show how to formulate this problem as a minimum cost flow problem.
3. PROPERTIES

Let us now look at some properties of the problem:

**Question 1:** What is the column in $A$ corresponding to a variable $x_{ij}$?

**Question 2:** Are the set of flow conservation constraints linearly independent?

**Question 3:** How many of them are linearly independent?

**Question 4:** Consider the columns formed by the arcs of a cycle. Are the columns linearly independent?

**Question 5:** Does the basis have any special structure?
Answer for Q1: 
\[
\begin{bmatrix}
0 \\
1 \\
\vdots \\
-1 \\
\vdots \\
0
\end{bmatrix}
\]

Answer for Q2: No. We multiply each constraint by 1 and then add them together. This linear combination of equations (with non-zero multipliers) produce a constraint where LHS is 0 and RHS is 0. Why does it happen? In the network flow problem, the total supply is equal to the total demand.

Answer for Q3: #constraints – 1. Why? Each column has exactly 2 non-zero elements. Suppose we delete a row from the set of constraints. There exists a column with only one non-zero element. This element cannot be obtained by any linear combination of other numbers.

Answer for Q4: NO! Can use an example to illustrate.

Answer for Q5: The basis has n-1 linearly independent columns (each column corresponds to an arc). The arcs for these n-1 independent columns cannot form any cycle. What is the definition for a graph with n nodes that is acyclic and has n-1 arcs? The basis is a spanning tree!

4. NETWORK SIMPLEX METHOD

Recall the basic steps involved in the simplex method include:

0) Begin with some basic feasible solution.
1) Compute the complementary dual solution, that is, a dual solution that satisfies complementary slackness.
2) Compute the reduced costs for the non-basic variables. Determine the entering variable, if any. Otherwise, an optimal solution is found.
3) For the given entering variable, find the leaving variable.
4) Update the set of basic and non-basic variables (that is, update the “basis”)
5) Go to 2).
The next table summarizes the steps of both the simplex method and the network simplex method.

<table>
<thead>
<tr>
<th>Simplex method</th>
<th>Network simplex method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0. Get an initial solution</strong></td>
<td><strong>0. Get an initial basic tree</strong></td>
</tr>
<tr>
<td><strong>1. Compute duals</strong>&lt;br&gt;Find $y$ by solving $y^T B = c_B^T$</td>
<td><strong>1. Compute duals</strong>&lt;br&gt;For each basis arc, we have $y_i - y_j = c_{ij}$</td>
</tr>
<tr>
<td><strong>2. Compute reduced costs</strong>&lt;br&gt;Get $\overline{c}_j = c_j - y^T A_j$ for non-basic variable $x_j$&lt;br&gt;If $x_j = 0$ and $\overline{c}_j &lt; 0$, then $x_j$ can enter&lt;br&gt;If $x_j = u_j$ and $\overline{c}_j &gt; 0$, then $x_j$ can enter&lt;br&gt;Let $x_s$ be the entering variable.&lt;br&gt;If there is no entering variable, optimal solution is found.</td>
<td><strong>2. Compute reduced costs</strong>&lt;br&gt;For a non-basic arc, we compute $\overline{c}<em>{ij} = c</em>{ij} - y_i + y_j$&lt;br&gt;If $x_{ij} = 0$ and $\overline{c}<em>{ij} &lt; 0$, then $x</em>{ij}$ can enter&lt;br&gt;If $x_{ij} = u_{ij}$ and $\overline{c}<em>{ij} &gt; 0$, then $x</em>{ij}$ can enter&lt;br&gt;If there is no entering variable, optimal solution is found.</td>
</tr>
<tr>
<td><strong>3. Find leaving variable (by ratio test)</strong>&lt;br&gt;Find $d$ by solving $d = B^{-1} A_s$&lt;br&gt;Find $r$ such that&lt;br&gt;$$r = \arg \min \left{ \left( \frac{x_j}{d_j}, d_j &gt; 0 \right), \left( \frac{u_j - x_j}{-d_j}, d_j &lt; 0 \right) \right}$$</td>
<td><strong>3. Find leaving variable (by flow augmenting cycle)</strong>&lt;br&gt;Adding the entering arc to the tree creates a unique cycle.&lt;br&gt;Find the maximum change of the arc flow along the cycle. (Note: think of finding the minimum residual capacity of a cycle)&lt;br&gt;The arc that blocks the change is the leaving variable.</td>
</tr>
<tr>
<td><strong>4. Update $B$</strong>&lt;br&gt;Replace the column for the leaving variable by the entering variable in the basis.</td>
<td><strong>4. Update $B$</strong>&lt;br&gt;“Augment” the cycle.&lt;br&gt;Update the basic spanning tree by adding the entering arc removing the leaving arc</td>
</tr>
<tr>
<td><strong>5. Repeat 1-4</strong></td>
<td><strong>5. Repeat 1-4</strong></td>
</tr>
</tbody>
</table>
Explanation of the steps of the network simplex method:

For our example network flow problem, assume that we are given a spanning tree: \{(1, 2), (1, 3), (2, 4), (3, 5), (5, 6)\}, and a set of arcs whose flow are at arc capacities \{(3, 5), (4, 5)\}.

The solid lines represent the basic variables and the dashed lines represent the non-basic variables that hit the capacity. The arcs represent the non-basic variables that have values of zero are not shown. The numbers shown next to the arcs are the flow on these arcs. Notice that the arcs corresponding to the basic variables form a tree.

1) Solve \( y^T B = c_y^T \) for \( y \).

For a basic variable \( x_{ij} \), we have
\[
\begin{bmatrix}
1 \\
-1
\end{bmatrix} y_j = \begin{bmatrix}
\cdots & y_i & \cdots & y_j & \cdots
\end{bmatrix}
\]

In our example, we have 5 equations and 6 unknowns. We need to set one to 0 (say \( y_1 = 0 \)). Then from the tree, we can compute the other values very easily:

\[
y_2 = y_1 - c_{12} = -3 \quad y_4 = y_2 - c_{24} = -8
\]
\[
y_3 = -2 \quad y_5 = -6
\]
2) For non-basic variables, compute
\[
\bar{c}_y = c_y - (y^TN)_y
\]
\[
= c_y - \begin{bmatrix}
\vdots & y_i & \cdots & y_j & \cdots \\
1 & & & & \\
-1 & & & & \\
\vdots & & & & 
\end{bmatrix}
\]
\[
= c_y - y_i + y_j
\]
For a variable at its lower bound, we can increase it if \(\bar{c}_y < 0\) (saving cost)
For a variable at its upper bound, we can decrease it if \(\bar{c}_y > 0\) (saving cost)
In our example, the non-basic variables are:

<table>
<thead>
<tr>
<th>At lower bounds</th>
<th>Reduced cost</th>
<th>Candidate for entering variable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arc (2,3)</td>
<td>(\bar{c}_{23} = 2 + 3 - 2 = 3)</td>
<td>No</td>
</tr>
<tr>
<td>Arc (5,4)</td>
<td>(\bar{c}_{54} = 5 + 6 - 6 = 5)</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>At upper bounds</th>
<th>Reduced cost</th>
<th>Candidate for entering variable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arc (2,5)</td>
<td>(\bar{c}_{25} = 2 + 3 - 6 = -1)</td>
<td>No</td>
</tr>
<tr>
<td>Arc (4,6)</td>
<td>(\bar{c}_{46} = 3 + 8 - 10 = 1)</td>
<td>YES!</td>
</tr>
</tbody>
</table>

Increase the flow on arc (4,6) will increase the total cost. That is, decrease the flow on arc (4,6) will decrease the total cost. Thus, \(x_{46}\) is the entering variable.

3) Find out the changes of the basic variables as a result of the change of the entering variable.
(In the revised simplex method, this is equivalent to finding the coefficients of the entering column \(d = B^{-1}A_{\text{new}}\))
Suppose that we decrease the flow of \(x_{46}\) by \(t\) (that is \(x_{46} = x_{46} - t\)). From the theory of the simplex method, we know that only some of the basic variables will be affected. From our knowledge of the network where the basic variables form a tree, we know that adding an entering variable to the tree will form a cycle. The effect of the change will all be reflected in the cycle:
Look at the net change (in cost): decrease flow on arcs (1,2), (2,4), (4,6) (cost savings is 3 + 5 + 3 = 11) and increase flow on arcs (1,3), (3,5), (5,6) (cost increase is 2 + 4 + 4 = 10). The net cost savings is 1 (which has already indicated by the reduced cost $\bar{c}_{46}$!)

4) Determine the leaving variable
   (In the simplex method, we use the smallest ratio to find out the blocking variable)
   After the change, all flow should still be within the bounds. Thus, we should have

   \[7-t \geq 0\]
   \[5-t \geq 0\]
   \[5-t \geq 0\]
   \[2+t \leq 3\]
   \[2+t \leq 3\]
   \[4+t \leq 6\]

   We can see that the maximum change is $t = 1$. The arcs (1,3) and (3,5) block the change of flow. We can take arc (3,5) as the leaving arc (that is, $x_{35}$ as the leaving variable).

5) Change the flow on the cycle and remove the leaving arc:

Go to step 1).
Next iteration:

1) Compute the dual prices:

\[ y_2 = y_1 - c_{12} = -3 \quad y_4 = y_2 - c_{24} = -8 \]

\[ y_1 = 0 \]

\[ y_3 = -2 \]

\[ y_5 = -11 + 4 = -7 \]

\[ y_6 = -8 - 3 = -11 \]

2) Find the entering variable:

<table>
<thead>
<tr>
<th>Arc</th>
<th>Reduced cost</th>
<th>Candidate for entering variable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2,3)</td>
<td>( \bar{c}_{23} = 2 + 3 - 2 = 3 )</td>
<td>No</td>
</tr>
<tr>
<td>(5,4)</td>
<td>( \bar{c}_{54} = 5 + 7 - 8 = 4 )</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arc</th>
<th>Reduced cost</th>
<th>Candidate for entering variable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2,5)</td>
<td>( \bar{c}_{25} = 2 + 3 - 7 = -2 )</td>
<td>No</td>
</tr>
<tr>
<td>(3,5)</td>
<td>( \bar{c}_{35} = 4 + 2 - 7 = -1 )</td>
<td>No</td>
</tr>
</tbody>
</table>

There is no candidate for the entering variable! The optimal solution is found.
Example:

Minimum cost flow problem
Minimum cost flow problem
How to get an initial basic feasible solution?

In real world applications of using minimum cost flow problem, we can usually find some obvious feasible solution of the problems and use it as a starting point. In case we do not have an obvious initial basic feasible solution, we can use the concept of Big-M idea as follows:

a) Create an artificial node with an external supply of 0.
b) From each supply node, create an arc to the artificial node. The arc cost is M and there is no arc capacity. Assign all supply to this newly created arc.
c) To each demand node, create an arc from the artificial node to this demand node. The arc cost is M and there is no arc capacity. The demand on this newly created arc is the amount of demand to the corresponding demand node.
d) Treat the transshipment nodes (with supply of 0) either as supply nodes or demand nodes.
e) Use the flow on these artificial arcs as an initial basic feasible solution (how can we ensure that it is basic). Solve the modified problem with the network simplex algorithm. If there is some flow on the artificially created arcs, there is no feasible solution to the original problem!

5. SOME SPECIAL CASES OF THE MINIMUM COST FLOW PROBLEMS

Transportation problem

There are m origins that contain various amounts of a commodity that must be shipped to n destinations to meet demand requirements.
Nodes are partitioned into two sets of nodes and there are arcs from nodes in set 1 to nodes in set 2. There is a cost associated with each arc and usually there is no arc capacity. Furthermore, the total supply and the total demand are the same (otherwise, there is no feasible solution since the flow conservation constraints are violated). If the total supply and the total demand are not the same, we can add dummy nodes. For example, if we have more supply than demand and the difference is the amount $V$, then we create a dummy demand node that has a demand of $V$. For each supply node, we create an arc to this dummy demand node with a cost of 0. If at the optimal solution, supply node $i$ has a flow of 4 to this dummy node, then it means that 4 units at node $i$ are not shipped. To solve a transportation problem, we can use the network simplex method. The steps can be further modified since we do not have arc capacity constraints.

**Assignment problem**

The assignment problem is a special case of the transportation problem that the supply and the demand for the nodes are 1. Such a problem is also called the *marriage* problem or (more mathematical oriented) the *bipartite* matching. Think about a group of men and a group of women available. For a person, he or she has multiple choices of possible spouse although different persons have different values to him or her. Our goal is to make the overall of 'happiness' the highest possible. (It is easy to see that someone may need to give up his or her first choice.) To solve it as a minimum cost flow problem, we represent the problem as follows:

There are also other very efficient methods for solving the assignment problem. One of them is called the Hungarian Method which is a method based on both the primal and dual of the assignment problem.

**Maximal flow problem**

We are determining the maximal flow possible from one given source node (say $s$) to a sink node (say $t$) under arc capacity constraints. Typical examples of this problem are finding the maximum
number of people can be evacuated during a fire in the building within a short period of time, and
determine the pipeline system capacity (think about the water supply facility).

The Linear Programming formulation of the problem is given as:

\[
\begin{align*}
\text{max} \quad & V \\
\text{s.t.} \quad & \sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} 0 & i \neq s, t \\ V & i = s \\ -V & i = t \end{cases} \\
& x_{ij} \leq u_{ij} \\
& x_{ij} \geq 0 
\end{align*}
\]

Q: Is this LP in the format of a minimum cost flow problem?

Q: How can we solve it as a minimum cost flow problem?

After we move the variable V from right to left in the formulation, we get:

\[
\begin{align*}
\text{min} \quad & -V \\
\text{s.t.} \quad & \sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} 0 & i \neq s, t \\ V & i = s \\ -V & i = t \end{cases} \\
& x_{ij} \leq u_{ij} \\
& x_{ij} \geq 0 
\end{align*}
\]
Let us create a new arc from \( t \) back to \( s \). Let the flow of this arc be \( x_{ts} \). We can see that this flow is simply the amount of \( V \) in the linear program. In other words, the graph for maximum flow problem can be rewritten as a minimum cost flow problem as:

![Graph](image)

Cost = -1, no arc capacity

Shortest path problem

The problem is to find the shortest distance from a source node to any other nodes in the network.

We now look at how to represent a shortest path problem as a network flow problem. Graphically, the problem is

![Graph](image)

To state it mathematically, we can write

\[
\begin{align*}
\min & \quad \sum_{j} c_{ij} x_{ij} \\
\text{s.t.} & \quad \sum_{j} x_{ij} - \sum_{k} x_{jk} = \begin{cases} 
  n & \text{if } i = s \\
  -1 & \text{if } i \neq s 
\end{cases}, \\
& \quad x_{ij} \geq 0
\end{align*}
\]