TRANSPORTATION AND ASSIGNMENT PROBLEMS

Transportation and assign. problems

- Two important types of linear programming problems
- **Transportation problems:**
  - many (not all) applications involve determining how to optimally transport goods
  - production scheduling
- **Assignment problems:**
  - a special type of transportation problems
  - Involves such applications as assigning people to tasks
- Both problems have *network* representations and are special cases of the *minimum flow cost problem* (see next chapter). 
Transportation problem

- **Example**
  - P&T Company produces canned peas.
  - Peas are prepared at three canneries (Bellingham, Eugene and Albert Lea).
  - Shipped by truck to four distributing warehouses (Sacramento, Salt Lake City, Rapid City and Albuquerque). 300 truckloads to be shipped.
  - **Problem:** minimize the total shipping cost.

P&T company problem
Shipping data for P&T problem

<table>
<thead>
<tr>
<th>Warehouse</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannery 1</td>
<td>464</td>
<td>513</td>
<td>654</td>
<td>867</td>
<td>75</td>
</tr>
<tr>
<td>Cannery 2</td>
<td>352</td>
<td>416</td>
<td>690</td>
<td>791</td>
<td>125</td>
</tr>
<tr>
<td>Cannery 3</td>
<td>995</td>
<td>682</td>
<td>388</td>
<td>685</td>
<td>100</td>
</tr>
</tbody>
</table>

Allocation: 80 65 70 85 300

Network representation
Formulation of the problem

minimize \[ Z = 464x_{11} + 513x_{12} + 654x_{13} + 867x_{14} + 352x_{21} + 416x_{22} \]
\[ + 690x_{23} + 791x_{24} + 995x_{31} + 682x_{32} + 388x_{33} + 685x_{34} \]

subject to
\[ x_{11} + x_{12} + x_{13} + x_{14} = 75 \]
\[ x_{21} + x_{22} + x_{23} + x_{24} = 125 \]
\[ x_{31} + x_{32} + x_{33} + x_{34} = 100 \]
\[ x_{11} + x_{21} + x_{31} = 80 \]
\[ x_{12} + x_{22} + x_{32} = 65 \]
\[ x_{13} + x_{23} + x_{33} = 70 \]
\[ x_{14} + x_{24} + x_{34} = 85 \]

and \[ x_{ij} \geq 0 \quad (i = 1, 2, 3; \ j = 1, 2, 3, 4) \]

Constraints coefficients for P&T Co.

Coefficient of:

\[
A = \begin{bmatrix}
x_{11} & x_{12} & x_{13} & x_{14} & x_{21} & x_{22} & x_{23} & x_{24} & x_{31} & x_{32} & x_{33} & x_{34} \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

\{ Cannery constraints \}
\{ Warehouse constraints \}
Transportation problem model

- **Transportation problem**: distributes *any* commodity from *any* group of *sources* to any group of *destinations*, *minimizing the total distribution cost*.

<table>
<thead>
<tr>
<th>Prototype example</th>
<th>General problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truckload of canned peas</td>
<td>Units of a commodity</td>
</tr>
<tr>
<td>Three canneries</td>
<td><em>m</em> sources</td>
</tr>
<tr>
<td>Four warehouses</td>
<td><em>n</em> destinations</td>
</tr>
<tr>
<td>Output from cannery <em>i</em></td>
<td>Supply <em>s</em>_i from source <em>i</em></td>
</tr>
<tr>
<td>Allocation to warehouse <em>j</em></td>
<td>Demand <em>d</em>_j at destination <em>j</em></td>
</tr>
<tr>
<td>Shipping cost per truckload from cannery <em>i</em> to warehouse <em>j</em></td>
<td>Cost <em>c</em>_ij per unit distributed from source <em>i</em> to destination <em>j</em></td>
</tr>
</tbody>
</table>

- Each source has a certain *supply* of units to distribute to the destinations.
- Each destination has a certain *demand* for units to be received from the source.
- **Requirements assumption**: Each source has a fixed *supply* of units, which must be entirely distributed to the destinations. Similarly, each destination has a fixed *demand* for units, which must be entirely received from the sources.
Transportation problem model

The feasible solutions property: a transportation problem has feasible solution if and only if

\[ \sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j \]

If the supplies represent maximum amounts to be distributed, a dummy destination can be added.
Similarly, if the demands represent maximum amounts to be received, a dummy source can be added.

The cost assumption: the cost of distributing units from any source to any destination is \textit{directly proportional} to the number of units distributed.
Thus, this cost is the unit cost of distribution \textit{times} the number of units distributed.

Integer solution property: for transportation problems where every \( s_i \) and \( d_j \) have an integer value, all basic variables in every BF solution also have integer values.
### Parameter table for transp. problem

<table>
<thead>
<tr>
<th></th>
<th>Destination</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$c_{11}$</td>
<td>$s_1$</td>
</tr>
<tr>
<td>2</td>
<td>$c_{12}$</td>
<td>$s_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$m$</td>
<td>$c_{m1}$</td>
<td>$s_m$</td>
</tr>
<tr>
<td>Demand</td>
<td>$d_1$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$d_2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$d_n$</td>
<td></td>
</tr>
</tbody>
</table>

### Transportation problem model

- **The model:** any problem fits the model for a transportation problem if it can be described by a parameter table and if it satisfies both the requirements assumption and the cost assumption.

- The objective is to minimize the total cost of distributing the units.

  - *Some problems that have nothing to do with transportation can be formulated as a transportation problem.*
Formulation of the problem

- \( Z \): total distribution cost
- \( x_{ij} \): number of units distributed from source \( i \) to destination \( j \)

\[
\text{minimize} \quad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij},
\]

subject to

\[
\sum_{j=1}^{n} x_{ij} = s_i, \quad \text{for } i = 1, 2, \ldots, m,
\]

\[
\sum_{i=1}^{m} x_{ij} = d_j, \quad \text{for } j = 1, 2, \ldots, n,
\]

and \( x_{ij} \geq 0, \) for all \( i \) and \( j \).
Coefficient constraints

Coefficient of:

\[ \begin{array}{cccccccc}
1 & 1 & \cdots & 1 \\
1 & 1 & \cdots & 1 \\
& & \ddots & \\
1 & 1 & \cdots & 1 \\
\end{array} \]

Supply constraints

Demand constraints

Example: solving with Excel

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>P&amp;T Co. Distribution Problem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit Cost</th>
<th>Destination (Warehouse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Bellingham</td>
</tr>
<tr>
<td>6</td>
<td>544</td>
</tr>
<tr>
<td>7</td>
<td>357</td>
</tr>
<tr>
<td>8</td>
<td>195</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shipment Quantity</th>
<th>Destination (Warehouse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Truckloads)</td>
<td>Sacramento</td>
</tr>
<tr>
<td>Source</td>
<td>Bellingham</td>
</tr>
<tr>
<td>(Cannery)</td>
<td>Eugene</td>
</tr>
<tr>
<td>Albert Lee</td>
<td>0</td>
</tr>
<tr>
<td>Total Received</td>
<td>60</td>
</tr>
<tr>
<td>Total Shipped</td>
<td>75</td>
</tr>
<tr>
<td>Total Cost</td>
<td>$182,515</td>
</tr>
</tbody>
</table>

Supply constraints

Demand constraints
Example: solving with Excel

Solver Parameters

Set Target Cell: TotalCost
Equal To: Max
By Changing Cells:

Solver Options

- Assume Linear Model
- Assume Non-Negative

Subject to the Constraints:

Transported = Demand
TotalShipped = Supply

Dimensions:

For a transportation problem with $m$ sources and $n$ destinations, simplex tableau has $m + n + 1$ rows and $(m + 1)(n + 1)$ columns.

The transportation simplex tableau has only $m$ rows and $n$ columns!

Transportation simplex method

Version of the simplex called transportation simplex method.

Problems solved by hand can use a transportation simplex tableau.

Dimensions:

For a transportation problem with $m$ sources and $n$ destinations, simplex tableau has $m + n + 1$ rows and $(m + 1)(n + 1)$ columns.

The transportation simplex tableau has only $m$ rows and $n$ columns!
Transportation simplex tableau (TST)

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>n</th>
<th>Supply</th>
<th>( u_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( c_{11} )</td>
<td>( c_{12} )</td>
<td>...</td>
<td>( c_{1n} )</td>
<td>( s_1 )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( c_{21} )</td>
<td>( c_{22} )</td>
<td>...</td>
<td>( c_{2n} )</td>
<td>( s_2 )</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>( c_{m1} )</td>
<td>( c_{m2} )</td>
<td>...</td>
<td>( c_{mn} )</td>
<td>( s_n )</td>
<td></td>
</tr>
<tr>
<td>Demand</td>
<td>( d_1 )</td>
<td>( d_2 )</td>
<td>...</td>
<td>( d_n )</td>
<td>( Z = )</td>
<td></td>
</tr>
</tbody>
</table>

\( v_j \)

Additional information to be added to each cell:

- If \( x_{ij} \) is a basic variable:
  - \( c_{ij} \)
  - \( x_{ij} \)
- If \( x_{ij} \) is a nonbasic variable:
  - \( c_{ij} \)
  - \( c_{ij} - u_i - v_j \)

Transportation simplex method

- **Initialization**: construct an initial BF solution. To begin, all source rows and destination columns of the TST are initially under consideration for providing a basic variable (allocation).

  1. From the rows and columns still under consideration, select the next basic variable (allocation) according to one of the criteria:
    - Northwest corner rule
    - Vogel’s approximation method
    - Russell’s approximation method
Transportation simplex method

2. Make that allocation large enough to exactly use up the remaining supply in its row or the remaining demand in its column (whatever is smaller).

3. Eliminate that row or column (whichever had the smaller remaining supply or demand) from further consideration.

4. If only one row or one column remains under consideration, then the procedure is completed. Otherwise return to step 1.

- Go to the optimality test.

Optimality test: derive $u_i$ and $v_j$ by selecting the row having the largest number of allocations, setting its $u_i=0$, and then solving the set of equations $c_{ij} = u_i + v_j$ for each $(i,j)$ such that $x_{ij}$ is basic. If $c_{ij} - u_i - v_j \geq 0$ for every $(i,j)$ such that $x_{ij}$ is nonbasic, then the current solution is optimal and stop. Otherwise, go to an iteration.
Transportation simplex method

- **Iteration:**
  1. Determine the entering basic variable: select the nonbasic variable \( x_{ij} \) having the **largest** (in absolute terms) **negative** value of \( c_{ij} - u_i - v_j \).
  2. Determine the leaving basic variable: identify the chain reaction required to retain feasibility when the entering basic variable is increased. From the donor cells, select the basic variable having the **smallest** value.

- **Iteration:**
  3. Determine the new BF solution: add the value of the leaving basic variable to the allocation for each recipient cell. Subtract this value from the allocation for each donor cell.
  4. Apply the **optimality test**.
Assignment problem

- Special type of linear programming where assignees are being assigned to perform tasks.
- **Example:** employees to be given work assignments
- Assignees can be machines, vehicles, plants or even time slots to be assigned tasks.

Assumptions of assignment problems

1. The number of assignees and the number of tasks are the same, and is denoted by $n$.
2. Each assignee is to be assigned to exactly one task.
3. Each task is to be performed by exactly one assignee.
4. There is a cost $c_{ij}$ associated with assignee $i$ performing task $j$ ($i, j = 1, 2, \ldots, n$).
5. The objective is to determine how well $n$ assignments should be made to minimize the total cost.
Prototype example

- **Job Shop Company** has purchased three new machines of different types, and there are four different locations in the shop where a machine can be installed.
- Objective: assign machines to locations.

<table>
<thead>
<tr>
<th>Location</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine 1</td>
<td>13</td>
<td>16</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>Machine 2</td>
<td>15</td>
<td>–</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>Machine 3</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

**Cost per hour of material handling (in €)**

Formulation as an assignment problem

- We need the dummy machine 4, and an extremely large cost $M$:

<table>
<thead>
<tr>
<th>Location</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignee (Machine)</td>
<td>13</td>
<td>16</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>$M$</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>4(D)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Optimal solution**: machine 1 to location 4, machine 2 to location 3 and machine 3 to location 1 (total cost of 29€ per hour).
Assignment problem model

- Decision variables \( x_{ij} = \begin{cases} 1 & \text{if assignee } i \text{ performs task } j, \\ 0 & \text{if not.} \end{cases} \)

minimize \( Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \)

subject to

\[ \sum_{j=1}^{n} x_{ij} = 1, \quad \text{for } i = 1,2,\ldots,n, \]

\[ \sum_{i=1}^{n} x_{ij} = 1, \quad \text{for } j = 1,2,\ldots,n, \]

and \( x_{ij} \geq 0, \quad \text{for all } i \text{ and } j \)

\( (x_{ij} \text{ binary, for all } i \text{ and } j) \)

Assignment vs. Transportation prob.

- Assignment problem is a special type of transportation problem where sources = assignees and destinations = tasks and:
  - #sources \( m = \# \)destinations \( n \); 
  - every supply \( s_i = 1 \); 
  - every demand \( d_j = 1 \).

- Due to the integer solution property, since \( s_i \) and \( d_j \) are integers, every BF solution is an integer solution for an assignment problem. We may delete the binary restriction and obtain a linear programming problem!
Network representation

Parameter table as in transportation

<table>
<thead>
<tr>
<th>Supply</th>
<th>1</th>
<th>1</th>
<th>...</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Source</td>
<td>1</td>
<td>2</td>
<td>...</td>
<td>m = n</td>
</tr>
<tr>
<td>Destination</td>
<td>1</td>
<td>2</td>
<td>...</td>
<td>n</td>
</tr>
<tr>
<td>Cost per unit distributed</td>
<td>c_{11}</td>
<td>c_{12}</td>
<td>...</td>
<td>c_{1n}</td>
</tr>
<tr>
<td></td>
<td>c_{21}</td>
<td>c_{22}</td>
<td>...</td>
<td>c_{2n}</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>c_{n1}</td>
<td>c_{n2}</td>
<td>...</td>
<td>c_{nn}</td>
</tr>
</tbody>
</table>

[c_{ij} represents the cost per unit distributed from source i to destination j]
Concluding remarks

- A special algorithm for the assignment problem is the Hungarian algorithm (more efficient).
- Therefore streamlined algorithms were developed to explore the special structure of some linear programming problems: transportation or assignment problems.
- Transportation and assignment problems are special cases of minimum cost flow problems.
- Network simplex method solves this type of problems.