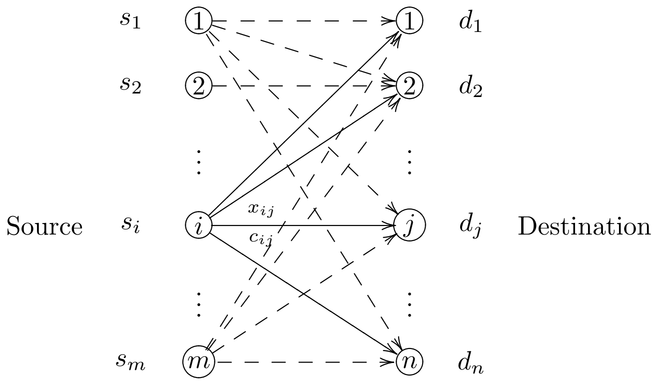
Chapter 6

TRANSPORTATION PROBLEMS

6.1 Transportation Model

Transportation models deal with the determination of a minimum-cost plan for transporting a commodity from a number of sources to a number of destinations. To be more specific, let there be m *sources* (or *origins*) that produce the commodity and n *destinations* (or *sinks*) that demand the commodity. At the i-th source, $i = 1, 2, \cdots, m$, there are s_i units of the commodity available. The demand at the j-th destination, $j = 1, 2, \cdots n$, is denoted by d_j. The cost of transporting one unit of the commodity from the i-th source to the j-th destination is c_{ij}. Let x_{ij}, $1 \leq i \leq m, 1 \leq j \leq n$, be the numbers of the commodity that are being transported from the i-th source to the j-th destination. Our problem is to determine those x_{ij} that will minimize the overall transportation cost. An optimal solution x_{ij} to the problem is called a *transportation plan*.



We note that at the i-th source, we have the i-th source equation

$$\sum_{j=1}^{n} x_{ij} = s_i, \qquad 1 \leq i \leq m,$$

while at the j-th destination, we have the j-th destination equation

$$\sum_{i=1}^{m} x_{ij} = d_j, \qquad 1 \leq j \leq n.$$

Notice that if the total demand equals the total supply, then we have the following *balanced trans-*

portation equation:

$$\sum_{i=1}^{m} s_i = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{n} d_j.$$

and the model is said to be *balanced*.

In the case of an unbalanced model, i.e. the total demand is not equal to the total supply, we can always add dummy source or dummy destination to complement the difference. In the following, we only consider balanced transportation models. They can be written as the following linear programming problem:

$$\begin{array}{ll} \min & x_0 = \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subject to} & \begin{cases} \displaystyle\sum_{j=1}^{n} x_{ij} = s_i & 1 \le i \le m, \\ \displaystyle\sum_{i=1}^{m} x_{ij} = d_j & 1 \le j \le n, \\ x_{ij} \ge 0 & 1 \le i \le m, 1 \le j \le n, \end{cases} \end{array} \tag{6.1}$$

where $\displaystyle\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$.

Notice that there are mn variables but only $m+n$ equations. To initiate the simplex method, we have to add $m+n$ more artificial variables and solving the problem by the simplex method seems to be a very tedious task even for moderate values of m and n. However, the transportation models possess some important properties that make the calculation easier to be handled.

Using the vector notations

$$\begin{aligned} \mathbf{x} &= [x_{11}, x_{12}, x_{13}, \cdots, x_{1n}, x_{21}, \cdots, x_{2n}, \cdots, x_{m1}, \cdots, x_{mn}]^T, \\ \mathbf{c} &= [c_{11}, c_{12}, c_{13}, \cdots, c_{1n}, c_{21}, \cdots, c_{2n}, \cdots, c_{m1}, \cdots, c_{mn}]^T, \\ \mathbf{b} &= [s_1, s_2, \cdots, s_m, d_1, d_2, \cdots, d_n]^T, \end{aligned}$$

the transportation model can be stated as the following linear programming problem:

$$\begin{array}{ll} \min & x_0 = \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \begin{cases} A\mathbf{x} = \mathbf{b}, \\ \mathbf{x} \ge \mathbf{0}. \end{cases} \end{array} \tag{6.2}$$

where the technology matrix A of the model is of the form:

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 & & & & & & & & & & & & & 0 \\ & & & & 1 & 1 & \cdots & 1 & & & & & & & & & \\ & & & & & & & & \ddots & & & & & & & & \\ & & & & & & & & & 1 & 1 & \cdots & 1 & & & & \\ & & & & & & & & & & & & & \ddots & & & \\ & & & & & & & & & & & & & & 1 & 1 & \cdots & 1 \\ 1 & & & & 1 & & & & & 1 & & & & & 1 & & & \\ & 1 & & & & 1 & & & & & 1 & & & & & 1 & & \\ & & \ddots & & & & \ddots & & \cdots & & & \ddots & & \cdots & & & \ddots & \\ 0 & & & 1 & & & & 1 & & & & & 1 & & & & & 1 \end{bmatrix} \tag{6.3}$$

Hence if we denote $\mathbf{a}_{ij}$ the $[(i-1)n+j]$-th column of the matrix A, then

$$\mathbf{a}_{ij} = \mathbf{e}_i + \mathbf{e}_{m+j} \qquad i = 1, 2, \cdots, m, \quad j = 1, 2, \cdots, n. \tag{6.4}$$

Here as usual, $\mathbf{e}_i$ denotes the i-th unit vector. Next we are going to prove some algebraic properties of the matrix A.

Theorem 6.1. *The rank of the matrix A is equal to $m + n - 1$.*

Proof. We first claim that $\operatorname{rank}(A) \leq m + n - 1$. Let $\mathbf{s}_i$ be the i-th row of A (*source rows*) and $\mathbf{d}_j$ be the $(m+j)$-th row of A (*destination rows*). Then it is clear from (6.3) that

$$\sum_{i=1}^{m} \mathbf{s}_i - \sum_{j=1}^{n} \mathbf{d}_j = \mathbf{0}.$$

Hence the rows $\mathbf{s}_i$ and $\mathbf{d}_j$ are linearly dependent. Thus $\operatorname{rank}(A) < m + n$.

Next we prove that $\operatorname{rank}(A) \geq m + n - 1$ by constructing a nonsingular $(m+n-1)$-by-$(m+n-1)$ submatrix of A. Suppose we take the n-th, the $2n$-th, the $3n$-th, $\cdots$, the mn-th columns of A together with the 1-st, the the 2-nd, the 3-rd, $\cdots$, the $(n-1)$-th columns of A. This resulting matrix is of order $(m+n)$ by $(m+n-1)$. If we delete the last row of the matrix, then we obtain the following $(m+n-1)$-by-$(m+n-1)$ matrix:

$$D = \begin{bmatrix} 1 & 0 & & 0 & 1 & 1 & \cdots & 1 \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & 0 & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix}.$$

Since D is a triangular matrix, $\det D = 1$. Therefore D is non-singular and $\operatorname{rank}(A) \geq \operatorname{rank}(D) = m + n - 1$. Thus we conclude that $\operatorname{rank}(A) = m + n - 1$, which is equivalent to saying that one of the equations in (6.1)is redundant. $\square$

Thus a basic solution to (6.1) has at most $m + n - 1$ nonzero entries.

Theorem 6.2. *Every minor of A can only have one of the values* 1, -1 *or* 0. *More precisely, given any A_k, a k-by-k submatrix of A, we have* $\det A_k = \pm 1$ *or* 0.

Proof. Notice first that every column of A has exactly two 1's, thus any column of A_k has either two 1's, only one 1 or exactly no 1. If A_k contains a column that has no 1, then clearly $\det A_k = 0$ and we are done. Thus we may assume that every column of A_k contains at least one 1. There are two cases to be considered. The first case is where every column of A_k contains two 1's. Then one of the 1's must come from the source rows and the other one must come from the destination rows. Hence subtracting the sum of all source rows from the sum of all destination rows in A_k will give us the zero vector. Thus the row vectors of A_k are linearly dependent. Hence $\det A_k = 0$. It remains to consider the case where at least one column of A_k contains exactly one 1. By expanding A_k with respect to this column, we have

$$\det A_k = \pm \det A_{k-1}$$

where the sign depends on the indices of that particular 1. Now the theorem is proved by repeating the argument to A_{k-1}. $\square$ $\square$

Definition 6.1. A matrix A is said to be *totally unimodular* if every minor of A is either 1, -1 or 0.

Thus the coefficient matrix of a transportation problem is totally unimodular.

6.2 The Simplex Method and Transportation Problems

Let us first prove that transportation models always have optimal solution. In fact, given problem (6.1), if we put

$$x_{ij} = \frac{s_i d_j}{\alpha}, \quad 1 \le i \le m, 1 \le j \le n,$$

where

$$\alpha = \sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j,$$

then it is easy to check that it is a solution to $A\mathbf{x} = \mathbf{b}$. Hence transportation problems always have a feasible solution. Since all x_{ij} and c_{ij} are nonnegative, $x_0 \ge 0$. In particular, the objective function is bounded from below. Hence it follows that a transportation problem must have an optimal solution.

Let us see what happens if (6.1) is solved by simplex method. Since rank $A = m + n - 1$, a basic optimal solution to (6.1) have only $m + n - 1$ basic variables, i.e. no more than $m + n - 1$ of the x_{ij} in the solution are different from zero. To solve (6.1) by simplex method, we first change it into standard form by adding $m + n$ artificial variables to (6.2). Then we have

$$\begin{array}{ll} \min & x_0 = \mathbf{c}^T\mathbf{x} + M\mathbf{1}^T\mathbf{x}_a \\ \text{subject to} & \begin{cases} [\ A, I\ ] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_a \end{bmatrix} = \mathbf{b}, \\ \qquad \mathbf{x}, \mathbf{x}_a \ge 0. \end{cases} \end{array} \tag{6.5}$$

Here $\mathbf{x}_a$ is the artificial variable. Since basic feasible solution exists, the artificial variables for the problem can always be driven to zero in phase I (or else the problem has no optimal solution, a contradiction). Since (6.2) and (6.5) have the same optimal solution, basic optimal solution to (6.5) can have no more than $m+n-1$ non-zero variables, i.e. a basic optimal solution to (6.5) must have at least one artificial variable in the basis at zero level. (Recall that artificial variables at zero level in Phase II indicate redundancy).

Suppose that we have found by some means a basic feasible solution to (6.5) which is also a feasible solution to (6.2), (i.e. we are in phase II). Let B be the basic matrix (of order $m + n$) of $[A, I]$, then B contains $m + n - 1$ columns of A and one artificial vector $\mathbf{q}$ with the corresponding artificial variable at zero level. Therefore we may consider the $m+n-1$ linearly independent column vectors of A in B as a set of basis vectors for (6.2). The collection of these $m + n - 1$ vectors will be denoted by $\mathbf{a}^B_{\alpha\beta}$ and the corresponding basic variables will be denoted by $x^B_{\alpha\beta}$. More precisely, if $B = [\mathbf{a}^B_{\alpha\beta}, \mathbf{q}]$ is basic matrix for (6.5), we then define $B = [\mathbf{a}^B_{\alpha\beta}]$ as a basic matrix for (6.2).

We observe that any column vector $\mathbf{a}_{ij}$ of A is just a linear combination of vectors of B, i.e.

$$\mathbf{a}_{ij} = \sum_{\alpha\beta} y_{(\alpha\beta)(ij)} \mathbf{a}^B_{\alpha\beta} \tag{6.6}$$

where $\sum\limits_{\alpha\beta}$ means summation over all vectors in the basis. We recall that (6.6) is just the change of basis equation (2.24):

$$\mathbf{a}_{ij} = B\mathbf{y}_{ij}, \tag{6.7}$$

where B is $(m + n)$-by-$(m + n - 1)$ and contains the columns $\mathbf{a}^B_{\alpha\beta}$. Thus in the language of simplex method, $y_{(\alpha\beta)(ij)}$ are just the entries in the simplex tableau at the current iteration. Now we prove the two most important properties of transportation models.

Theorem 6.3. *The coefficients $y_{(\alpha\beta)(ij)}$ can only take the values $1, -1$ or 0.*

Proof. Let R_i be the $(m+n-1)$-by-$(m+n-1)$ matrix obtained from B in (6.7) by deleting the ith row of B. By (6.7), $R_i\mathbf{y}_{ij}$ is the same as $\mathbf{a}_{ij}$ with the ith entry removed. Hence by (6.4), we see that

$$R_i\mathbf{y}_{ij} = \mathbf{e}_{m-1+j}.$$

Thus

$$\mathbf{y}_{ij} = R_i^{-1}\mathbf{e}_{m+1-j} = \frac{1}{\det R_i}(\operatorname{adj} R_i)\mathbf{e}_{m-1+j},$$

where $\operatorname{adj} R_i$ is the adjoint of R_i. Note that R_i is obtained by taking $(m+n-1)$ columns and $(m+n-1)$ rows of A, hence is a submatrix of A. This also follows from the fact that R_i is a submatrix of B and B is a submatrix of A. Since B is a basic matrix, R_i has full rank. Thus, by Theorem 6.2, we have $\det R_i = \pm 1$. Since the entries of $\operatorname{adj} R_i$ are just minors of R_i and hence of A, their values can only be ± 1 or 0. Thus we see that $\mathbf{y}_{ij} = \pm 1$ or 0. □

Thus (6.6) becomes

$$\mathbf{a}_{ij} = \sum_{\alpha\beta}(\pm 1)\mathbf{a}^B_{\alpha\beta},$$

where we have omitted those $\mathbf{a}^B_{\alpha\beta}$ with $y_{(\alpha\beta)(ij)} = 0$ in the summation. We note that the conclusion of Theorem 6.3 holds for any linear programming problem where its coefficient matrix is totally unimodular.

Theorem 6.4 (The Stepping Stones Theorem). *Let $B = \{\mathbf{a}_{\alpha\beta}\}$ be a set of $(m+n-1)$ linearly independent columns of A. Then for all column vector $\mathbf{a}_{ij}$ of A, $1 \le i \le m$, $1 \le j \le n$, we have*

$$\mathbf{a}_{ij} = \mathbf{a}_{ii_1} - \mathbf{a}_{i_2i_1} + \mathbf{a}_{i_2i_3} - \mathbf{a}_{i_4i_3} + \cdots + (-1)^k\mathbf{a}_{i_kj}, \tag{6.8}$$

where $\mathbf{a}_{ii_1}, \mathbf{a}_{i_l,i_{l+1}}, \mathbf{a}_{i_kj}$ are in B for $l = 1, \cdots, k-1$. Moreover, the expression (6.8) is unique.

Proof. Since $\operatorname{rank} A = m+n-1$, all column vectors of A can be written as a linearly combinations of vectors in B. Moreover by Theorem 6.3, we have

$$\mathbf{a}_{ij} = \sum_{\alpha\beta}(\pm 1)\mathbf{a}_{\alpha\beta} = \sum_{\alpha\beta\in I^+}\mathbf{a}_{\alpha\beta} - \sum_{\gamma\delta\in I^-}\mathbf{a}_{\gamma\delta},$$

where $I^\pm$ are index sets depending on the $\mathbf{a}_{ij}$. By (6.4), this becomes

$$\mathbf{e}_i + \mathbf{e}_{m+j} = \sum_{\alpha\beta\in I^+}\mathbf{e}_\alpha + \sum_{\alpha\beta\in I^+}\mathbf{e}_{m+\beta} - \sum_{\gamma\delta\in I^-}\mathbf{e}_\gamma - \sum_{\gamma\delta\in I^-}\mathbf{e}_{m+\delta}.$$

From this expression, it is clear that there exists an $1 \le i_1 \le n$ such that $(ii_1) \in I^+$. Subtracting $\mathbf{a}_{ii_1}$ from both sides, we get

$$\mathbf{e}_{m+j} - \mathbf{e}_{m+i_1} = \sum_{\alpha\beta\in I_1^+}\mathbf{e}_\alpha + \sum_{\alpha\beta\in I_1^+}\mathbf{e}_{m+\beta} - \sum_{\gamma\delta\in I^-}\mathbf{e}_\gamma - \sum_{\gamma\delta\in I^-}\mathbf{e}_{m+\delta},$$

where $I_1^+ = I^+ \setminus \{(ii_1)\}$. Now if $i_1 = j$, we are done. If not, then from the expression, it is clear that there exists an $1 \le i_2 \le m$ such that $(i_2i_1) \in I^-$. Subtracting $\mathbf{a}_{i_2i_1} = \mathbf{e}_i - \mathbf{e}_{m+i_1}$ from both sides, we get

$$\mathbf{e}_{m+j} + \mathbf{e}_{i_2} = \sum_{\alpha\beta\in I_1^+}\mathbf{e}_\alpha + \sum_{\alpha\beta\in I_1^+}\mathbf{e}_{m+\beta} - \sum_{\gamma\delta\in I_2^-}\mathbf{e}_\gamma - \sum_{\gamma\delta\in I_2^-}\mathbf{e}_{m+\delta},$$

where $I_2^- = I^- \setminus \{(i_2i_1)\}$. Equation (6.8) now follows by repeating the arguments again until I_l^- is empty. Since B is a basis, it is clear that (6.8) is unique. □

In the following, we consider how to iterate from one simplex tableau to the next.

Update of the solution $\mathbf{x}$.
Let $\mathbf{a}_{st}$ be the entering vector and $\mathbf{a}_{uv}^B$ be the leaving vector. Then the solution x_{ij} are updated according to

$$\begin{cases} \hat{x}_{\alpha\beta}^B = x_{\alpha\beta}^B - x_{uv}^B \dfrac{y_{(\alpha\beta)(st)}}{y_{(uv)(st)}} & \text{if } (\alpha\beta) \neq (uv) \\ \hat{x}_{uv} = \dfrac{x_{uv}^B}{y_{(uv)(st)}} & \end{cases} \tag{6.9}$$

This equation is to be compared with the updating rule in simplex method:

$$\begin{cases} \hat{x}_{B_i} = x_{B_i} - x_{B_r} \dfrac{y_{ij}}{y_{rj}} & \text{if } i \neq r \\ \hat{x}_{B_r} = \dfrac{x_{B_r}}{y_{rj}} & \end{cases}$$

But by Theorem 6.3, the pivot element $y_{(uv)(st)}$ will always be equal to 1 and that the other $y_{(\alpha\beta)(st)} = \pm 1$ or 0. Therefore, we see that (6.9) can be rewritten as

$$\begin{cases} \hat{x}_{\alpha\beta}^B = x_{\alpha\beta}^B \quad \text{or} \quad \hat{x}_{\alpha\beta}^B = x_{\alpha\beta}^B \pm x_{uv}^B \\ \hat{x}_{uv}^B = x_{uv}^B \end{cases} \tag{6.10}$$

The property (6.10) is usually referred to as *integer property*. It shows that if the starting basic feasible solution $\mathbf{x}$ is an integral vector (i.e. all entries are integer), then at each subsequent iteration, the solution $\mathbf{x}$ is also an integral vector. In particular, the optimal solution $\mathbf{x}^*$ is also an integral vector.

We remark that the integer property of transportation problems is derived from the fact that all entries of $\mathbf{y}_{ij}$ can either be 1, -1 or 0. Thus by recalling Theorem 6.3, we see that if the coefficient matrix of a linear programming problem is totally unimodular, then the problem will have the integer property.

Update of tableau entries $\mathbf{y}_{ij}$.
For usual simplex method, the tableau entries $\mathbf{y}_j$ are updated by the elementary row operations:

$$\begin{cases} \hat{y}_{B_i} = y_{B_i} - \dfrac{y_{ij}}{y_{rj}} & \text{if } i \neq r \\ \hat{y}_{B_r} = \dfrac{y_{B_r}}{y_{rj}} & \end{cases}$$

In our notations, we then have

$$\begin{cases} \hat{y}_{(\alpha\beta)(ij)} = y_{(\alpha\beta)(ij)} - \dfrac{y_{(uv)(ij)}}{y_{(uv)(st)}} & \text{if } (\alpha\beta) \neq (uv) \\ \hat{y}_{(uv)(ij)} = \dfrac{y_{(uv)(ij)}}{y_{(uv)(st)}} & \end{cases}$$

Since the pivot element $y_{(uv)(st)}$ is always equal to 1, we have

$$\begin{cases} y_{(\alpha\beta)(ij)} = y_{(\alpha\beta)(ij)} - y_{(uv)(ij)} & \text{if } (\alpha\beta) \neq (uv) \\ y_{(uv)(ij)} = y_{(\alpha\beta)(ij)} & \end{cases}$$

Computation of $z_{ij} - c_{ij}$.

Recall that in the simplex method,

$$z_j - c_j = \mathbf{c}_{B_i}^T \mathbf{y}_j - c_j = \sum_{x_i \in B} c_{B_i} y_{ij} - c_j.$$

Hence in our notations, we have

$$z_{ij} - c_{ij} = \sum_{x_{(\alpha\beta)} \in B} y_{(\alpha\beta)(ij)} c_{\alpha\beta}^B - c_{ij}. \tag{6.11}$$

Because of the simple algebraic structure of the transportation models, it is not necessary to use the simplex tableau, which is of size $(m+n+1)$ by $(mn+n+n+1)$, to hold all necessary information. In the following, we will construct a different tableau, called the *transportation tableau*, that can hold the same pieces of information and yet is easy to be handled. For the transportation model in (6.1), its transportation tableau consists of m by n boxes and is of the form

						Supply
	x_{11} c_{11}	x_{12} c_{12}	$\cdots$ $\cdots$	$\cdots$ $\cdots$	x_{1n} c_{1n}	s_1
	x_{21} c_{21}	x_{22} c_{22}	$\cdots$ $\cdots$	$\cdots$ $\cdots$	x_{2n} c_{2n}	$\vdots$
	$\vdots$ $\vdots$	$\vdots$ $\vdots$	x_{ij} c_{ij}	$\vdots$ $\vdots$	$\vdots$ $\vdots$	s_i
	$\vdots$ $\vdots$	$\vdots$ $\vdots$	$\vdots$ $\vdots$	$\vdots$ $\vdots$	$\vdots$ $\vdots$	$\vdots$
	x_{m1} c_{m1}	x_{m2} c_{m2}	$\cdots$ $\cdots$	$\cdots$ $\cdots$	x_{mn} c_{mn}	s_m
Demand	d_1	$\cdots$	d_j	$\cdots$	d_n	

In the transportation tableau, nonbasic variables (i.e. those $\mathbf{a}_{ij}$ not in the basis) are not written out explicitly.

Recall that simplex tableau contains the following information:

(i) The current solution in the $\mathbf{b}$ column.

(ii) The x_0 row contains the reduced cost coefficients $z_j - c_j$.

(iii) The transformed columns of A, denoted as usual by $\mathbf{y}_j$. They are related to the columns $\mathbf{a}_j$ of A by (2.20): $\mathbf{y}_j = B^{-1}\mathbf{a}_j$.

(iv) The current basic variables.

We will see that the transportation tableau can be manipulated easily to give us these necessary pieces of information. For one thing, according to our convention on the transportation tableau, those variables that are not listed in the tableau are nonbasic. Those variables that are listed are basic and their values are the values of the current basic feasible solution. Next we show by an example how to compute the current coefficient matrix $y_{(\alpha\beta)(ij)}$ and the corresponding reduced cost coefficient $z_{ij} - c_{ij}$.

Example 6.1. Let us consider a problem with eight variables x_{ij}, $1 \le i \le 2$, $1 \le j \le 4$. We then have the following 2-by-4 transportation tableau.

x_{11}	x_{12}		
	x_{22}	x_{23}	x_{24}

Since rank $A = 2 + 4 - 1 = 5$, there will be five basic variables in any basic feasible solutions of the problem. According to our convention, $x_{11}, x_{12}, x_{22}, x_{23}$ and x_{24} are the current basic variable. Thus $\mathbf{a}_{11}, \mathbf{a}_{12}, \mathbf{a}_{22}, \mathbf{a}_{23}$ and $\mathbf{a}_{24}$ are the basis vectors. The other three vectors are just linearly combinations of these five vectors. For example,

$$\mathbf{a}_{21} = \mathbf{a}_{22} - \mathbf{a}_{12} + \mathbf{a}_{11}.$$

Thus $y_{(22)(21)} = y_{(11)(21)} = 1$ and $y_{(12)(21)} = -1$. Therefore, according to (6.11),

$$z_{21} - c_{21} = c_{22} - c_{12} + c_{11} - c_{21}.$$

Similarly, we have

$$\mathbf{a}_{13} = \mathbf{a}_{12} - \mathbf{a}_{22} + \mathbf{a}_{23},$$

i.e. $y_{(12)(13)} = y_{(23)(13)} = 1$ and $y_{(22)(13)} = -1$. Therefore,

$$z_{13} - c_{13} = c_{12} - c_{22} + c_{23} - c_{13}.$$

Finally,

$$\mathbf{a}_{14} = \mathbf{a}_{12} - \mathbf{a}_{22} + \mathbf{a}_{24}$$

and hence $y_{(12)(14)} = y_{(24)(14)} = 1$ and $y_{(22)(14)} = -1$. Thus we have

$$z_{14} - c_{14} = c_{12} - c_{22} + c_{24} - c_{14}.$$

We remark that a loop is formed each time. For example, for x_{14}, we have the following loop.

x_{11}	x_{12}		
	x_{22}	x_{23}	x_{24}

The loop starts with a nonbasic variable, passes through a sequence of basic variables (stepping stones) and finally returns to the starting nonbasic variable. This is a consequence of Theorem 6.4. We note moreover that for each of these nonbasic variable, there exists one and only one such loop. In fact, by Theorem 6.4, if there are two such loops, then the nonbasic vector can be expressed in terms of two different linear combinations of vectors in the basis. This will be a contradiction to the linear independence of the basic vectors.

6.3 The Starting Basic Feasible Solution

In this section, we are concerned with the problem of finding a starting basic feasible solution to the transportation problem. There are many methods for finding such a starting BFS. The easier ones are the *northwest-corner method*, the *column minima method* and the *row-minima method*. In the following, we introduce the *matrix minima method* (or *least cost method*) that will give a better starting BFS in the sense that the starting value of the objective function is usually smaller.

At each step of the method, we select, among all *uncrossed-out* x_{ij}, the x_{ij} with smallest unit cost, and assign as large as possible to x_{ij}. Then we cross out the satisfied row or column. (That means x_{ij} so added must either satisfy a row equation or a column equation). If both the row and the column are satisfied simultaneously, we only cross out one. Then we adjust the supply and demand for all uncrossed-out rows and columns and repeat the whole process. The starting basic feasible solution is obtained when exactly one row *or* one column is left uncrossed out.

Example 6.2. Consider the following transportation tableau.

10	0	20	11	15
12	7	9	10	25
0	14	16	18	5
5	15	15	10	

It has 3 source and 4 destinations. Thus the number of basic variables is $3 + 4 - 1 = 6$. To find the starting basic feasible solution, we begin by searching the smallest cost in the tableau. We note that both c_{12} and c_{31} are zero. We break the tie arbitrary and choose c_{31}. Hence we set $x_{31} = 5$. In that case, both row 3 and column 1 are satisfied. We readjust the row sum and column sum accordingly. Though both row sum and column sum are zero, we can only cross out one of them. In the example, we cross out only the first column. The next four tableaus show the steps in obtaining the starting BFS.

1				
10	0	20	11	15
12	7	9	10	25
0	14	16	18	0
5				
0	15	15	10	

1	2			
10	0	20	11	0
	15			
12	7	9	10	25
0	14	16	18	0
5				
0	0	15	10	

	1	2	3		
	^{10}	^{0} 15	^{20}	^{11}	0
	^{12}	^{7}	^{9} 15	^{10}	10
	^{0} 5	^{14}	^{16}	^{18}	0
	0	0	0	10	

	1	2	3	4	
	^{10}	^{0} 15	^{20}	^{11}	0
	^{12}	^{7}	^{9} 15	^{10} 10	0
	^{0} 5	^{14}	^{16}	^{18}	0
	0	0	0	0	

Thus the starting basic feasible solution is given by

$$x_{14} = x_{34} = 0,\ x_{31} = 5,\ x_{24} = 10,\ x_{12} = x_{23} = 15,$$

and the starting basis vectors are $\mathbf{a}_{14}, \mathbf{a}_{34}, \mathbf{a}_{31}, \mathbf{a}_{24}, \mathbf{a}_{12}$ and $\mathbf{a}_{23}$. The starting value of the objective function is

$$x_0 = 15 \times 9 + 10 \times 10 = 235.$$

Example 6.3. For the following problem, the starting BFS is given by the boxed variables in the following tableau.

	6	1		5		3	
4	^{2} $\boxed{0}$	^{1} $\boxed{50}$	^{3}	^{3}	^{2}	^{5}	50
7	^{3} $\boxed{30}$	^{12}	^{9}	^{4}	^{3} $\boxed{10}$	^{4}	40
	^{3}	^{5}	^{4} $\boxed{20}$	^{2} $\boxed{9}$	^{4} $\boxed{20}$	^{1} $\boxed{11}$	60
2	^{4}	^{2}	^{2}	^{1} $\boxed{31}$	^{2}	^{2}	31
	30	50	20	40	30	11	

6.4 Iteration on the Transportation Tableau

Having found the starting BFS, we will need to determine the entering and leaving variables for the next iteration.

The Entering Variable.

Recall that the entering variable is determined by the reduced cost coefficients. Since we are doing a minimization problem, the optimality condition is that

$$z_{ij} - c_{ij} \geq 0, \quad 1 \leq i \leq m, 1 \leq j \leq n.$$

The entering variable is chosen to be the variable with positive $z_{ij} - c_{ij}$. Usually, we choose the first x_{ij} which has positive $z_{ij} - c_{ij}$ to be our entering variable or we can also choose the x_{ij} with the largest positive $z_{ij} - c_{ij}$. If all $z_{ij} - c_{ij}$ are non-positive, then we have reached the optimal solution. Recall that $z_{ij} - c_{ij}$ is always zero for basic variables. Notice that by (6.11) and Theorem 6.4, $z_{ij} - c_{ij}$ is of the form

$$z_{ij} - c_{ij} = c_{ii_1} - c_{i_2i_1} + c_{i_2i_3} - \cdots + c_{i_kj} - c_{ij}$$

where $x_{ii_1}, x_{i_2i_1}, x_{i_2i_3}, \cdots, x_{i_kj}$ is a closed loop of basic variables that starts and ends at x_{ij}. Thus the reduced cost can be computed as follows:

(i) Find for each nonbasic variable x_{ij}, a closed loop that starts and ends at the nonbasic variable, yet each corner of the loop is a basic variable.

(ii) Then z_{ij} is computed by adding and subtracting in sequence the unit cost of each of the basic variable in the loop.

(iii) Finally, we compute $z_{ij} - c_{ij}$ and write it in the lower left hand corner of the box that belongs to x_{ij}.

Notice that once we have found an $z_{ij} - c_{ij}$ that is positive, we can choose the corresponding x_{ij} as the entering variable. If we want the leaving variable to be the one with the most positive $z_{ij} - c_{ij}$, then at each iteration, we have to compute all $z_{ij} - c_{ij}$.

Example 6.4. Consider the following transportation problem.

$\boxed{10}$ 10	$\boxed{20}$	$\boxed{5}$	$\boxed{7}$
$\boxed{13}$	$\boxed{9}$ 20	$\boxed{12}$	$\boxed{8}$
$\boxed{4}$ 30	$\boxed{15}$	$\boxed{7}$	$\boxed{9}$
$\boxed{14}$	$\boxed{7}$ 10	$\boxed{1}$ 20	$\boxed{0}$ 10
$\boxed{3}$ 20	$\boxed{12}$ 30	$\boxed{5}$	$\boxed{19}$

We have

$$\begin{aligned} z_{12} - c_{12} &= c_{11} - c_{51} + c_{52} - c_{12} \\ &= 10 - 3 + 12 - 20 = -1 \\ z_{13} - c_{13} &= c_{11} - c_{51} + c_{52} - c_{42} + c_{43} - c_{13} \\ &= 10 - 3 + 12 - 7 + 1 - 5 = 8. \end{aligned}$$

All other reduced cost coefficients are evaluated similarly. In the following transportation tableau, we list all these reduced cost coefficients in the lower left hand corner of their respective boxes.

10 10 0	20 -1	5 8	7 5
13 -13	9 20 	12 -9	8 -6
4 30 	15 -2	7 0	9 -3
14 -10	7 10 	1 20 	0 10
3 20 	12 30 	5 1	9 -14

We see that the entering variable is x_{13} with $z_{13} - c_{13} = 8$.

The Leaving Variable.

Next we consider the leaving variable which is determined by the feasibility condition. Recall that the y_{ij} are either 1, -1 or 0. The leaving variable is thus the variable with positive y_{ij} and minimum x_{ij}.

Example 6.5. Continuing with Example 4, since the entering vector $\mathbf{a}_{13}$ is given by

$$\mathbf{a}_{13} = \mathbf{a}_{11} - \mathbf{a}_{51} + \mathbf{a}_{52} - \mathbf{a}_{42} + \mathbf{a}_{43},$$

it follows that $y_{11} = 1$, $y_{52} = y_{43} = 1$ and $y_{51} = y_{42} = -1$. Hence we only have to consider x_{11}, x_{52} and x_{43}. Since x_{11} is the minimum, x_{11} is the leaving variable.

The Iteration.

Now we have to update the solution $\mathbf{x}$. This can be done by using the integer property (6.10). However, there is an easy way. We note that x_{11} has to be set to zero in the next iteration and x_{13} is going to be nonzero, yet we have to satisfy the constraints. How do we accomplish this? When x_{11} is reduced by δ to $10 - \delta$, then we can set x_{51} to $20 + \delta$ to make up the difference in the first column. But then x_{52} has to be $30 - \delta$ to satisfy the constraint for the fifth row. Using this idea recursively, we get

10 $10-\delta$	20	5 δ	7
13	9 20	12	8
4 30	15	7	9
14	7 $10+\delta$	1 $20-\delta$	0 10
3 $20+\delta$	12 $30-\delta$	5	19

Finally we see that $x_{13} = \delta$. Since $x_{11} = 0$ in the next iteration, $\delta = 10$, and we obtain the following tableau.

10	20	5 10	7
13	9 20	12	8
4 30	15	7	9
14	7 20	1 10	0 10
3 30	12 20	5	19

Using the same optimality and feasibility criteria, we see that in the next iteration, the entering variable is x_{53} and the leaving variable is x_{43}. By setting $\hat{x}_{53} = x_{43}$, $\hat{x}_{43} = 0$ and adjusting the values of the other basic variables in the loop, we have the following tableau.

10	20	5 / 10	7
13	9 / 20	12	8
4 / 30	15	7	9
14	7 / 30	1	0 / 10
3 / 30	12 / 10	5 / 10	19

Computing all the reduced cost coefficients for the tableau, we find that we have reached the optimal solution with optimal cost

$$x_0 = 5 \cdot 10 + 9 \cdot 20 + 4 \cdot 30 + 7 \cdot 30 + 5 \cdot 10 + 12 \cdot 10 + 3 \cdot 30 = 820.$$

6.5 Method of Multipliers

This is another method for evaluating $z_{ij} - c_{ij}$. The idea is to use the dual variables. It can be checked that the dual of (6.1)is given by:

$$\begin{array}{ll} \max & \displaystyle\sum_{i=1}^{m} s_i u_i + \sum_{j=1}^{n} d_j v_j \\ \text{subject to} & \begin{cases} u_i + v_j \le c_{ij} & 1 \le i \le m,\ 1 \le j \le n \\ \quad u_i, v_j & \text{free} \end{cases} \end{array} \tag{6.12}$$

In the method of multipliers, we write, for each basic variable x_{ij}^B,

$$c_{ij} = u_i + v_j. \tag{6.13}$$

This follows from the Theorem of complementary slackness, namely if the primal structural variable is positive, then the dual slack variable must be zero. Since

$$\begin{aligned} z_{ij} - c_{ij} &= \left(c_{ii_1} - c_{i_2 i_1} + \cdots + c_{ik_j}\right) - c_{ij} \\ &= u_i + v_{i_1} - (u_{i_2} + v_{i_1}) + \cdots + (u_{i_k} + v_j) - c_{ij} \\ &= u_i + v_j - c_{ij}, \end{aligned}$$

we have

$$z_{ij} - c_{ij} = u_i + v_j - c_{ij}. \tag{6.14}$$

Since there are $(m + n - 1)$ basic variables, there are $(m + n - 1)$ equations in (6.13). However, there are $m + n$ unknown variables u_i and v_j in (6.13). Thus we can arbitrarily assign any one of the u_i or v_j a value and evaluate all the other unknowns accordingly. Once all u_i and v_j have been determined, $z_{ij} - c_{ij}$ can be evaluated by (6.14).

Example 6.6. Consider the transportation tableau in Example 6.4 again. We have

$$\begin{cases} u_1 + v_1 = 10 \\ u_2 + v_2 = 9 \\ u_3 + v_1 = 4 \\ u_4 + v_2 = 7 \\ u_4 + v_3 = 1 \\ u_4 + v_4 = 0 \\ u_5 + v_1 = 3 \\ u_5 + v_2 = 12 \end{cases}$$

By assigning $u_1 = 0$, we have

$$u_1 = 0, u_2 = -10, u_3 = -6, u_4 = -12, u_5 = -7$$

and

$$v_1 = 10, v_2 = 19, v_3 = 13, v_4 = 12.$$

We can write them in the transportation tableau as follows:

$u_i \backslash v_j$	10	19	13	12
0	^{10} 10	^{20}	^{5}	^{7}
-10	^{13}	^{9} 20	^{12}	^{8}
-6	^{4} 30	^{15}	^{7}	^{9}
-12	^{14}	^{7} 10	^{1} 20	^{0} 10
-7	^{3} 20	^{12} 30	^{5}	^{19}

Using (6.14), we can get $z_{ij} - c_{ij}$ for all i and j easily. For example,

$$z_{21} - c_{21} = u_2 + v_1 - c_{21} = -10 + 10 - c_{21} = -c_{21} = -13 .$$

Before we end this section, let us note that at the optimal tableau, the u_i and v_j so solved by (6.13) will gives the optimal values of the dual variables for the dual problem (6.12). In fact, by Theorem 5.5, we see that at optimal, the dual vector

$$\mathbf{u} = [u_1, u_2, \cdots, u_m, v_1, v_2, \cdots, v_n]$$

is given by

$$\mathbf{c}_B^T = \mathbf{u}^T B, \tag{6.15}$$

where B is made up of $(m+n-1)$ columns of A. By recalling the special structure of A in (6.3), we see that (6.15) is equivalent to (6.13). In fact, if $\mathbf{a}_{ij}^B$ is a vector in the basis (i.e. in B), then by (6.4) and (6.15), we have

$$c_{ij}^B = \mathbf{u}^T \mathbf{a}_{ij}^B = \mathbf{u}^T(\mathbf{e}_i + \mathbf{e}_{m+j}) = u_i + v_j.$$

6.6 Transshipment Model

The standard transportation model assumes that the direct route between a source and a destination is a minimum-cost route. However, in actual application, the minimum-cost route is not known a priori. In fact, the minimum-cost route from one source to another destination may well pass through another source first. The transportation techniques we developed in the previous sections can be adapted to find the minimum-cost route systematically.

The idea is to formulate the problem of finding the minimum-cost route as a *transshipment model* and then solve the transshipment model by transportation techniques. In transshipment model, commodities are allowed to pass transiently through other sources and destinations before it ultimately reaches its designated destination. It therefore is capable of seeking the minimum-cost route between a source and a destination.

To put the transshipment model in the context of transportation problem, we note that in transshipment model, the entire supply from all sources could potentially pass through any source or destination before it is redistributed again. This means that each source or destination node in the transportation network can be considered both as a transient source and a transient destination. Thus the number of sources equals to the number of destinations in the transshipment models and that number is equal to the sum of sources and destinations in the corresponding transportation models.

To allow transient passing of the commodity, an additional buffer stock B has to be allowed at each source and destination. Since potentially, the entire supply from all sources could pass through any one of the node, the size of the buffer stock has to be at least equals to the sum of supply or demand of the transportation model, i.e.

$$B \geq \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

This amount of buffer stock has to be added to each source and destination nodes in the transportation network.

Before one can use the transportation technique to solve the transshipment model, one has to determine the unit cost of shipping the commodities through the transient nodes. In general, the shipping cost from one location to itself should be zero and the shipping cost from the source S_i to the destination D_j should be the same as the shipping cost from D_j to S_i, but that may change depending on the problem. However, one should note that the unit shipping cost from a source to another source or from a destination to another destination is in general not given in the original transportation problem. Thus these figures have to be given before the transshipment models is completed.

Example 6.7. Consider the following transportation problem.

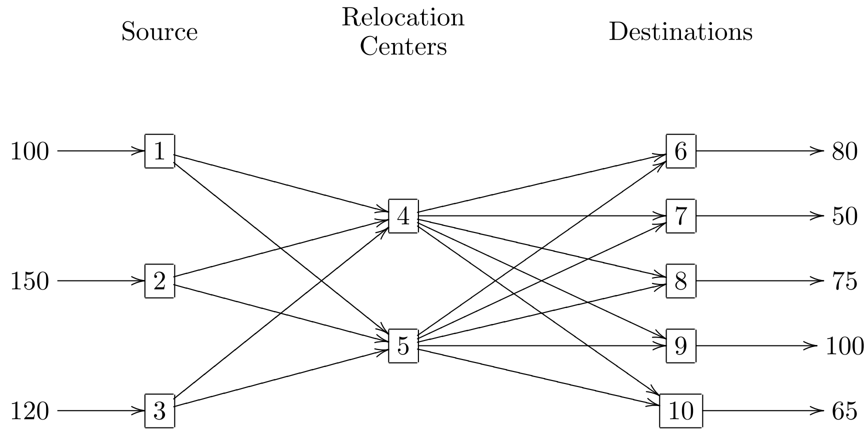
	D_1	D_2	
S_1	5	7	50
S_2	12	8	40
S_3	7	9	60
	70	80	

To formulate the problem as a transshipment problem, we assume that the commodities can pass through any one of the nodes in the network before they finally reach their destinations. We suppose further that the cost is the same for shipments in opposite directions and unit cost of shipment amongst the sources is 10 while amongst destinations is 5. The transshipment problem is thus changed into the following transportation problem.

	S_1	S_2	S_3	D_1	D_2	
S_1	0	10	10	5	7	200
S_2	10	0	10	12	8	190
S_3	10	10	0	7	9	210
D_1	5	12	7	0	5	150
D_2	7	8	9	5	0	150
	150	150	150	220	230	

Sometimes in a transportation problem, the commodities have to be shipped to an relocation centers before they are shipped to their final destinations. In that case, only the relocation centers can act as both a destination and a source. Clearly, the size of the buffer stock at these relocation centers should be the same as the total supply of the transportation problem. Notice that if the commodities are not allowed to be shipped from source S_i to destination D_j directly, then the unit cost for such a shipment should be set to an arbitrarily large number, i.e. $c_{ij} = M \gg 1$.

Example 6.8. Consider the following transportation network.



The buffer size at the relocation centers should be set to 370. The corresponding transportation tableau is given as follows:

	4	5	6	7	8	9	10	
1			M	M	M	M	M	100
2			M	M	M	M	M	150
3			M	M	M	M	M	120
4	0							370
5		0						370
	370	370	80	50	75	100	65	

6.7 Assignment Problems

Consider assigning n jobs to n machines such that one job is assigned to one machine and one machine gets only one job. Thus the total number of possible assignments is $n!$. A cost c_{ij} is associated with assigning job i to machine j, $i = 1, 2, \cdots, n$; $j = 1, 2, \cdots, n$. The least total cost assignment is then a (zero-one) linear program as follows:

$$\max \qquad x_0 = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{subject to} \begin{cases} \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1, 2, \cdots, n) \quad \text{(one job sets one machines)} \\ \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1, 2, \cdots, n) \quad \text{(one machine gets one job)} \\ \quad x_{ij} = 0, 1 \quad (1 = 1, 2, \cdots, n;\ j = 1, 2, \cdots, n) \end{cases}$$

Notice that both the transportation problem and the assignment problem have *totally unimodualr* coefficient matrices, i.e. the determinants of all their square submatrices are equal to $0, +1$ or -1. This implies that both problems have the integer value property and hence all BFS solutions, in particular, the optimal solutions, are integer-valued even if the integral constraints are discarded, provided that all s_i and d_j are integers in the case of transportation problems.

Standard LP methods and the transportation techniques are applicable for assignment problems with $x_{ij} = 0, 1$ replaced by $0 \leq x_{ij} \leq 1$. However, there is a much more efficient direct method generally known as the *assignment algorithm.* The key observation of the assignment algorithm is that without loss of generality, we may assume that the cost $c_{ij} \geq 0$ for all i, j.

Theorem 6.5. *Let C be the cost matrix with entries c_{ij}, $1 \leq i \leq m$ and $1 \leq j \leq n$. If a constant is added to or subtracted from any row or column of C, giving C'; the minimization of the modified objective function ${x_0}' = \sum_{i,j} {c_{ij}}' x_{ij}$ yields the same solution x_{ij} as the original objective function $x_0 = \sum_{i,j} c_{ij}x_{ij}$.*

Proof. Suppose p_i is added to row i and q_j is subtracted from column j. Then

$$\begin{aligned} {x_0}' &= \sum_{i,j} {c_{ij}}' x_{ij} = \sum_{i,j} (c_{ij} \pm p_i \pm q_j) x_{ij} \\ &= \sum_{i,j} c_{ij}x_{ij} \pm \sum_i p_i \sum_j x_{ij} \pm \sum_j q_j \sum_i x_{ij} \\ &= \sum_{i,j} c_{ij}x_{ij} \pm \big(\sum_i p_i \pm \sum_j q_i\big) \\ &= x_0 + \text{constant} \end{aligned}$$

□

We use this idea to create a new coefficient matrix C' with at least one zero element in each row and in each column, and if using *zero elements only* (or a subset of which) yields a feasible assignment (with total cost = 0, of course), then this assignment is optimal because the total cost of any feasible assignment is nonnegative, since $c'_{ij} \geq 0$ for all ij.

To determine if the zero elements alone can yield a feasible assignment solution, we first cover the cost matrix C' by lines. Define *cover* c to be the minimum number of lines that can cover all zero elements. Then $c \leq n$. If $c = n$, then we have an assignment on only the zero elements. The actual assignment of jobs to machines is obtained by a *trace-back* as follows:

Let

$$z_{ij} \equiv \text{number of zeros in row } i + \text{ column } j,$$

where the (i, j)th entry is zero. Make successive assignments in *increasing* z_{ij} order. Delete row i and column j upon assignment i-j is made.

Example 6.9. Consider the following assignment problem.
Tableau 1. Subtract p_i from row i. $\sum_i p_i = 28$.

5	7	9	$p_1 = 5$
14	10	12	$p_2 = 10$
15	13	16	$p_3 = 13$

Tableau 2. Subtract q_j from column j. $\sum_j q_j = 2$.

0	2	4
4	0	2
2	0	3
$q_1 = 0$	$q_2 = 0$	$q_3 = 2$

Then we have the following tableau with its cover.

		L	
L	0^*	2	2
L	4	0	0^*
	2	0^*	1

Now the number of zero cells is 4 and the cover $c = n = 3$. Using the trace-back algorithm, we have the optimal assignment of $x^*_{11} = x^*_{23} = x^*_{32} = 1$, and $x^*_{ij} = 0$ for all other i, j with $x^*_0 = \sum_i p_i + \sum_j q_j = 30$.

Basically the assignment algorithm approach is a dual method because at any time, p_i and q_j together yield a feasible solution to the dual of assignment problems. This is because $c'_{ij} \geq 0$ implies that $c_{ij} - p_i - q_j \geq 0$ or $p_i + q_j \leq c_{ij}$.

When the cover $c < n$, we can improve the algorithm as follows: Let

$$h \equiv \underset{\{(i,j) | c_{ij} - (p_i + q_j) > 0\}}{\text{Min}} [c_{ij} - (p_i + q_j)] > 0.$$

Set

$$\begin{cases} p_i \leftarrow p_i - h & \text{if } i \text{ is a covered row} \\ p_i \text{ unchanged} & \text{if not} \end{cases}$$

Set

$$\begin{cases} q_j \leftarrow q_j + h & \text{if } j \text{ is } \textit{not} \text{ a covered column} \\ q_j \text{ unchanged} & \text{if covered} \end{cases}$$

Graphically, we have

	covered	uncovered
covered	$+h$	0
uncovered	0	$-h$

Observe that since all zeros are previously covered, all entries after this change remain nonnegative. This algorithm creates at least one more zero entry, which is previously uncovered and positive; while possibly increasing some previously zero entries that are covered by 2 lines. To apply this algorithm, select the smallest uncovered element, subtract that from every *uncovered* element and add that to every element covered by two lines.

Example 6.10. Subtracting constants from rows and columns of the assignment tableau gives:

$$
\begin{array}{|cccc|}
\hline
1 & 4 & 6 & 2 \\
8 & 7 & 10 & 9 \\
4 & 5 & 11 & 7 \\
6 & 7 & 8 & 5 \\
\hline
\end{array}
\quad
\begin{array}{c}
1 \\ 7 \\ 4 \\ \underline{5} \\ \sum = 17
\end{array}
\quad
\begin{array}{cccc}
\hline
\multicolumn{1}{|c}{0} & 3 & 5 & \multicolumn{1}{c|}{2} \\
\multicolumn{1}{|c}{1} & 0 & 3 & \multicolumn{1}{c|}{2} \\
\multicolumn{1}{|c}{0} & 1 & 7 & \multicolumn{1}{c|}{3} \\
\multicolumn{1}{|c}{1} & 2 & 3 & \multicolumn{1}{c|}{0} \\
\hline
0 & 0 & 3 & 0
\end{array}
\;\Big|\; \sum = 3
\qquad
\begin{array}{c|cccc|}
 & L & & & \\
\hline
 & 0 & 3 & 2 & 2 \\
L & 1 & 0 & 0 & 2 \\
 & 0 & 1 & 4 & 3 \\
L & 1 & 2 & 0 & 0 \\
\hline
\end{array}
\;\left(\sum p_i + \sum q_j = 20\right)
$$

Note that the cover $c = 3 < 4 = n$. The smallest uncovered element is $c'_{32} = 1$. Subtracting that from every *uncovered* element and adding that to every element covered by two lines gives:

$$
\begin{array}{c|cccc|c}
 & L & & & & \\
\hline
 & 0 & 3 & 2 & 2 & \\
L & 1 & 0 & 0 & 2 & \\
 & 0 & 1 & 4 & 3 & -h \\
L & 1 & 2 & 0 & 0 & -h \\
\hline
 & & +h & +h & +h &
\end{array}
\quad \xrightarrow{h=1} \quad
\begin{array}{|cccc|}
\hline
\boxed{0} & 2 & 1 & 1 \\
2 & 0 & \boxed{0} & 2 \\
0 & \boxed{0} & 3 & 2 \\
2 & 2 & 0 & \boxed{0} \\
\hline
\end{array}
$$

Now the cover $c = 4 = n$. Assignment in increasing order of number of zeros in row and column gives:

$$x_{11}{}^* = x_{23}{}^* = x_{32}{}^* = x_{44}{}^* = 1,$$

and all other $x_{ij}{}^* = 0$. The minimal cost is given by

$$x_0{}^* = 20 + [3(1) - 2(1)] = 21$$

or alternatively,

$$x_0{}^* = c_{11} + c_{23} + c_{32} + c_{44} = 1 + 10 + 5 + 5 = 21.$$