Operations Research
Semester 5

Ahmad Abu Zainab

Contents

Chapter 1

Linear Programming

A linear programming problem is a problem in the form

$$\max Z = \sum_{i=0}^{n} (c_i) x_i.$$

Subject to

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &\leqslant b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n &\leqslant b_2 \\ &\vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n &\leqslant b_1 \end{aligned}.$$

Where x_i are the decision variables
In matrix form

$$\max Z = \mathbf{c}^T \mathbf{x}.$$

Subject to

$$\mathbf{Ax} \leqslant \mathbf{b}.$$

$$\mathbf{x} \geqslant \mathbf{0}.$$

1.1 Simplex Method

1.1.1 Augmented Form

First we need to convert the problem to the standard form

$$\max Z = \mathbf{c}^T \mathbf{x}.$$

Subject to

$$\mathbf{Ax} = \mathbf{b}.$$

$$\mathbf{x} \geqslant \mathbf{0}.$$

Then we transform the problem to the augmented form by adding slack variables. We add m slack variables to the problem, where m is the number of constraints.

$$\max Z = \mathbf{c}^T \mathbf{x}.$$

Subject to

$$\mathbf{Ax} + \mathbf{Is} = \mathbf{b}.$$

$$\mathbf{x}, \mathbf{s}, \mathbf{b} \geqslant \mathbf{0}.$$

Where $\mathbf{I}$ is the identity matrix and $\mathbf{s}$ is the vector of slack variables.

1.1.2 Basic Feasible Solution

A basic solution is a solution where n of the variables are set to zero and the rest are set to the m values of the corresponding entries in $\mathbf{b}$.
A basic *feasible* solution is a basic solution where all the variables are non-negative, and it corresponds to a vertex of the feasible region. Two adjacent vertices share all but one basic variable.
Variables set to zero are called non-basic variables, and the rest are called basic variables. The choice of basic variables is called the basis.
A basic feasible solution is called degenerate if one of the basic variables is zero.

1.1.3 Initialising the Simplex Method

We set up the initial tableau by adding the slack variables and the objective function to the augmented form, as follows

BV	x_1	x_2	$\cdots$	x_n	s_1	s_2	$\cdots$	s_m	RHS
Z	$-c_1$	$-c_2$	$\cdots$	$-c_n$	0	0	$\cdots$	0	0
s_1	a_{11}	a_{12}	$\cdots$	a_{1n}	1	0	$\cdots$	0	b_1
s_2	a_{21}	a_{22}	$\cdots$	a_{2n}	0	1	$\cdots$	0	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s_m	a_{m1}	a_{m2}	$\cdots$	a_{mn}	0	0	$\cdots$	1	b_m

Let's take an example

$$\max Z = 3x_1 + 5x_2.$$

Subject to

$$\begin{aligned} x_1 \quad\quad &\leqslant 4 \\ 2x_2 &\leqslant 12 . \\ 3x_1 + 2x_2 &\leqslant 18 \end{aligned}$$

$$x_1, x_2 \geqslant 0.$$

First we convert the problem to the augmented form by adding slack variables

$$\boxed{\begin{array}{l} \max Z = 3x_1 + 5x_2 \\ \\ \text{Subject to} \\ \begin{array}{rcl} x_1 \quad\quad + s_1 \quad\quad &=& 4 \\ 2x_2 + \quad s_2 \quad &=& 12 \\ 3x_1 + 2x_2 + \quad\quad s_3 &=& 18 \end{array} \end{array}}$$

Then we set up the initial tableau

BV	x_1	x_2	s_1	s_2	s_3	RHS
Z	-3	-5	0	0	0	0
s_1	1	0	1	0	0	4
s_2	0	2	0	1	0	12
s_3	3	2	0	0	1	18

First, we identify the entering variable. The entering variable is the variable with the most negative coefficient in the objective function. In this case, the entering variable is x_2.

BV	x_1	x_2	s_1	s_2	s_3	RHS
Z	−3	−5	0	0	0	0
s_1	1	0	1	0	0	4
s_2	0	2	0	1	0	12
s_3	3	2	0	0	1	18

.

Then, we identify the leaving variable. The leaving variable is the variable with the smallest ratio of the RHS to the coefficient of the entering variable in the objective function. In this case, the leaving variable is s_2.

BV	x_1	x_2	s_1	s_2	s_3	RHS
Z	−3	−5	0	0	0	0
s_1	1	0	1	0	0	4
s_2	0	2	0	1	0	12
s_3	3	2	0	0	1	18

.

Then we perform the pivot operation on the leaving variable and the entering variable. The pivot operation is performed by dividing the row of the leaving variable by the coefficient of the entering variable in that row, and then subtracting the resulting row from the other rows, multiplied by the coefficient of the entering variable in that row.
After the pivot operation, the tableau becomes

BV	x_1	x_2	s_1	s_2	s_3	RHS
Z	−3	0	0	2.5	0	30
s_1	1	0	1	0	0	4
x_2	0	1	0	0.5	0	6
s_3	3	0	0	−1	1	6

.

Finally, we repeat the process until the objective function has no negative coefficients.

Note:-

The *optimality test* is as follows

- If the objective function has no negative coefficients, then the current solution is optimal.
- If the objective function has negative coefficients, then the current solution is not optimal, and we repeat the process.

In our case, after iterating another time (x_1 entering and s_3 leaving), we get

BV	x_1	x_2	s_1	s_2	s_3	RHS
Z	0	0	0	1.5	1	36
s_1	0	0	1	1/3	−1/3	2
x_2	0	1	0	0.5	0	6
x_1	1	0	0	−1/3	1/3	2

.

Since the objective function has no negative coefficients, the current solution is optimal. The optimal solution is $x_1 = 2, x_2 = 6, s_1 = 2, s_2 = 0, s_3 = 0$ and $Z = 36$.

1.1.4 Unbounded Solution

In a given tableau, if the entering variable has all zero entries in its column, then the solution is unbounded. Which means that the objective function can be increased indefinitely.

1.1.5 Alternative Optimal Solutions

Alternative optimal solutions occur when one of the non-basic variables has a zero coefficient in the objective function in the final tableau. In this case, the solution is degenerate.

1.1.6 Non-Standard Form

$$0.4x_1 - 0.3x_2 \geqslant -10 \overset{\times -1}{\Longrightarrow} -0.4x_1 + 0.3x_2 \leqslant 10$$
$$\min Z = 0.4x_1 + 0.3x_2 \overset{\times -1}{\Longrightarrow} \max Z = -0.4x_1 - 0.3x_2$$

1.1.7 Artificial Variables

Consider the following linear system

$$\mathbf{Ax} = \mathbf{b}.$$

and

$$\mathbf{x} \geqslant \mathbf{0}.$$

We can use the simplex method to solve this system by adding artificial variables to the system, as follows

$$\mathbf{Ax} + \mathbf{Ia} = \mathbf{b}.$$

$$\mathbf{x}, \mathbf{a} \geqslant \mathbf{0}.$$

Where $\mathbf{I}$ is the identity matrix and $\mathbf{a}$ is the vector of artificial variables.
Take the following example

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 4 \\ x_1 + x_2 \phantom{{}+x_3} &= 3 \end{aligned}.$$

The augmented form is

$$\begin{aligned} x_1 + 2x_2 + x_3 + a_1 \phantom{{}+a_2} &= 4 \\ x_1 + x_2 \phantom{{}+x_3} + a_2 &= 3 \end{aligned}.$$

And we can solve it using the simplex method.

Two-Phase Method

In the case where the basic feasible solution isn't very obvious, we can use the two-phase method to find the basic feasible solution by adding artificial variables to the system, as follows

$$\mathbf{Ax} + \mathbf{Is} + \mathbf{Ia} = \mathbf{b}.$$

$$\mathbf{x}, \mathbf{s}, \mathbf{a} \geqslant \mathbf{0}.$$

Where $\mathbf{I}$ is the identity matrix and $\mathbf{s}$ is the vector of slack variables and $\mathbf{a}$ is the vector of artificial variables.
In the first phase we solve the problem starting from the BFS where all the artificial variables are basic variables. If the problem is feasible, we will be able to find a BFS where all artificial variables are 0. This automatically gives us a BFS for the original problem.
For example, consider the following problem

$$\min Z = 2x_1 + 3x_2.$$

Subject to

$$\begin{aligned} x_1 + 2x_2 &= 4 \\ 2x_1 - x_2 &= 3 \end{aligned}.$$

We can convert it to the following augmented form

$$\begin{aligned} x_1 + 2x_2 \phantom{{}+a_2} + a_1 &= 4 \\ 2x_1 - x_2 + a_2 \phantom{{}+a_1} &= 3 \end{aligned}.$$

We define a new objective function

$$\max W = -a_1 - a_2.$$

The initial tableau is set up as follows

BV	x_1	x_2	a_1	a_2	RHS
W	-3	-1	0	0	7
Z	-2	-3	0	0	0
a_1	1	2	1	0	4
a_2	2	-1	0	1	3

The resulting solution is a BFS for the original problem. In the second phase we solve the original problem starting from the BFS we found in the first phase.

Big-M Method

The Big-M method is a variation of the two-phase method where we add a large number M to the objective function for each artificial variable. This ensures that the artificial variables will be set to zero in the optimal solution.
The objective function looks like this

$$\max Z = \mathbf{c}^T\mathbf{x} - M\mathbf{a}.$$

The initial simplex tableau is set up as follows

BV	x_1	x_2	$\cdots$	x_n	s_1	s_2	$\cdots$	s_m	RHS
Z	$-c_1$	$-c_2$	$\cdots$	$-c_n$	M	M	$\cdots$	M	0
s_1	a_{11}	a_{12}	$\cdots$	a_{1n}	1	0	$\cdots$	0	b_1
s_2	a_{21}	a_{22}	$\cdots$	a_{2n}	0	1	$\cdots$	0	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s_m	a_{m1}	a_{m2}	$\cdots$	a_{mn}	0	0	$\cdots$	1	b_m

1.1.8 Infeasible LPs

An LP is infeasible if in the final tablea there are still artificial variables that are basic variables.

1.1.9 Constraints with $\geqslant$

If we have a contraint with $\geqslant$ and a negative RHS we can multiply the constraint by -1 and proceed as usual. However, if we have a constraint with $\geqslant$ and a positive RHS we turn the constraint into an equality by adding a slack variable as seen below

$$x_1 - 2x_2 + x_3 \geqslant 20 \quad\implies\quad x_1 - 2x_2 + x_3 - s_1 = 20.$$

Where $s_1 > 0$.

But for initial BFS we cannot simplex s_1 in to the base so we need to add an artificial variable a_1 to the contraint as if it were an equality contraint

$$x_1 - 2x_2 + x_3 - s_1 + a_1 = 20.$$

Where $a_1 > 0$.

1.1.10 Variables Unconstrained in Sign

If we have a variable that is unconstrained in sign we can split it into two variables ${x_1}^+$ and ${x_1}^-$ where $x_1 = {x_1}^+ - {x_1}^-$.

$${x_1}^+ = \max(0, x_1) \quad {x_1}^- = \max(0, -x_1).$$

1.1.11 Maximize the Minimum

Some real life problems consist of maximizing the lowest of many economic functions.

$$\max Z = \min(3x_1 + 2x_2, x_1 - 2x_3).$$

We can solve this problem by adding a new variable u and adding the following constraints

$$\begin{array}{r} 3x_1 + 2x_2 - u \geqslant 0 \\ x_1 - 2x_3 - u \geqslant 0 \end{array}.$$

where u is unconstrained in sign.

Chapter 2

Network Optimization Models

2.1 Max Flow Problem

Consider a directed graph $G = (V, E)$ with a source node s and a sink node t. Each edge $(i, j) \in E$ has a capacity u_{ij}. The max flow problem is to find the maximum flow from s to t.

LP Model

$$\max Z = \sum_{i \in \delta^+(s)} x_{si}.$$

or

$$\max Z = \sum_{i \in \delta^-(t)} x_{it}.$$

where $\delta^+(s)$ is the set of nodes that are flowing out of to s and $\delta^-(t)$ is the set of nodes that are flowing in to t.

Subject to

$$-u_{ij} \leqslant x_{ij} \leqslant u_{ij}.$$

and

$$\sum_{i \in \delta^+(j)} x_{ij} = \sum_{i \in \delta^-(j)} x_{ji} \quad \forall j \in V \setminus \{s, t\}.$$

2.2 Min Cost Flow Problem

Consider a directed graph $G = (V, E)$. Each edge $(i, j) \in E$ has a capacity u_{ij}, a cost c_{ij}, and a supply b_i. The min cost flow problem is to find the minimum cost flow within this network.

LP Model

$$\min Z = \sum_{(i,j) \in E} c_{ij} x_{ij}.$$

Subject to

$$\sum_{i \in \delta^+(j)} x_{ji} - \sum_{i \in \delta^-(j)} x_{ij} = b_j \quad \forall j \in V.$$

and

$$x_{ij} \leqslant u_{ij} \quad \forall (i, j) \in E.$$

Note:-

The supply b_i is positive if the node is a source, negative if the node is a sink, and zero if the node is a transshipment node.

Note:-

If the capacity u_{ij} is not specified, we can assume it is ∞. (The edge is unconstrained in capacity)

2.2.1 Special Cases

Transportation Problem

Sources have $b_i > 0$, sinks have $b_i < 0$, and transshipment nodes have $b_i = 0$. All nodes have unlimited capacity, and edges all have a cost c_{ij}.

Shortest Path Problem

Sources have $b_i = 1$, sinks have $b_i = -1$, and transshipment nodes have $b_i = 0$. All nodes have unlimited capacity, and edges all have a cost c_{ij}.

Sink Tree Problem

Sources have b_i = number of nodes in the network, and at every other node $b_i = -1$. All nodes have unlimited capacity, and edges all have a cost c_{ij}.

Max Flow Problem

At the source we set $b = M$, where M is a very large number, and we set $b = -M$ for target nodes. We then create a dummy node d and connect it to the source and target nodes with unlimited capacity and cost 1. The simplex algorithm will route flow through the non-dummy links since they have 0 cost.

Theorem 2.2.1 Integrality Theorem

If all b_i and u_{ij} are integers, then there exists an optimal solution where all x_{ij} are integers.

2.3 Shortest Path Problem

Given a directed graph $G = (V, E)$ with a source node s and a sink node t. Each edge $(i, j) \in E$ has a cost c_{ij}. The shortest path problem is to find the shortest path (minimum cost) from s to t.

LP Model

$$\min Z = \sum_{(i,j)\in E} c_{ij}x_{ij}.$$

Subject to

$$\sum_{i\in\delta^+(j)} x_{ji} - \sum_{i\in\delta^-(j)} x_{ij} = \begin{cases} 1 & \text{if } j = s \\ -1 & \text{if } j = t \\ 0 & \text{otherwise} \end{cases}.$$

where x_{ij} is a binary value (0 or 1).

2.3.1 Dijkstra's Algorithm

Given the same setup as the shortest path problem, we can solve it using Dijkstra's algorithm. The algorithm is as follows

1. Set $d(s) = 0$ and $d(i) = \infty$ for all other nodes.
2. Set $S = \emptyset$ (the set of visited nodes).
3. Set $i = s$.
4. Find the node j with the smallest $d(j)$ such that $j \notin S$.
5. Add j to S.
6. For all $k \notin S$, set $d(k) = \min(d(k), d(j) + c_{jk})$.
7. If $j \neq t$, set $i = j$ and go to step 4.
8. Repeat step 4 until $S = V$.

2.4 Minimum Spanning Tree Problem

Given a connected undirected graph $G = (V, E)$ with a cost c_{ij} for each edge $(i, j) \in E$. The minimum spanning tree problem is to find the minimum cost tree that connects all nodes.

We can solve this using an algorithm called Prim's algorithm. The algorithm is as follows

1. Set $S = \{s\}$ (the set of visited nodes).
2. Set $i = s$.
3. Find the edge (i, j) with the smallest c_{ij} such that $j \notin S$.
4. Add j to S.
5. If $S = V$, stop.
6. Set $i = j$ and go to step 3.