Linear programming

Source: Ahmadi, ORF 363 slides.

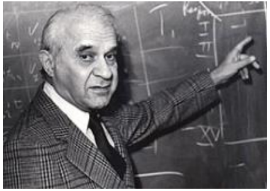
Leontief input-output model

Applications of linear programming

History of Linear Programming

Leontief input-output model of an economy

- The Leontief input-output model breaks a nation's economy into sectors (so-called producing sectors)
  - Agriculture
  - Manufacturing
  - Services
  - Education



Wassily Leontief
1906-1999

Leontief input-output model of an economy

- The Leontief input-output model breaks a nation's economy into sectors (so-called producing sectors)
  - Agriculture
  - Manufacturing
  - Services
  - Education

- Each sector needs the output of the other sectors in order to produce its own output

- Each sector should produce to meet the demand of the other sectors, as well as the demand of the society

Consumption matrix

	Transportation	Agriculture	Services	Manufacturing
Transportation	.2	.3	.5	.3
Agriculture	.5	.3	.1	.0
Services	.1	.2	.2	.5
Manufacturing	.1	.1	.1	.1

This is called the **consumption matrix**, denoted by C

Consumption matrix

	Transportation	Agriculture	Services	Manufacturing
Transportation	.2	.3	.5	.3
Agriculture	.5	.3	.1	.0
Services	.1	.2	.2	.5
Manufacturing	.1	.1	.1	.1

This is called the **consumption matrix**, denoted by C

In order to produce one unit of transportation, the transportation industry needs to consume .2 units of transportation itself, .5 units of agriculture, .1 units of services, and .1 units of manufacturing

Leontief production equation

- Let $d = (d_1, d_2, d_3, d_4)^T$ be a vector denoting the demand of the society for each of the producing sectors

Leontief production equation

- Let $d = (d_1, d_2, d_3, d_4)^T$ be a vector denoting the demand of the society for each of the producing sectors

- So we are trying to solve the following equation:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} .2 & .3 & .5 & .3 \\ .5 & .3 & .1 & .0 \\ .1 & .2 & .2 & .5 \\ .1 & .1 & .1 & .1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix}$$

”Amount produced = intermediate demand + final demand”

Leontief production equation

- Let $d = (d_1, d_2, d_3, d_4)^T$ be a vector denoting the demand of the society for each of the producing sectors

Leontief production equation

- Let $d = (d_1, d_2, d_3, d_4)^T$ be a vector denoting the demand of the society for each of the producing sectors

- So we are trying to solve the following equation:

  "Amount produced = intermediate demand + final demand"

$$x = Cx + d$$

This is called the **Leontief production equation**

Leontief production equation

- For a given C and d, we need to solve the following linear system to figure out how much each sector should produce:
$$(I - C)x = d$$

Leontief production equation

- For a given C and d, we need to solve the following linear system to figure out how much each sector should produce:
$$(I - C)x = d$$
- An economy is called ”productive” if for every demand vector d there exists a nonnegative production vector x satisfying the above linear system. This is a property of the consumption matrix only.

Linear programming

- A linear program is an optimization problem of the form:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ \text{and} \quad & x \geq 0 \end{aligned}$$

  where $c \in \mathbb{R}^n, b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$.

- This is called a linear program in standard form

Linear programming

- A linear program is an optimization problem of the form:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ \text{and} \quad & x \geq 0 \end{aligned}$$

  where $c \in \mathbb{R}^n, b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$.

- This is called a linear program in standard form

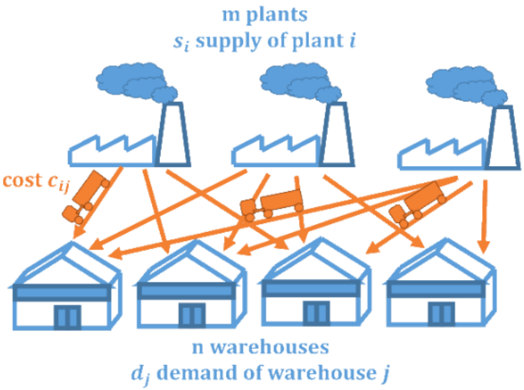
- Not all linear programs appear in this form but we will see later that they can all be rewritten in this form

Leontief input-output model

Applications of linear programming

History of Linear Programming

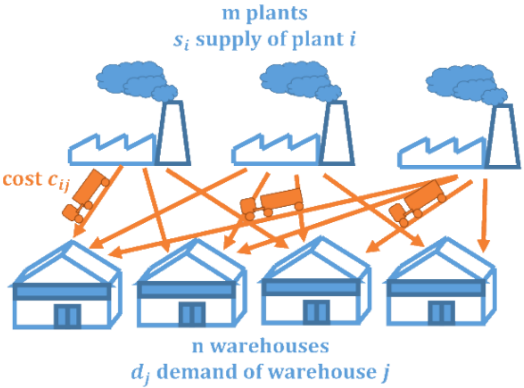
Transportation problem



- All plants produce product A (in different quantities) and all warehouses need product A (also in different quantities)

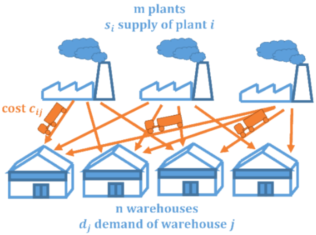
- The cost of transporting one unit of product A from i to j is c_{ij}

Transportation problem



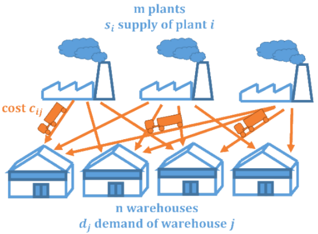
- We want to minimize the total cost of transporting product A while still fulfilling the demand from the warehouses and without exceeding the supply produced by the plants

Transportation problem



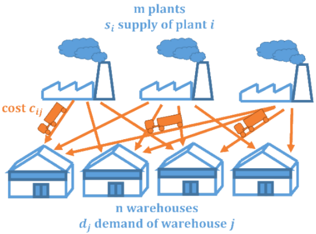
- Decision variables: x_{ij}, quantity transported from i to j

Transportation problem



- Decision variables: x_{ij}, quantity transported from i to j
- The objective function to minimize: $\sum_{ij} c_{ij}x_{ij}$
  The constraints are:

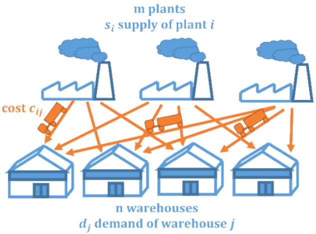
Transportation problem



- Decision variables: x_{ij}, quantity transported from i to j

- The objective function to minimize: $\sum_{ij} c_{ij} x_{ij}$
  The constraints are:
  - Not exceed the supply in any factory: $\sum_{j=1}^{n} x_{ij} \leq s_i, i = 1, 2, 3$

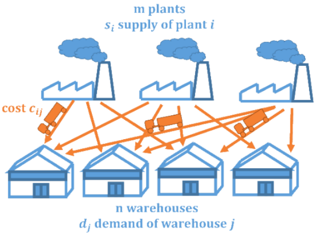
Transportation problem



- Decision variables: x_{ij}, quantity transported from i to j

- The objective function to minimize: $\sum_{ij} c_{ij} x_{ij}$
  The constraints are:
  - Not exceed the supply in any factory: $\sum_{j=1}^{n} x_{ij} \leq s_i, i = 1, 2, 3$
  - Fulfill needs of the warehouses: $\sum_{i=1}^{n} x_{ij} \geq d_j$

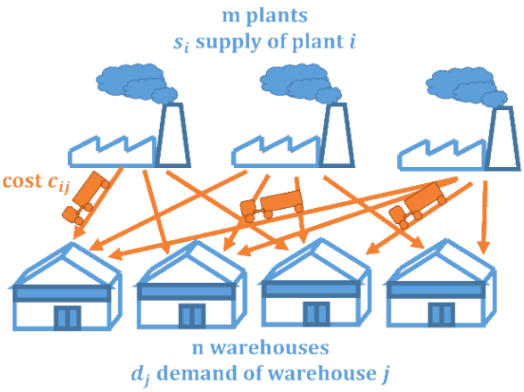
Transportation problem



- Decision variables: x_{ij}, quantity transported from i to j

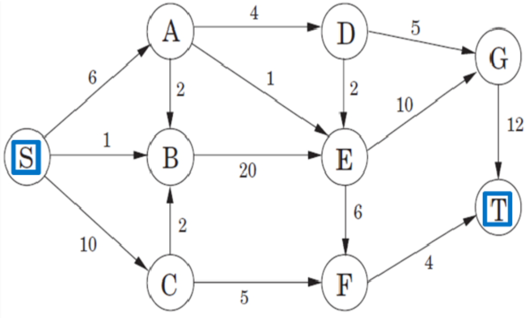
- The objective function to minimize: $\sum_{ij} c_{ij}x_{ij}$
  The constraints are:
  - Not exceed the supply in any factory: $\sum_{j=1}^{n} x_{ij} \leq s_i, i = 1, 2, 3$
  - Fulfill needs of the warehouses: $\sum_{i=1}^{n} x_{ij} \geq d_j$
  - Quantity must be non-negative: $x_{ij} \geq 0$ for all i, j

Transportation problem



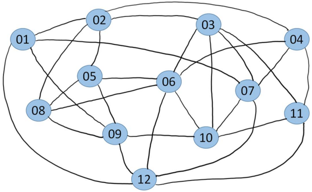
- We want to minimize the total cost of transporting product A while still fulfilling the demand from the warehouses and without exceeding the supply produced by the plants

The maximum flow problem



- The goal is to ship as much oil as possible from S to T
- We cannot exceed the capacities on the edges
- No storage at the nodes: for every node, flow in=flow out

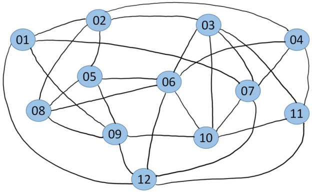
LP relaxation for the largest independent set problem



- Find the largest collection of nodes among which no two share an edge

- We can write this problem as a linear program with integer constraints
  Such a problem is called **integer program**

LP relaxation for the largest independent set problem



- Integer programs (IPs) are in general difficult to solve. An easier problem in the **LP relaxation** of this problem by replacing the constraint $x_i(1-x_i)=0$ with $0 \le x_i \le 1$

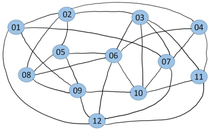
Minimize $x_1 + x_2 + \ldots + x_{12}$ s.t. $x_1 + x_2 \le 1$

$$x_1 + x_8 \le 1$$

$$\vdots$$

~~$x_i(1-x_i)=0$~~ $0 \le x_i \le 1$ for all i

LP relaxation for the largest independent set problem



Minimize $x_1 + x_2 + \ldots + x_{12}$ s.t. $x_1 + x_2 \leq 1$

$$x_1 + x_8 \leq 1$$

$$\vdots$$

~~$x_i(1 - x_i) = 0$~~ $0 \leq x_i \leq 1$ for all i

The optimal solution to the LP (denoted by OPT_{LP}), is an upperbound to the optimal solution to the IP (denoted by OPT_{IP}):

$$OPT_{IP} \leq OPT_{LP}$$

Scheduling nurses

- A hospital wants to start weekly nightshifts for its nurses. The goal is to hire the fewest number of nurses possible
  - There is demand for nurses on days $j = 1, ...7$
  - Each nurse wants to work 5 consecutive days if possible

- How many nurses should we hire?

Scheduling nurses

- A hospital wants to start weekly nightshifts for its nurses. The goal is to hire the fewest number of nurses possible
  - There is demand for nurses on days $j = 1, ...7$
  - Each nurse wants to work 5 consecutive days if possible

- How many nurses should we hire?
  - The decision variables here will be $x_1, x_2, \dots, x_7$ where x_j is the number of nurses hired for day j

Scheduling nurses

- A hospital wants to start weekly nightshifts for its nurses. The goal is to hire the fewest number of nurses possible
  - There is demand for nurses on days $j = 1, ...7$
  - Each nurse wants to work 5 consecutive days <u>if possible</u>

- How many nurses should we hire?
  - The decision variables here will be $x_1, x_2, \ldots, x_7$ where x_j is the number of nurses hired for day j
  - The objective is to minimize the total number of nurses:

$$\sum_{j=1}^{7} x_j$$

Scheduling nurses

- How many nurses should we hire?
  - The constraints take into account the demand for each day but also the fact that the nurses want to work 5 consecutive days. This means that if the nurses work on day 1, they will work all the way through day 5

$$
\begin{aligned}
x_1 & \geq d_1 \\
x_1 + x_2 & \geq d_2 \\
x_1 + x_2 + x_3 & \geq d_3 \\
x_1 + x_2 + x_3 + x_4 & \geq d_4 \\
x_1 + x_2 + x_3 + x_4 + x_5 & \geq d_5 \\
x_2 + x_3 + x_4 + x_5 + x_6 & \geq d_6 \\
x_3 + x_4 + x_5 + x_6 + x_7 & \geq d_7
\end{aligned}
$$

Scheduling nurses

$$
\begin{aligned}
\text{Minimize } & \sum_{j=1}^{7} x_j \\
s.t.\ & x_1 && \geq d_1 \\
& x_1 + x_2 && \geq d_2 \\
& x_1 + x_2 + x_3 && \geq d_3 \\
& x_1 + x_2 + x_3 + x_4 && \geq d_4 \\
& x_1 + x_2 + x_3 + x_4 + x_5 && \geq d_5 \\
& \quad x_2 + x_3 + x_4 + x_5 + x_6 && \geq d_6 \\
& \qquad x_3 + x_4 + x_5 + x_6 + x_7 && \geq d_7
\end{aligned}
$$

- This is also an IP programming, which is difficult to solve. The LP relaxation of this IP is the following:

$$x_1, x_2, \ldots, x_7 \geq 0$$

-

We have $OPT_{LP} \leq OPT_{IP}$

Leontief input-output model

Applications of linear programming

History of Linear Programming

History of linear programming

- Solving systems of linear inequalities goes at least as far back as the late 1700s, when Fourier invented a (pretty inefficient) solution technique, known today as the ”Fourier-Motzkin” elimination method

History of linear programming

- Solving systems of linear inequalities goes at least as far back as the late 1700s, when Fourier invented a (pretty inefficient) solution technique, known today as the ”Fourier-Motzkin” elimination method

- In 1930s, Kantorovich and Koopmans brought new life to linear programming by showing its widespread applicability in resource allocation problems. They jointly received the Nobel Prize in Economics in 1975.

History of linear programming

John von Neumann (1903-1957)   George Dantzig (1914-2005)

- Von Neumann is often credited with the theory of **LP duality** (the topic of our next lecture)

History of linear programming

John von Neumann (1903-1957)

George Dantzig (1914-2005)

- Von Neumann is often credited with the theory of **LP duality** (the topic of our next lecture)

- In 1947, Dantzig invented the first practical algorithm for solving LPs: the **simplex method**. This essentially revolutionized the use of linear programming in practice

History of linear programming

Leonid Khachiyan (1952 - 2005)   Narendra Karmarkar (b. 1957)

- In 1979, Khachiyan showed that LPs were solvable in polynomial time using the **ellipsoid method**. This was a theoretical breakthrough more than a practical one, as in practice the algorithm was quite slow.

History of linear programming

Leonid Khachiyan (1952 - 2005)   Narendra Karmarkar (b. 1957)

- In 1979, Khachiyan showed that LPs were solvable in polynomial time using the **ellipsoid method**. This was a theoretical breakthrough more than a practical one, as in practice the algorithm was quite slow.
- In 1984, Karmarkar developed the **interior point method**, another polynomial time algorithm for LPs, which was also efficient in practice. Along with the simplex method, this is the method of choice today for solving LPs