ORF307 – Optimization

16. Network optimization
Ed forum

• How did we arrive at \( p^*(u) = \min \{ c'x \mid Ax = b + u, x \geq 0 \} \)?

• Do we have an efficient way of finding the solution to the modified problem if \( u \) is large enough that the basis changes, or do we just have to start the simplex method from scratch?
Recap
Changes in problem data

**Goal:** extract information from $x^*, y^*$ about their sensitivity with respect to changes in problem data

**Modified LP**

minimize \[ c^T x \]

subject to \[ Ax = b + u \]

\[ x \geq 0 \]

Optimal cost $p^*(u)$
Global sensitivity

Dual of modified LP

maximize \(- (b + u)^T y\)
subject to \(A^T y + c \geq 0\)
Global sensitivity

Dual of modified LP

maximize \(- (b + u)^T y\)
subject to \(A^T y + c \geq 0\)

Global lower bound

Given \(y^*\) a dual optimal solution for \(u = 0\), then

\[
p^*(u) \geq -(b + u)^T y^* = p^*(0) - u^T y^*
\]

(from weak duality and dual feasibility)
Global sensitivity

Dual of modified LP

maximize \(- (b + u)^T y\)

subject to \(A^T y + c \geq 0\)

Global lower bound

Given \(y^*\) a dual optimal solution for \(u = 0\), then

\[p^*(u) \geq - (b + u)^T y^*\]

\[= p^*(0) - u^T y^*\]

(from weak duality and dual feasibility)

It holds for any \(u\)
Local sensitivity

in neighborhood of the origin

Original LP

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
\text{and} & \quad x \geq 0 
\end{align*}
\]

Optimal solution

Primal

\[
\begin{align*}
x_i &= 0, \quad i \notin B \\
x_B^* &= A_B^{-1} b 
\end{align*}
\]

Dual

\[
\begin{align*}
y^* &= -A_B^{-T} c_B 
\end{align*}
\]
Local sensitivity

\( u \) in neighborhood of the origin

Original LP

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

Optimal solution

Primal

\[
x_i^* = 0, \quad i \notin B
\]

Dual

\[
y^* = -A_B^{-T}c_B
\]

Modified LP

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b + u \\
& \quad x \geq 0
\end{align*}
\]

Modified dual

\[
\begin{align*}
\text{maximize} & \quad -(b + u)^T y \\
\text{subject to} & \quad A^T y + c \geq 0
\end{align*}
\]

Optimal basis does not change
Local sensitivity

$u$ in neighborhood of the origin

Original LP

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

Optimal solution

$$\begin{align*}
\text{Primal} & \quad x^*_i = 0, \quad i \notin B \\
& \quad x^*_B = A_B^{-1}b \\
\text{Dual} & \quad y^* = -A_B^{-T}c_B
\end{align*}$$

Modified LP

$$\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b + u \\
& \quad x \geq 0
\end{align*}$$

Modified dual

$$\begin{align*}
\text{maximize} & \quad -(b + u)^T y \\
\text{subject to} & \quad A^T y + c \geq 0
\end{align*}$$

Optimal basis does not change

Modified optimal solution

$$\begin{align*}
& \quad x^*_B(u) = A_B^{-1}(b + u) = x^*_B + A_B^{-1}u \\
& \quad y^*(u) = y^*
\end{align*}$$
Derivative of the optimal value function

Modified optimal solution

\[ x^*_B(u) = A_B^{-1}(b + u) = x^*_B + A_B^{-1}u \]
\[ y^*(u) = y^* \]
Derivative of the optimal value function

Modified optimal solution

\[ x^*_B(u) = A_B^{-1}(b + u) = x^*_B + A_B^{-1}u \]
\[ y^*(u) = y^* \]

Optimal value function

\[ p^*(u) = c^T x^*(u) \]
\[ = c^T x^* + c_B^T A_B^{-1}u \]
\[ = p^*(0) - y^{*T}u \] (affine for small \( u \))
Derivative of the optimal value function

Modified optimal solution
\[
x_B^*(u) = A_B^{-1}(b + u) = x_B^* + A_B^{-1}u
\]
\[
y^*(u) = y^*
\]

Optimal value function
\[
p^*(u) = c^T x^*(u)
\]
\[
= c^T x^* + c_B^T A_B^{-1}u
\]
\[
= p^*(0) - y^*^T u \quad \text{(affine for small } u)\]

Local derivative
\[
\nabla p^*(u) = -y^* \quad \text{(}y^*\text{ are the shadow prices)}
\]
Today’s lecture
Network optimization

• Network flows
• Minimum cost network flow problem
• Network flow solutions
• Examples: maximum flow, shortest path, assignment
Network flows
Networks

- Electrical and power networks
- Road networks
- Airline routes
- Printed circuit boards
- Social networks
Network modelling

A network (or directed graph, or digraph) is a set of $m$ nodes and $n$ directed arcs

- Arcs are ordered pairs of nodes $(a, b)$ (leaves $a$, enters $b$)
- **Assumption** there is at most one arc from node $a$ to node $b$
- There are no loops (arcs from $a$ to $b$)
Arc-node incidence matrix

$m \times n$ matrix $A$ with entries

$$A_{ij} = \begin{cases} 1 & \text{if arc } j \text{ starts at node } i \\ -1 & \text{if arc } j \text{ ends at node } i \\ 0 & \text{otherwise} \end{cases}$$

**Note** Each column has one $-1$ and one $1$
Arc-node incidence matrix

$m \times n$ matrix $A$ with entries

$$A_{ij} = \begin{cases} 1 & \text{if arc } j \text{ starts at node } i \\ -1 & \text{if arc } j \text{ ends at node } i \\ 0 & \text{otherwise} \end{cases}$$

**Note** Each column has one $-1$ and one $1$
Network flow

**flow vector** $x \in \mathbb{R}^n$

$x_j$: flow (of material, traffic, information, electricity, etc) through arc $j$
Network flow

flow vector $x \in \mathbb{R}^n$

$x_j$: flow (of material, traffic, information, electricity, etc) through arc $j$

total flow leaving node $i$

$$\sum_{j=1}^{n} A_{ij}x_j = (Ax)_i$$

Diagram:
- $x_j$ pointing out of node $i$
- $x_k$ pointing into node $i$
- $A_{ij} = -1$
- $A_{ik} = 1$
External supply

**supply vector** $b \in \mathbb{R}^m$

- $b_i$ is the external supply at node $i$ (if $b_i < 0$, it represents demand)
- We must have $1^T b = 0$ (total supply = total demand)
External supply

**supply vector** $b \in \mathbb{R}^m$

- $b_i$ is the external supply at node $i$ (if $b_i < 0$, it represents demand)
- We must have $1^T b = 0$ (total supply = total demand)

**Balance equations**

$$\sum_{j=1}^{n} A_{ij} x_j = (Ax)_i = b_i, \quad \text{for all } i$$
External supply

**supply vector** $b \in \mathbb{R}^m$

- $b_i$ is the external supply at node $i$ (if $b_i < 0$, it represents demand)
- We must have $1^T b = 0$ (total supply = total demand)

**Balance equations**

$$\sum_{j=1}^{n} A_{ij} x_j = (Ax)_i = b_i, \quad \text{for all } i$$

Total leaving flow
External supply

**Supply vector** $b \in \mathbb{R}^m$

- $b_i$ is the external supply at node $i$ (if $b_i < 0$, it represents demand)
- We must have $1^T b = 0$ (total supply = total demand)

**Balance equations**

$$\sum_{j=1}^{n} A_{ij} x_j = (Ax)_i = b_i, \quad \text{for all } i$$

- Total leaving flow
- Supply
External supply

**supply vector** \( b \in \mathbb{R}^m \)

- \( b_i \) is the external supply at node \( i \)
  (if \( b_i < 0 \), it represents demand)
- We must have \( 1^T b = 0 \)
  (total supply = total demand)

**Balance equations**

\[
\sum_{j=1}^{n} A_{ij} x_j = (Ax)_i = b_i, \quad \text{for all } i
\]

Total leaving flow \quad Supply

\[Ax = b\]
Minimum cost network flow problem
Minimum cost network flow problem

minimize  \( c^T x \)
subject to  \( Ax = b \)
\[ 0 \leq x \leq u \]

- \( c_i \) is unit cost of flow through arc \( i \)
- Flow \( x_i \) must be nonnegative
- \( u_i \) is the maximum flow capacity of arc \( i \)
- Many network optimization problems are just special cases
Example
Transportation

Goal ship \( x \in \mathbb{R}^n \) to satisfy demand

Supply \quad Demand

7 \rightarrow 5 \quad 6 \rightarrow -10

11 \rightarrow 8 \quad 4 \rightarrow -23

18 \rightarrow 9 \quad 3 \rightarrow -15

12 \rightarrow 3 \quad 6 \rightarrow

(arc costs shown)
All capacities 20
Example
Transportation

Goal ship $x \in \mathbb{R}^n$ to satisfy demand

Supply

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>11</th>
<th>18</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

Demand

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-10</td>
</tr>
<tr>
<td></td>
<td>-23</td>
</tr>
<tr>
<td></td>
<td>-15</td>
</tr>
</tbody>
</table>

(arc costs shown)

All capacities 20

$c = (5, 6, 8, 4, 3, 9, 3, 6)$

$$A = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
-1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & -1
\end{bmatrix}$$

$b = (7, 11, 18, 12, -10, -23, -15)$

$u = 20$
Example
Transportation

Goal: ship $x \in \mathbb{R}^n$ to satisfy demand

Supply | Demand
---|---
7 | 5
11 | 8
18 | 9
12 | 6

\(-10\) \hspace{2cm} \(-23\) \hspace{2cm} \(-15\)

(arc costs shown)
All capacities 20

$\mathbf{A} = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
-1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & -1
\end{bmatrix}$

$c = (5, 6, 8, 4, 3, 9, 3, 6)$

$b = (7, 11, 18, 12, -10, -23, -15)$

$u = 20 \mathbf{1}$

Minimum cost network flow

minimize \( c^T x \)

subject to \( Ax = b \)

\( 0 \leq x \leq u \)

\( x^* = (7, 0, 3, 0, 8, 18, 5, 7) \)
Example

Airline passenger routing

• United Airlines has 5 flights per day from BOS to NY (10am, 12pm, 2pm, 4pm, 6pm)

• Flight capacities (100, 100, 100, 150, 150)

• Costs: $50/hour of delay

• Last option: 9pm flight with other company (additional cost $75)

• Today’s reservations (110, 118, 103, 161, 140)
Airline passenger routing

Network

10am → 12am → 2pm → 4pm → 6pm

NY → 9pm
Airline passenger routing

Network

Decisions

$x_j$: passengers flowing on arc $j$
Airline passenger routing

Network

Decisions
\( x_j \): passengers flowing on arc \( j \)

Costs
\( c_j \): cost of moving passenger on arc \( j \)
- Between flights: $50/hour
- To 9pm flight: $75 additional
- To NY: $0 (as scheduled)
Airline passenger routing

Network

Decisions
$x_j$: passengers flowing on arc $j$

Costs
$c_j$: cost of moving passenger on arc $j$
  • Between flights: $50$/hour
  • To 9pm flight: $75$ additional
  • To NY: $0$ (as scheduled)

Supplies
$b_i$ reserved passengers for flight $i$
  • 9pm flight: $b_i = 0$
  • NY supply: $- \text{total reserved passeng.}$
**Airline passenger routing**

### Network

![Diagram of airline passenger routing]

### Decisions

\( x_j \): passengers flowing on arc \( j \)

### Costs

\( c_j \): cost of moving passenger on arc \( j \)
- Between flights: $50/hour
- To 9pm flight: $75 additional
- To NY: $0 (as scheduled)

### Supplies

\( b_i \) reserved passengers for flight \( i \)
- 9pm flight: \( b_i = 0 \)
- NY supply: - total reserved passengers.

### Capacities

\( u_j \) maximum passengers over arc \( j \)
- Between flights: \( u_j = \infty \)
- To NY: \( u_i = \text{flight capacity} \)
Airline passenger routing

Network flow formulation
minimize \( c^T x \)
subject to \( Ax = b \)
\( 0 \leq x \leq u \)

Decisions
\( x_j \): passengers flowing on arc \( j \)

Costs
\( c_j \): cost of moving passenger on arc \( j \)
- Between flights: $50/hour
- To 9pm flight: $75 additional
- To NY: $0 (as scheduled)

Supplies
\( b_i \) reserved passengers for flight \( i \)
- 9pm flight: \( b_i = 0 \)
- NY supply: - total reserved passeng.

Capacities
\( u_j \) maximum passengers over arc \( j \)
- Between flights: \( u_j = \infty \)
- To NY: \( u_i = \) flight capacity
Network flow solutions
Remove arc capacities

**Goal:** create equivalent network without arc capacities

minimize \( c^T x \)
subject to \( Ax = b \)
\( 0 \leq x \leq u \)
Remove arc capacities

**Goal**: create equivalent network without arc capacities

minimize $c^T x$
subject to $Ax = b$
$0 \leq x \leq u$

$\tilde{c}^T \tilde{x}$
$\tilde{A}\tilde{x} = \tilde{b}$
$\tilde{x} \geq 0$

Standard form LP with arc-node incidence matrix
Remove arc capacities

Idea: slack variables

$x_j \leq u_j \Rightarrow x_j + s_j = u_j, \ s_j \geq 0$

\[ \cdots + x_j \cdots = b_p \]
\[ \cdots - x_j \cdots = b_q \]
\[ x_j + s_j = u_j \]
Remove arc capacities

Idea: slack variables

\[ x_j \leq u_j \quad \Rightarrow \quad x_j + s_j = u_j, \quad s_j \geq 0 \]

Network structure lost
no longer one \(-1\)
and one \(1\) per column

\[ \cdots + x_j \cdots = b_p \]
\[ \cdots - x_j \cdots = b_q \]

\[ x_j + s_j = u_j \]
Remove arc capacities

**Idea:** slack variables

\[ x_j \leq u_j \Rightarrow x_j + s_j = u_j, \quad s_j \geq 0 \]

Network structure lost
no longer one \(-1\)
and one \(1\) per column

\[ \cdots + x_j \cdots = b_p \]
\[ \cdots - x_j \cdots = b_q \]
\[ x_j + s_j = u_j \]

Network structure recovered
(new node and new arc)

\[ \cdots - s_j = b_p - u_j \]
\[ \cdots - x_j \cdots = b_q \]
\[ x_j + s_j = u_j \]
Equivalent uncapacitated network flow

minimize \[ c^T x \]
subject to \[ Ax = b \]
\[ x \geq 0 \]

- \( A \) still an arc-node incidence matrix
- Can we say something about the extreme points?
Total unimodularity

A matrix is **totally unimodular** if all its minors are $-1$, $0$ or $1$ (minor is the determinant of a square submatrix of $A$)
Total unimodularity

A matrix is **totally unimodular** if all its minors are $-1$, $0$ or $1$ (minor is the determinant of a square submatrix of $A$)

**example:** a node-arc incidence matrix of a directed graph

\[
A = \begin{bmatrix}
1 & 0 & -1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & -1 & -1 & 0 \\
-1 & -1 & 0 & 1 & 0 & -1
\end{bmatrix}
\]
**Total unimodularity**

A matrix is **totally unimodular** if all its minors are $-1$, $0$, or $1$ (minor is the determinant of a square submatrix of $A$)

**example:** a node-arc incidence matrix of a directed graph

\[
A = \begin{bmatrix}
1 & 0 & -1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & -1 & -1 & 0 \\
-1 & -1 & 0 & 1 & 0 & -1
\end{bmatrix}
\]

**properties**
- the entries of $A_{ij}$ (i.e., its minors of order 1) are $-1$, $0$, or $1$
- The inverse of any nonsingular square submatrix of $A$ has entries $+1$, $-1$, or $0$
Integrality theorem

Given a polyhedron

\[ P = \{ x \in \mathbb{R}^n \mid Ax = b, \ x \geq 0 \} \]

where

- \( A \) is totally unimodular
- \( b \) is an integer vector

all the extreme points of \( P \) are integer vectors.
Integrality theorem

Given a polyhedron

\[ P = \{ x \in \mathbb{R}^n \mid Ax = b, \quad x \geq 0 \} \]

where

- \( A \) is totally unimodular
- \( b \) is an integer vector

all the extreme points of \( P \) are integer vectors.

Proof

- All extreme points are basic feasible solutions
  with \( x_B = A_B^{-1} b \) and \( x_i = 0, \ i \neq B \)
- \( A_B^{-1} \) has integer components because of total unimodularity of \( A \)
- \( b \) has also integer components
- Therefore, also \( x \) is integral
Implications for network and combinatorial optimization

Minimum cost network flow

minimize \( c^T x \)
subject to \( Ax = b \)
\( 0 \leq x \leq u \)

If \( b \) and \( u \) are integral solutions \( x^* \) are integral
Implications for network and combinatorial optimization

Minimum cost network flow

minimize \( c^T x \)
subject to \( Ax = b \)
\( 0 \leq x \leq u \)

If \( b \) and \( u \) are integral solutions \( x^* \) are integral

Integer linear programs

minimize \( c^T x \)
subject to \( Ax = b \)
\( 0 \leq x \leq u \)
\( x \in \mathbb{Z}^n \)

Very difficult in general
(more on this in a few weeks)
Implications for network and combinatorial optimization

Minimum cost network flow

minimize \( c^T x \)
subject to \( Ax = b \)
\( 0 \leq x \leq u \)

If \( b \) and \( u \) are integral
solutions \( x^* \) are integral

Integer linear programs

minimize \( c^T x \)
subject to \( Ax = b \)
\( 0 \leq x \leq u \)
\( x \in \mathbb{Z}^n \)

Very difficult in general
(more on this in a few weeks)

If \( A \) totally unimodular
and \( b, u \) integral, we can
relax integrality and solve
a fast LP instead
Examples
Maximum flow problem

**Goal** maximize flow from node 1 (source) to node $m$ (sink) through the network

maximize \[ t \]
subject to \[ Ax = te \]
\[ 0 \leq x \leq u \]
\[ e = (1, 0, \ldots, 0, -1) \]
Maximum flow as minimum cost flow

minimize \(-t\)

subject to \[
\begin{bmatrix}
A & -e
\end{bmatrix}
\begin{bmatrix}
x \\
t
\end{bmatrix}
= 0
\]

\[
0 \leq \begin{bmatrix}
x \\
t
\end{bmatrix}
\leq \begin{bmatrix}
u \\
\infty
\end{bmatrix}
\]
Maximum flow example

(arc capacities shown)
Maximum flow example

First flow

(ARC capacities shown)
Maximum flow example

First flow

Second flow

(arc capacities shown)
Maximum flow example

First flow

Second flow

Third flow

(arc capacities shown)
Maximum flow example

First flow

Second flow

Third flow

Total flow: 19
Shortest path problem

**Goal** Find the shortest path between nodes 1 and \( m \)

Paths can be represented as vectors \( x \in \{0, 1\}^n \)
Shortest path problem

Goal Find the shortest path between nodes 1 and $m$

Formulation

minimize $c^T x$
subject to $Ax = e$
$x \in \{0, 1\}^n$

paths can be represented as vectors $x \in \{0, 1\}^n$

- $c_j$ is the “length” of arc $j$
- $e = (1, 0, \ldots, 0, -1)$
- Variables are binary (include or not arc in path)
Shortest path as minimum cost flow

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = e \\
& \quad x \in \{0, 1\}^n
\end{align*}
\]
Shortest path as minimum cost flow

minimize \( c^T x \)
subject to \( Ax = e \)
\( x \in \{0, 1\}^n \)

Relaxation

minimize \( c^T x \)
subject to \( Ax = e \)
\( 0 \leq x \leq 1 \)
Shortest path as minimum cost flow

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = e \\
x & \in \{0, 1\}^n
\end{align*}
\]

Relaxation

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = e \\
0 & \leq x \leq 1
\end{align*}
\]

Extreme points satisfy \( x_i \in \{0, 1\} \)
Shortest path as minimum cost flow

minimize \( c^T x \)
subject to \( Ax = e \)
\( x \in \{0, 1\}^n \)

Relaxation

minimize \( c^T x \)
subject to \( Ax = e \)
\( 0 \leq x \leq 1 \)

Example (arc costs shown)

Extreme points satisfy \( x_i \in \{0, 1\} \)
Shortest path as minimum cost flow

minimize \( c^T x \)
subject to \( Ax = e \)
\( x \in \{0, 1\}^n \)

Relaxation

minimize \( c^T x \)
subject to \( Ax = e \)
\( 0 \leq x \leq 1 \)

Example (arc costs shown)

\[ c = (11, 8, 10, 12, 4, 11, 7, 15, 4) \]
\[ x^* = (0, 1, 0, 0, 0, 1, 0, 0, 1) \]
\[ c^T x^* = 24 \]
Assignment problem

Goal match $N$ persons to $N$ tasks

- Each person assigned to one task, each task to one person
- $C_{ij}$ Cost of matching person $i$ to task $j$
Assignment problem

**Goal** match $N$ persons to $N$ tasks
- Each person assigned to one task, each task to one person
- $C_{ij}$ Cost of matching person $i$ to task $j$

**LP formulation**

minimize $\sum_{i,j=1}^{N} C_{ij} X_{ij}$

subject to $\sum_{i=1}^{N} X_{ij} = 1, \quad j = 1, \ldots, N$

$\sum_{i=1}^{N} X_{ij} = 1, \quad i = 1, \ldots, N$

$X_{ij} \in \{0, 1\}$
Assignment problem

Goal match $N$ persons to $N$ tasks
- Each person assigned to one task, each task to one person
- $C_{ij}$ Cost of matching person $i$ to task $j$

LP formulation

minimize $\sum_{i,j=1}^{N} C_{ij} X_{ij}$

subject to $\sum_{i=1}^{N} X_{ij} = 1, \quad j = 1, \ldots, N$

$\sum_{i=1}^{N} X_{ij} = 1, \quad i = 1, \ldots, N$

$X_{ij} \in \{0, 1\}$

How do you define the network?
Task assignment as minimum cost network flow

(arc costs shown)
Task assignment as minimum cost network flow

Person | Task
---|---
1 | 5 6
1 | 8 1
1 | 4 3 9

(arc costs shown)

\[
c = (5, 6, 2, 8, 1, 3, 4, 3, 9)
\]

\[
A = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
-1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \\
\end{bmatrix}
\]

\[
b = (1, 1, 1, -1, -1, -1)
\]
Task assignment as minimum cost network flow

Person  Task

\[
\begin{array}{lllllllll}
1 & 5 & 6 & -1 \\
1 & 8 & 1 & -1 \\
1 & 4 & 3 & 9 & -1 \\
\end{array}
\]

\[
\begin{array}{lllllllll}
1 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
-1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \\
\end{array}
\]

\[
b = (1, 1, 1, -1, -1, -1)
\]

\[
c = (5, 6, 2, 8, 1, 3, 4, 3, 9)
\]

\[
A = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
-1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \\
\end{bmatrix}
\]

Minimum cost network flow

minimize \( c^T x \)

subject to \( Ax = b \)

\( 0 \leq x \leq 1 \)
Task assignment as minimum cost network flow

\[ c = (5, 6, 2, 8, 1, 3, 4, 3, 9) \]

\[ A = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
-1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 
\end{bmatrix} \]

\[ b = (1, 1, 1, -1, -1, -1) \]

**Minimum cost network flow**

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad 0 \leq x \leq 1
\end{align*}
\]

Extreme points satisfy \( x_i \in \{0, 1\} \)
Task assignment as minimum cost network flow

\[ c = (5, 6, 2, 8, 1, 3, 4, 3, 9) \]

\[ A = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
-1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1
\end{bmatrix} \]

\[ b = (1, 1, 1, -1, -1, -1) \]

Minimum cost network flow

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad 0 \leq x \leq 1
\end{align*}
\]

Optimal solution

\[ x^* = (0, 0, 1, 0, 1, 0, 0, 0, 1) \]

\[ c^T x^* = 7 \]
Network optimization

Today, we learned to:

• Model flows across networks

• Formulate minimum cost network flow problems

• Analyze network flow problem solutions (integrality theorem)

• Formulate maximum-flow, shortest path, and assignment problems as minimum cost network flows
References

• D. Bertsimas and J. Tsitsiklis: Introduction to Linear Optimization
  • Chapter 7: Network flow problems

• R. Vanderbei: Linear Programming
  • Chapter 14: Network Flow Problems
  • Chapter 15: Applications
Next lecture

- Interior point algorithms