

LECTURE 2: MATHEMATICAL PRELIMINARIES

1. Sets and Relations
2. Graph Theory
3. Linear Programming

Notations

- Sets and Relations

- ◇ Basic set operations and notations:

- $\in, \notin, \sim, \cup, \cap, \subseteq, \subset, \emptyset$, etc

- ◇ $S \subset T$ means S is a *proper subset* of T ,
i.e., $S \subseteq T$ but $S \neq T$.

- ◇ $S + e = S \cup \{e\}$

- and

- $S - e = S - \{e\}$

- ◇ *Symmetric Difference*:

- $S \oplus T = (S \cup T) - (S \cap T)$ = the set of all elements
contained in S or in T , but not both.

- ◇ $|S|$ = the number of elements in S (*cardinality* of S .)

- ◇ $\mathcal{P}(S)$ = the *power set* of S .

Sets and Relations

- ◇ Suppose that $\mathcal{F}$ is a family of sets.
 - $S \in \mathcal{F}$ is *minimal* in $\mathcal{F}$ if there is no $T \in \mathcal{F}$ such that $T \subset S$.
 - $S \in \mathcal{F}$ is *maximal* in $\mathcal{F}$ if there is no $T \in \mathcal{F}$ such that $S \subset T$.
- ◇ Same concepts of minimality and maximality go for ordered sets.
- ◇ If S is a finite set of numbers, $\min S$ ($\max S$) denotes the numerically smallest (largest) element in S .
- ◇ Alternative notations for $\min A$, where $A = \{a_1, a_2, \dots, a_n\}$ are:
 $\min\{a_i | 1 \leq i \leq n\}$ or $\min_{1 \leq i \leq n} \{a_i\}$ or $\min_i \{a_i\}$
- ◇ A is a matrix whose typical element is a_{ij} , written $A = (a_{ij})$. Then
 - $\min_j a_{ij}$ = the smallest element in row i
 - $\max_i a_{ij}$ = the largest element in column j .

Sets and Relations

- ◇ Suppose $\leq$ is a total ordering of A , i.e., a partial ordering such that for each pair of elements a, b , in A either $a \leq b$ or $b \leq a$. Then this total ordering induces a *lexicographic ordering* “ $\preceq$ ” of A^n , the set of all n -tuples of elements of A .

Let $a = (a_1, a_2, \dots, a_n)$ and $b = (b_1, b_2, \dots, b_n)$. Then $a \preceq b$ if either $a = b$ or there is some k , $1 \leq k \leq n$, such that $a_i = b_i$, $i = 1, 2, \dots, k - 1$, and $a_k < b_k$.

Suppose $\mathcal{A} = A \cup A^2 \cup A^3 \cup \dots$. We can define a lexicographic ordering on $\mathcal{A}$ as follows. Let $a = (a_1, a_2, \dots, a_m)$ and $b = (b_1, b_2, \dots, b_n)$, where $m \leq n$. Then $a \preceq b$ if $a \preceq (b_1, b_2, \dots, b_m)$, as defined above, and $b \preceq a$ otherwise.

Graphs and Digraphs

- ◇ A *graph* $G = (N, A)$ is a structure consisting of a finite set N of elements called *nodes* and a set A of unordered pairs of nodes called *arcs*. A *directed graph* or *digraph* is defined similarly, except that each arc is an ordered pair, giving it direction from one node to another.
- ◇ For both undirected and directed graphs, an arc from node i to node j is denoted by (i, j) . An arc (i, i) is called a *loop*. Ordinarily we deal with undirected graphs with no loops and at most one arc between a given pair of nodes i, j .

Graphs and Digraphs

- ◇ If there exists an arc (i, j) we say that nodes i and j are *adjacent*.
For an undirected graph, the *adjacency matrix* $A = (a_{ij})$ is defined as follows:

$$\begin{aligned} a_{ij} &= 1 && \text{if there is an arc } (i, j) \text{ between nodes } i \text{ and } j \\ &= 0 && \text{otherwise.} \end{aligned}$$

- ◇ In the case of a directed graph, if there is an arc (i, j) we say that node i is *adjacent to* node j and node j is *adjacent from* node i .
The *adjacency matrix* $A = (a_{ij})$ is defined as follows:

$$\begin{aligned} a_{ij} &= 1 && \text{if there is an arc } (i, j) \text{ from } i \text{ to } j \\ &= 0 && \text{otherwise.} \end{aligned}$$

Graphs and Digraphs

- ◇ An arc (i, j) is said to be *incident* to each of the nodes i and j , and conversely. Each row of the node-arc incidence matrix is identified with a node and each column with an arc. If the arcs are numbered by the index k , then the *incidence matrix* $B = (b_{ik})$ is defined as follows:

$$\begin{aligned} b_{ik} &= 1 && \text{if node } i \text{ is incident to arc } k \\ &= 0 && \text{otherwise} \end{aligned}$$

- ◇ In the case of a directed graph the arc (i, j) , directed from i to j , is said to be *incident from* i and *incident to* j . The arc-node *incidence matrix* $B = (b_{ik})$ is defined as follows:

$$\begin{aligned} b_{ik} &= +1 && \text{if arc } k \text{ is incident to node } i \\ &= -1 && \text{if arc } k \text{ is incident from node } i \\ &= 0 && \text{otherwise.} \end{aligned}$$

Graphs and Diagraphs

- ◇ Of special interest is the *bipartite* graph. The nodes of a bipartite graph can be partitioned into two sets S and T , such that no two nodes in S or in T are adjacent. If a graph $G = (N, A)$ is bipartite, we commonly denote it as $G = (S, T, A)$ where $N = S \cup T$.
- ◇ **Proposition**
 G is a bipartite graph if and only if its nodes can be numbered in such a way that its adjacency matrix takes on the form:

$$A = \left(\begin{array}{c|c} 0 & \overline{A} \\ \hline \overline{A}^T & 0 \end{array} \right)$$

Graphs and Digraphs

- ◇ The *degree* d_i of node i is the number of arcs incident to the node. If B is the incidence matrix,

$$d_i = \sum_k b_{ik}.$$

- ◇ In the case of a digraph, the *out-degree* $d_i^{(\text{out})}$ of node i is the number of arcs incident from the node, and the *in-degree* $d_i^{(\text{in})}$ is the number of arcs incident to the node. Note that

$$d_i^{(\text{in})} - d_i^{(\text{out})} = \sum_k b_{ik}$$

Subgraphs, Cliques and Multigraphs

- ◇ The *complete graph* K_n has n nodes any two of which are adjacent. The complete graph has $n(n-1)/2$ arcs. The *complete digraph* on n nodes has $n(n-1)$ arcs. The *complete bipartite graph* $K_{p,q}$ is a bipartite graph $G = (S, T, A)$, with $|S| = p$, $|T| = q$, and $|A| = pq$.
- ◇ A graph $G = (N', A')$ is called a *subgraph* of the graph $G = (N, A)$ if $N' \subseteq N$ and $A' \subseteq A$. If $N' \subseteq N$, then the *subgraph of G induced by N'* has the node set N' and all arcs (i, j) in A such that both i and j are in N' . If a subgraph of G is a complete graph it is a *complete subgraph*. A maximal complete subgraph is called a *clique*.

Subgraphs, Cliques and Multigraphs

- ◇ The *complement* of the graph $G = (N, A)$ is the graph $\overline{G}$ obtained by deleting the arcs of G from the complete graph on the same nodes.
- ◇ The *contraction* of an arc (i, j) is accomplished by replacing nodes i and j by a single node k . An arc (k, l) is provided in the contracted graph for each arc (i, l) or (j, l) in the original graph, except arc (i, j) . The contraction of a graph may well result in a graph with multiple arcs between nodes. Such a structure we call a *multigraph*.

Connectivity in Graphs

- ◇ A *path* between s and t , or simply an (s, t) *path*, is a sequence of arcs of the form $(s, i_1), (i_1, i_2), \dots, (i_k, t)$. If $s, i_1, i_2, \dots, i_k, t$ are distinct nodes, we say that the path is *minimal* or *without repeated nodes*.
- ◇ An (s, t) path is *open* if $s \neq t$ and *closed* if $s = t$. A *cycle* is an (s, s) path containing at least one arc, in which no node except s is repeated. In an ordinary graph, a cycle must contain at least three arcs. A graph which contains no cycles is *acyclic*.
- ◇ Two nodes i and j are said to be *connected* if there exists an (i, j) path. A graph G is said to be *connected* if all pairs of nodes are connected. A *component* of a graph G is a maximal connected subgraph. A graph is connected if and only if it has exactly one component.

Connectivity in Graphs

◇ Proposition

If a graph G has p components, then its nodes can be numbered in such a way that its adjacency matrix takes on the block diagonal form

$$A = \begin{pmatrix} \boxed{A_1} & 0 & 0 & 0 \\ 0 & \boxed{A_2} & 0 & 0 \\ 0 & 0 & \boxed{} & 0 \\ 0 & 0 & 0 & \boxed{A_p} \end{pmatrix}$$

◇ A *tree* is a connected acyclic graph.

Connectivity in Graphs

◇ Proposition

The following statements are equivalent for a graph G with n nodes:

- (1) G is a tree.
- (2) Every two nodes of G are connected by a unique path.
- (3) G is connected and has $n - 1$ arcs.
- (4) G is acyclic and has $n - 1$ arcs.
- (5) G is acyclic and if any two nonadjacent nodes of G are joined by an arc e , then $G + e$ has exactly one cycle.
- (6) G is connected, is not K_n for $n \geq 3$, and if any two nonadjacent nodes of G are joined by a new arc e , then $G + e$ has exactly one cycle.

Connectivity in Graphs

- ◇ A *tree* in G is a connected acyclic subgraph on the nodes of G . A *forest* in G is an acyclic subgraph on the nodes of G . A maximal forest in a connected graph is a *spanning tree*.
- ◇ **Theorem**
Every maximal forest in a graph with n nodes and p components contains $n - p$ arcs.
- ◇ **Theorem**
(C. W. Borchardt) K_n contains n^{n-2} distinct spanning trees.

Connectivity in Digraphs

- ◇ A *directed path* from s to t , or simply an (s, t) *path*, is a sequence of arcs from s to t , where the p th arc is incident to the same node from which the $(p + 1)$ st arc is incident. A *directed cycle* is a minimal nonempty closed directed path.
- ◇ A node i is said to be *connected* to node j , and j is said to be *connected from* i if there exists an (i, j) path. A digraph G is said to be *strongly connected* if, for all pairs of nodes i and j , i is connected to j and j is connected to i . A *strong component* of a graph G is a strongly connected subgraph of G which is maximal.

Connectivity in Digraphs

◇ Proposition

If a directed graph G has p strong components, then its nodes can be numbered in such a way that its adjacency matrix takes on the form

$$A = \begin{pmatrix} \boxed{A_1} & & & \\ 0 & \boxed{A_2} & & \\ 0 & 0 & \boxed{} & \\ 0 & 0 & 0 & \boxed{A_p} \end{pmatrix}$$

where the entries above the block diagonal submatrices are 0's and 1's.

Connectivity in Digraphs

- ◇ A *directed tree* is either rooted to a node or from a node. A *tree rooted from node i* is a tree in which the in-degree of i is zero, and the in-degree of each of the other nodes is at most one. A *tree rooted to node i* is a tree in which the out-degree of i is zero and the out-degree of the other nodes is at most one. A *directed spanning tree* is just as its name suggests.
- ◇ A directed graph is called *acyclic* if it contains no directed cycles.

Connectivity in Digraphs

◇ Proposition

A directed graph is acyclic if and only if its nodes can be numbered in such a way that for all arcs (i, j) , $i < j$.

◇ RENUMBERING THE NODES OF AN ACYCLIC DIGRAPH

Step 0 (Start)

Set $d_j^{(\text{in})} = \sum_{i=1}^n a_{ij}$, $j = 1, 2, \dots, n$,

Set $N = \{1, 2, \dots, n\}$.

Set $m = 1$.

Step 1 (Detection of Node with Zero In-Degree)

Find $k \in N$ such that $d_k^{(\text{in})} = 0$. If there is no such k , stop; the digraph is not acyclic.

Set $v(k) = m$.

Set $m = m + 1$.

Set $N = N - k$.

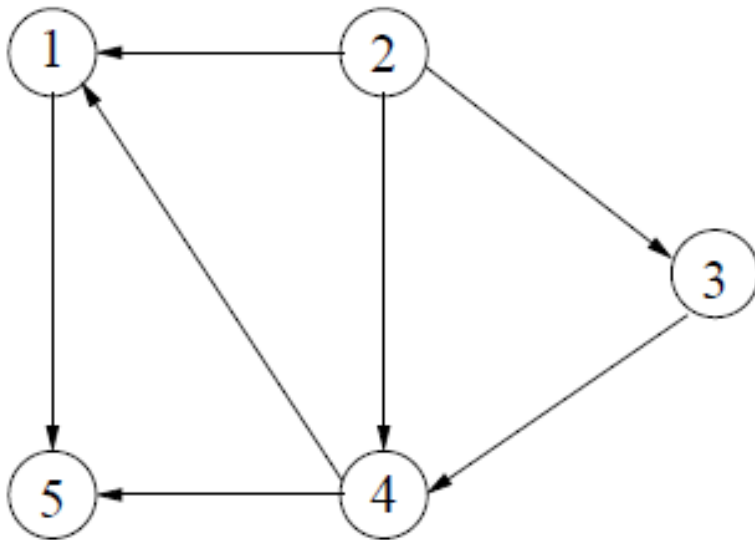
If $N = \emptyset$, stop; the computation is completed.

Step 2 (Revision of In-Degrees)

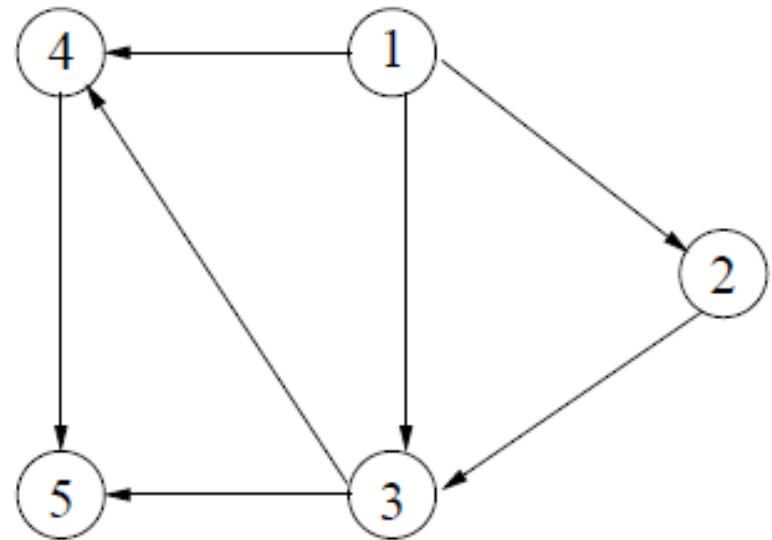
Set $d_j^{(\text{in})} = d_j^{(\text{in})} - a_{kj}$, for all $j \in N$.

Return to Step 1.

Example



Acyclic ?



Proposition 6.2 ?

Questions

1. Is Proposition 6.2 valid?
2. How to re-number nodes?
3. Does the re-numbering algorithm work?
 - (a) Is each step valid?
 - (b) Will it stop?
 - (c) When it stops, does it provide a solution?
4. How efficient is the algorithm?
5. How to identify an acyclic digraph from the adjacency matrix?

Part of Answers

a). Computation work at each step

Step 0: $n(n-1)$ additions plus minors
 $\approx O(n^2)$

Step 1: at most n comparisons to find k plus minors
 $\approx O(n)$

Step 2: at most $(n-1)$ subtractions
 $\approx O(n)$

b). How many iterations ?

Step 0: once only $\approx O(n^2)$

Steps 1 & 2: at most n times $\approx O(n^2)$

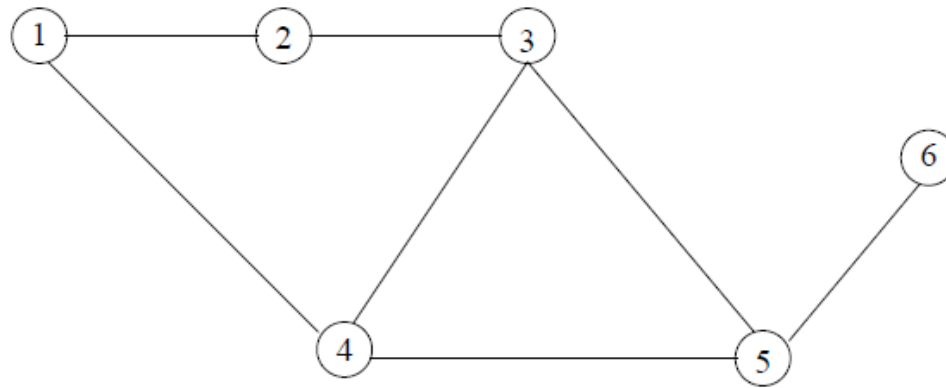
Total complexity $O(n^2)$.

5. Adjacency matrix: upper-triangular.

Cocycles and Directed Cocycles

- ◇ Let $G = (N, A)$ be a graph, or a directed graph in which the directions of the arcs are ignored. A subset $C \subseteq A$, such that $G' = (N, A - C)$ contains more components than G , is a *separating set* of G . A minimal separating set is a *cocycle* of G .
- ◇ Given an arbitrary node partition S, T , the set of arcs extending between S and T is not necessarily a cocycle. We call a separating set determined by such a partition a *cutset* and we may refer to it by any one of the node partitions S, T which determines it. An (s, t) – *cutset* is any cutset (S, T) , where $s \in S$ and $t \in T$.

Example



Separating sets:

$$C_1 = \{(1, 2), (1, 4)\}$$

$$C_2 = \{(1, 2), (2, 3), (1, 4)\}$$

$$C_3 = \{(2, 3), (1, 4)\}$$

$$C_4 = \{(1, 2), (3, 4), (4, 5), (5, 6)\}$$

$$C_5 = \{(1, 2), (3, 4), (4, 5)\}$$

$$C_6 = \{(5, 6)\}$$

Example

<u>Cocycles:</u>	<u>Cutsets</u>
C_1 ?	$S = \{1, 4, 6\}$
C_3 ?	$T = \{2, 3, 5\}$
C_5 ?	C_4 is an (S, T) cutset
C_6 ?	but $C_4 = C_5 \cup C_6$
$\vdots$	

Cocycles and Directed Cocycles

◇ Proposition

Every cutset is a union of disjoint cocycles.

◇ Theorem (*Minty*)

Let G be a directed graph with a distinguished arc (s, t) . Then, for any painting of the arcs green, yellow, and red, with (s, t) painted yellow, exactly one of the following alternatives holds:

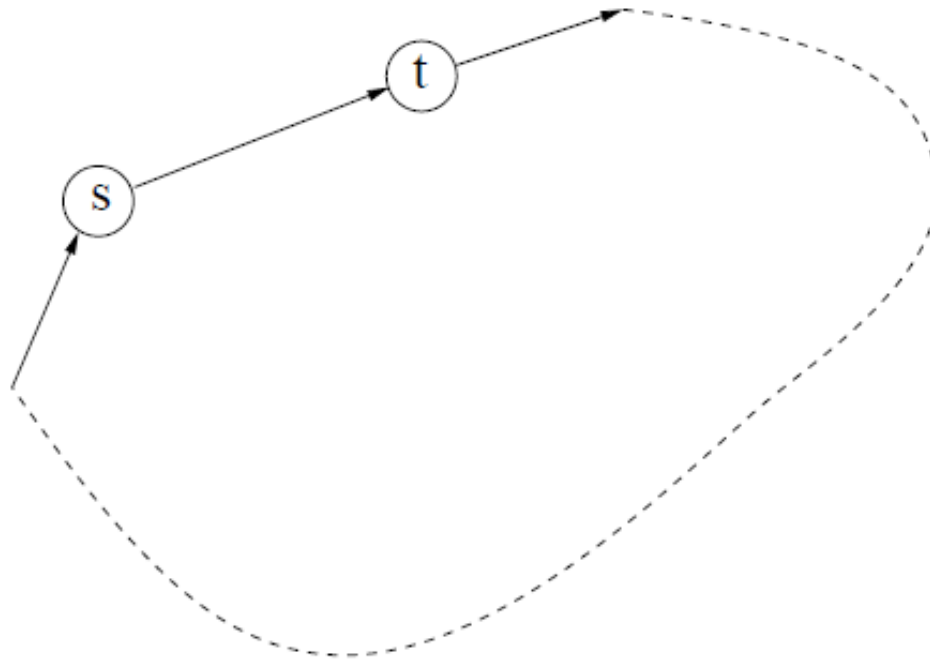
- (1) (s, t) is contained in a cycle of yellow and green arcs, in which all yellow arcs have the same direction.
- (2) (s, t) is contained in a cocycle of yellow and red arcs, in which all yellow arcs have the same direction.

Proof of Minty's Theorem

Green: two-way street

Yellow: one-way street

Red: blocked street



Proof of Minty's Theorem

Case 1: There is a directed path from t to s . So we have a cycle and *condition 1 is met*.

Case 2: No directed path from t to s . Let
 $T = \{ \text{all nodes accessible from } t \},$
 $S = N - T, (s \in S ?)$

So we have a cutset of reds and yellows in the same direction as (s, t) . By Proposition 7.1, there is such a cocycle and *condition 2 is met*.

Eulerian and Hamiltonian Graphs

- ◇ The general question, for a given graph G , is whether there exists a closed path which contains each arc exactly once. Such a path, if it exists, we call an *Euler path*, and we say the graph is an *Euler graph*, or *Eulerian*.
- ◇ **Theorem**
A graph (or multigraph) G is Eulerian if and only if G is connected and each node of G has even degree.

Eulerian and Hamiltonian Graphs

◇ We call a cycle that passes through each node of a graph exactly once a *Hamilton cycle*, and the graph which contains it a *Hamilton graph*, or *Hamiltonian*.

◇ **Theorem** (*Chvátal*)

Let G be a graph with $n \geq 3$ nodes and no loops or multiple arcs in which the nodes are numbered so that $d_1 \leq d_2 \leq \dots \leq d_n$. G is Hamiltonian if

$$d_k \leq k \Rightarrow d_{n-k} \geq n - k, \quad \text{for } 1 \leq k \leq \frac{n}{2}.$$

Intersection Graphs

- ◇ Let S be a set and $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ be a family of distinct nonempty subsets of S whose union is S . The *intersection graph* of $\mathcal{S}$ is a graph whose nodes are identified with sets in $\mathcal{S}$, with S_i and S_j adjacent whenever $i \neq j$ and $S_i \cap S_j \neq \emptyset$. A graph G is an *intersection graph on S* if there exists a family $\mathcal{S}$ of subsets of S , with G the intersection graph of $\mathcal{S}$.
- ◇ **Theorem**
Every graph $G = (N, A)$ is an intersection graph.

Line Graphs

- ◇ For a given graph $G = (N, A)$, we can let $S = N$ and $\mathcal{S} = A$. The intersection graph of A is called the *line graph of G* , denoted $L(G)$. A graph G' is called a *line graph* if there exists a graph G , with $G' = L(G)$. Sometimes $L(G)$ is called the “arc-to-node dual” of G .
- ◇ **Theorem**
 G is a line graph if and only if the arcs of G can be partitioned into complete subgraphs in such a way that no node lies in more than two of the subgraphs.

Eulerian and Hamiltonian Graphs

- ◇ **Theorem**

G is Eulerian if and only if $L(G)$ is Hamiltonian.

- ◇ **Theorem**

If G is Eulerian, then $L(G)$ is Eulerian.

Linear Programming

minimize

$$z = c^T x$$

subject to

$$Ax = b,$$

$$x \geq 0,$$

- ◇ Any m linearly independent columns of A will be referred to as a *basis* of the linear system $Ax = b$.

Linear Programming

◇ If, for a basis B , we suppress the $n - m$ secondary variables, the linear system $Bx^B = b$ is obtained, and this system possesses a unique solution $x^B = B^{-1}b$. The *basic solution associated with B* is defined as $x^B = B^{-1}b, x^N = 0$, but often we refer to the basic solution as simply x^B . A basic solution x^B which is feasible (i.e., $x^B \geq 0$) we call a *basic feasible solution* and a basic solution which is optimal we call a *basic optimal solution*.

◇ **Theorem**

If there exists a feasible solution to the LP , there exists a basic feasible solution.

◇ **Theorem**

If there exists an optimal solution to the LP , there exists a basic optimal solution.

Linear Programming

◇ Proposition

A system of linear inequalities determines a convex polyhedron with integer vertices if and only if, for all possible choices of an objective function, there exists a finite optimal solution in integers.

Duality Theory

Primal Problem	Dual Problem
Minimize $z = \sum_{j=1}^n c_j x_j$ Subject to $\sum_{j=1}^n a_{ij} x_j \geq b_i$ $\sum_{j=1}^n a_{ij} x_j = b_i$ $x_j \geq 0$ x_j unrestricted	Minimize $w = \sum_{i=1}^m (-b_i) u_i$ Subject to $u_i \geq 0$ u_i unrestricted $\sum_{i=1}^m (-a_{ij}) u_i \geq -c_j$ $\sum_{i=1}^m (-a_{ij}) u_i = -c_j$

Linear Duals

◇ The following pairs of problems are duals:

minimize $c^T x$	maximize $u^T b$
subject to	subject to
$Ax \geq b$	$uA \leq c$
$x \geq 0$	$u \geq 0$

minimize $c^T x$	maximize $u^T b$
subject to	subject to
$Ax = b$	$uA \leq c$
$x \geq 0$	u unrestricted

minimize $c^T x$	maximize $u^T b$
subject to	subject to
$Ax = b$	$uA = c$
x unrestricted	u unrestricted

Duality Theory

- ◇ **Theorem** (*Weak Duality*)

If $\bar{x}$ and $\bar{u}$ are feasible solutions to dual problems, then $c^T \bar{x} \geq \bar{u}^T b$.

- ◇ **Corollary**

If $\bar{x}$ and $\bar{u}$ are feasible solutions to dual problems and $c^T \bar{x} = \bar{u}^T b$ then $\bar{x}$ and $\bar{u}$ are optimal solutions.

- ◇ **Theorem** (*Strong Duality*)

If either problem of a dual pair of problems has a finite optimum, then the other does also and the two optimal objective values are equal; if either has an unbounded optimum, the other has no feasible solution.

Duality Theory

◇ **Theorem** (*Orthogonality of Optimal Solutions*)

If $\bar{x}$ and $\bar{u}$ are feasible solutions to

minimize $c^T x$	maximize $u^T b$
subject to	subject to
$Ax \geq b$	$uA \leq c$
$x \geq 0$	$u \geq 0$

then $\bar{x}$ and $\bar{u}$ are optimal if and only if

$(\bar{u}A - c)^T \bar{x} = \bar{u}^T (A\bar{x} - b) = 0$. That is, if and only if, for $j = 1, 2, \dots, n$,

$$\bar{x}_j > 0 \quad \text{implies} \quad \sum_{i=1}^m \bar{u}_i a_{ij} = c_j$$

and, for $i = 1, 2, \dots, m$,

$$\bar{u}_i > 0 \quad \text{implies} \quad \sum_{j=1}^n a_{ij} \bar{x}_j = b_i.$$