DM545/DM871
Linear and Integer Programming

Lecture 11
Network Flows

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

Separation problem

$\max\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in X\} \equiv \max\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in \text{conv}(X)\}$
$X \subseteq \mathbb{Z}^n$, P a polyhedron $P \subseteq \mathbb{R}^n$ and $X = P \cap \mathbb{Z}^n$

Definition (Separation problem for a COP)

Given $\mathbf{x}^* \in P$; is $\mathbf{x}^* \in \text{conv}(X)$? If not find an inequality $\mathbf{a}\mathbf{x} \leq \mathbf{b}$ satisfied by all points in X but violated by the point $\mathbf{x}^*$.

(Farkas' lemma states the existence of such an inequality.)

Properties of Easy Problems

Four properties that often go together:

Definition

(i) Efficient optimization property: $\exists$ a polynomial algorithm for $\max\{\mathbf{c}\mathbf{x} : \mathbf{x} \in X \subseteq \mathbb{R}^n\}$

(ii) Strong duality property: $\exists$ strong dual D $\min\{w(\mathbf{u}) : \mathbf{u} \in U\}$ that allows to quickly verify optimality

(iii) Efficient separation problem: $\exists$ efficient algorithm for separation problem

(iv) Efficient convex hull property: a compact description of the convex hull is available

Example:
  If explicit convex hull   strong duality holds
                            efficient separation property (just description of
                            $\mathrm{conv}(X)$)

Theoretical analysis to prove results about

- strength of certain inequalities that are facet defining
  2 ways

- descriptions of convex hull of some discrete $X \subseteq \mathbb{Z}^*$
  several ways, we see one next

Example

Let

$$X = \{(x, y) \in \mathbb{R}_+^m \times \mathbb{B}^1 : \sum_{i=1}^{m} x_i \leq my, x_i \leq 1 \text{ for } i = 1, \ldots, m\}$$

$$P = \{(x, y) \in \mathbb{R}_+^n \times \mathbb{R}^1 : x_i \leq y \text{ for } i = 1, \ldots, m, y \leq 1\}$$

.

Polyhedron P describes conv(X)

Totally Unimodular Matrices

When the LP solution to this problem

$$IP : \max\{\mathbf{c}^T\mathbf{x} : A\mathbf{x} \le \mathbf{b}, \mathbf{x} \in \mathbb{Z}_+^n\}$$

with all data integer will have integer solution?

$$\begin{bmatrix} & & & \\ & A_N & A_B & \mathbf{0} & \mathbf{b} \\ & & & & \\ \hline & \mathbf{c}_N^T & \mathbf{c}_B^T & 1 & 0 \end{bmatrix}$$

$A_B x_B + A_N x_N = b$

$\mathbf{x}_N = \mathbf{0} \rightsquigarrow A_B\mathbf{x}_B = \mathbf{b}$,
A_B $m \times m$ non singular matrix
$\mathbf{x}_B \ge 0$

Cramer's rule for solving systems of linear equations:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix} \qquad x = \frac{\begin{vmatrix} e & b \\ f & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} \qquad y = \frac{\begin{vmatrix} a & e \\ c & f \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} \qquad \mathbf{x} = A_B^{-1}\mathbf{b} = \frac{A_B^{adj}\mathbf{b}}{\det(A_B)}$$

Definition

- A square integer matrix B is called unimodular (UM) if $\det(B) = \pm 1$
- An integer matrix A is called totally unimodular (TUM) if every square, nonsingular submatrix of A is UM

Proposition

- *If A is TUM then all vertices of $R_1(A) = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ are integer if $\mathbf{b}$ is integer*
- *If A is TUM then all vertices of $R_2(A) = \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ are integer if $\mathbf{b}$ is integer.*

Proof: if A is TUM then $\left[A \vdots I \right]$ is TUM

Any square, nonsingular submatrix C of $\left[A \vdots I \right]$ can be written as

$$C = \begin{bmatrix} B & \vdots & 0 \\ D & \vdots & I_k \end{bmatrix}$$

where B is square submatrix of A. Hence $\det(C) = \det(B) = \pm 1$

Proposition

The transpose matrix A^T of a TUM matrix A is also TUM.

Theorem (Sufficient condition)

An integer matrix A is TUM if

1. *$a_{ij} \in \{0, -1, +1\}$ for all i, j*
2. *each column contains at most two non-zero coefficients ($\sum_{i=1}^{m} |a_{ij}| \leq 2$)*
3. *if the rows can be partitioned into two sets I_1, I_2 such that:*
   - *if a column has 2 entries of same sign, their rows are in different sets*
   - *if a column has 2 entries of different signs, their rows are in the same set*

$$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Proof: by induction

Basis: one matrix of one element $\{0, +1, -1\}$ is TUM

Induction: let C be of size k.

If C has column with all 0s then it is singular.

If a column with only one 1 then expand on that by induction

If 2 non-zero in each column then

$$\forall j : \sum_{i \in I_1} a_{ij} = \sum_{i \in I_2} a_{ij}$$

but then a linear combination of rows is zero and $\det(C) = 0$

Other matrices with integrality property:

- TUM
- Balanced matrices
- Perfect matrices
- Integer vertices

Defined in terms of forbidden substructures that represent fractionating possibilities.

Proposition

A is always TUM if it comes from

- *node-edge incidence matrix of undirected bipartite graphs*
  (ie, no odd cycles) ($I_1 = U, I_2 = V, B = (U, V, E)$)
- *node-arc incidence matrix of directed graphs ($I_2 = \emptyset$)*

Eg: Shortest path, max flow, min cost flow, bipartite weighted matching

Summary

1. Well Solved Problems

2. (Minimum Cost) Network Flows

3. Assignment and Transportation

Outline

Terminology

Network:
- directed graph $D = (V, A)$
- arc, directed link, from tail to head
- lower bound $l_{ij} > 0$, $\forall ij \in A$, capacity $u_{ij} \geq l_{ij}$, $\forall ij \in A$
- cost c_{ij}, linear variation (if $ij \notin A$ then $l_{ij} = u_{ij} = 0$, $c_{ij} = 0$)
- balance vector $b(i)$, $b(i) > 0$ supply node (source), $b(i) < 0$ demand node (sink, tank), $b(i) = 0$ transhipment node (assumption $\sum_i b(i) = 0$)
- $N = (V, A, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mathbf{c})$
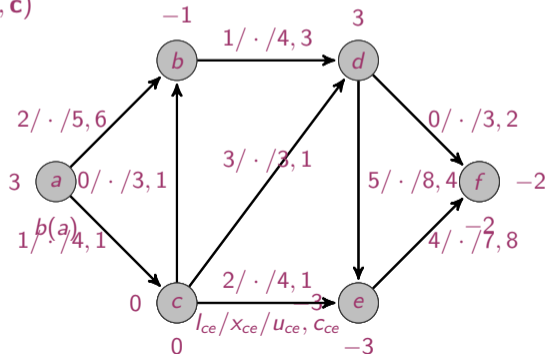
Network Flows

Flow $\mathbf{x} : A \to \mathbb{R}$

    balance vector of $\mathbf{x}$: $b_{\mathbf{x}}(v) = \sum\limits_{vu \in A} x_{vu} - \sum\limits_{wv \in A} x_{wv}, \; \forall v \in V$

$$b_{\mathbf{x}}(v) \begin{cases} > 0 & \text{source} \\ < 0 & \text{sink/target/tank} \\ = 0 & \text{balanced} \end{cases}$$
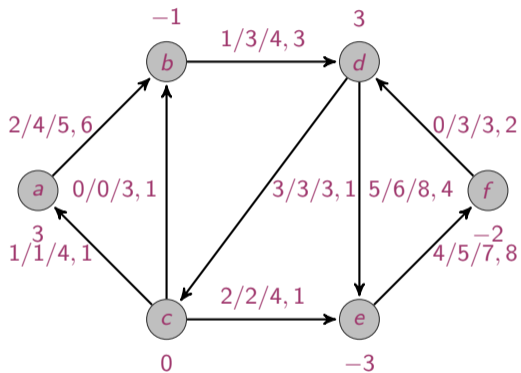
(generalizes the concept of path with $b_{\mathbf{x}}(v) = \{0, 1, -1\}$)

feasible $l_{ij} \leq x_{ij} \leq u_{ij}$, $b_{\mathbf{x}}(i) = b(i)$
cost $\mathbf{c}^T\mathbf{x} = \sum_{ij \in A} c_{ij}x_{ij}$ (varies linearly with $\mathbf{x}$)

If iji is a 2-cycle and all $l_{ij} = 0$, then at least one of x_{ij} and x_{ji} is zero.

Example

Feasible flow of cost 109

Minimum Cost Network Flows

Find cheapest flow through a network in order to satisfy demands at certain nodes from available supplier nodes.

Variables:

$$x_{ij} \in \mathbb{R}_0^+$$

Objective:

$$\min \sum_{ij \in A} c_{ij} x_{ij}$$

$$\min \mathbf{c}^T \mathbf{x}$$
$$N\mathbf{x} = \mathbf{b}$$
$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$$

Constraints: mass balance + flow bounds

$$\sum_{j:ij \in A} x_{ij} - \sum_{j:ji \in A} x_{ji} = b(i) \quad \forall i \in V$$

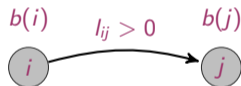$$l_{ij} \leq x_{ij} \leq u_{ij}$$

N node arc incidence matrix

(assumption: all values are integer, we can multiply if rational)

	x_{e_1}	x_{e_2}	$\dots$	x_{ij}	$\dots$	x_{e_m}		
	c_{e_1}	c_{e_2}	$\dots$	c_{ij}	$\dots$	c_{e_m}		
1	-1	.	$\dots$	.	$\dots$	.	$=$	b_1
2	.	.	$\dots$	.	$\dots$	.	$=$	b_2
$\vdots$	$\vdots$	$\ddots$					$=$	$\vdots$
i	1	.	$\dots$	-1	$\dots$	.	$=$	b_i
$\vdots$	$\vdots$	$\ddots$					$=$	$\vdots$
j	.	.	$\dots$	1	$\dots$	.	$=$	b_j
$\vdots$	$\vdots$	$\ddots$					$=$	$\vdots$
n	.	.	$\dots$	.	$\dots$	.	$=$	b_n
e_1	1						$\leq$	u_1
e_2		1					$\leq$	u_2
$\vdots$	$\vdots$	$\ddots$					$\leq$	$\vdots$
(i,j)				1			$\leq$	u_{ij}
$\vdots$	$\vdots$	$\ddots$					$\leq$	$\vdots$
e_m						1	$\leq$	u_m

Reductions/Transformations

Lower bounds

Let $N = (V, A, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mathbf{c})$

$$N' = (V, A, \mathbf{l}', \mathbf{u}', \mathbf{b}', \mathbf{c})$$
$$b'(i) = b(i) - l_{ij}$$
$$b'(j) = b(j) + l_{ij}$$
$$u'_{ij} = u_{ij} - l_{ij}$$
$$l'_{ij} = 0$$

$$\mathbf{c}^T \mathbf{x}$$
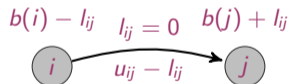
$$\mathbf{c}^T \mathbf{x}' + \sum_{ij \in A} c_{ij} l_{ij}$$

Undirected arcs
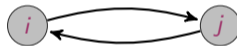
Vertex splitting

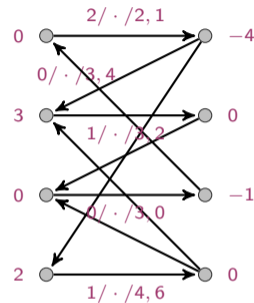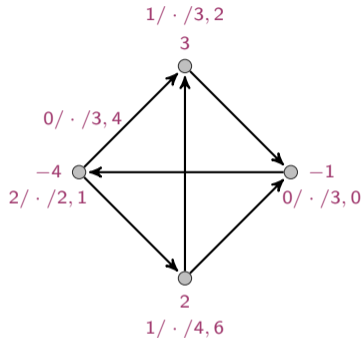If there are bounds and costs of flow passing through vertices where $b(v) = 0$ (used to ensure that a node is visited):

$N = (V, A, \mathbf{l}, \mathbf{u}, \mathbf{c}, \mathbf{l}^*, \mathbf{u}^*, \mathbf{c}^*)$
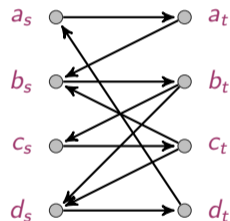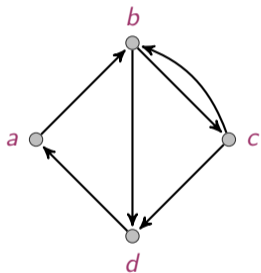
From D to D_{ST} as follows:

$$\forall v \in V \qquad \rightsquigarrow v_s, v_t \in V(D_{ST}) \text{ and } v_s v_t \in A(D_{ST})$$
$$\forall xy \in A(D) \rightsquigarrow x_t y_s \in A(D_{ST})$$

$$\forall v \in V \text{ and } v_s v_t \in A_{ST} \rightsquigarrow h'(v_s v_t) = h^*(v), \ h^* \in \{l^*, u^*, c^*\}$$
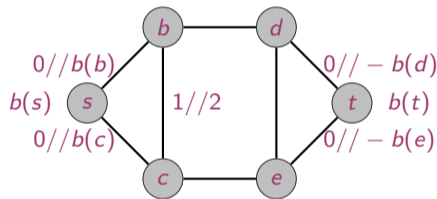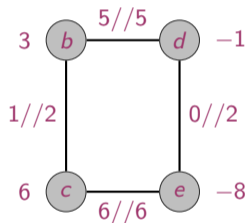$$\forall xy \in A \text{ and } x_t y_s \in A_{ST} \rightsquigarrow h'(x_t y_s) = h(x, y), h \in \{l, u, c\}$$

If $b(v) = 0$, then $b'(v_s) = b'(v_t) = 0$
If $b(v) < 0$, then $b'(v_s) = 0$ and $b'(v_t) = b(v)$
If $b(v) > 0$, then $b'(v_s) = b(v)$ and $b'(v_t) = 0$

(s, t)-**flow:**

$$b_{\mathbf{x}}(v) = \begin{cases} k & \text{if } v = s \\ -k & \text{if } v = t \\ 0 & \text{otherwise} \end{cases}, \qquad |\mathbf{x}| = |b_{\mathbf{x}}(s)|$$

$$b(s) = \sum_{v:b(v)>0} b(v) = M$$
$$b(t) = \sum_{v:b(v)<0} b(v) = -M$$

$\exists$ feasible flow in $N \iff \exists (s, t)$-flow in N_{st} with $|x| = M \iff$ max flow in N_{st} is M
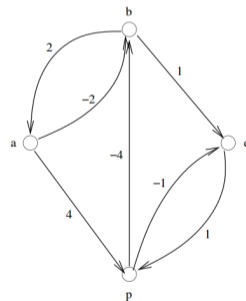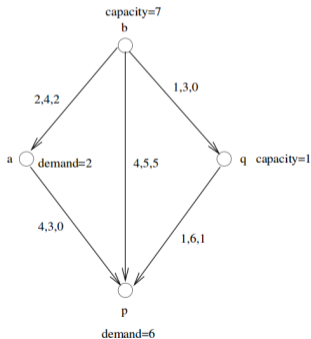
Residual Network

Residual Network $N(\mathbf{x})$: given that a flow $\mathbf{x}$ already exists, how much flow excess can be moved in G?

Replace arc $ij \in N$ with arcs:

	residual capacity	cost
$ij:$	$r_{ij} = u_{ij} - x_{ij}$	c_{ij}
$ji:$	$r_{ji} = x_{ij}$	$-c_{ij}$

$(N, \mathbf{c}, \mathbf{u}, \mathbf{x})$

$(N(\mathbf{x}), \mathbf{c}')$

Special cases

Shortest path problem path of minimum cost from s to t with costs $\lessgtr 0$

$\qquad b(s) = 1, b(t) = -1, b(i) = 0$

$\qquad$ if to any other node? $b(s) = (n-1), b(i) = 1, u_{ij} = n - 1$

Max flow problem incur no cost but restricted by bounds

$\qquad$ steady state flow from s to t

$\qquad b(i) = 0 \; \forall i \in V, \qquad c_{ij} = 0 \; \forall ij \in A \qquad ts \in A$

$\qquad c_{ts} = -1, \qquad u_{ts} = \infty$

Assignment problem min weighted bipartite matching,

$\qquad |V_1| = |V_2|, A \subseteq V_1 \times V_2$

$\qquad c_{ij}$

$\qquad b(i) = 1 \; \forall i \in V_1 \qquad b(i) = -1 \; \forall i \in V_2 \qquad u_{ij} = 1 \; \forall ij \in A$

Special cases

Transportation problem/Transhipment distribution of goods, warehouses-costumers
$$|V_1| \neq |V_2|, \qquad u_{ij} = \infty \text{ for all } ij \in A$$

$$\min \sum c_{ij} x_{ij}$$
$$\sum_i x_{ij} \geq b_j \qquad\qquad \forall j$$
$$\sum_j x_{ij} \leq a_i \qquad\qquad \forall i$$
$$x_{ij} \geq 0$$

if $\sum a_i = \sum b_i$ then $\geq / \leq$ become $=$
if $\sum a_i > \sum b_i$ then add dummy tank nodes
if $\sum a_i < \sum b_i$ then infeasible

Multi-commodity flow problem   ship several commodities using the same network, different origin destination pairs separate mass balance constraints, share capacity constraints, min overall flow

$$\min \sum_k \mathbf{c}^k \mathbf{x}^k$$
$$N\mathbf{x}^k \geq \mathbf{b}^k \qquad \forall k$$
$$\sum_k \mathbf{x}^k_{ij} \leq \mathbf{u}_{ij} \qquad \forall ij \in A$$
$$0 \leq \mathbf{x}^k_{ij} \leq \mathbf{u}^k_{ij}$$

What is the structure of the matrix now? Is the matrix still TUM?

Application Example
Ship loading problem

Plenty of applications. See Ahuja Magnanti Orlin, Network Flows, 1993

- A cargo company (eg, Maersk) uses a ship with a capacity to carry at most r units of cargo.
- The ship sails on a long route (say from Southampton to Alexandria) with several stops at ports in between.
- At these ports cargo may be unloaded and new cargo loaded.
- At each port there is an amount b_{ij} of cargo which is waiting to be shipped from port i to port $j > i$
- Let f_{ij} denote the income for the company from transporting one unit of cargo from port i to port j.
- The goal is to plan how much cargo to load at each port so as to maximize the total income while never exceeding ship's capacity.
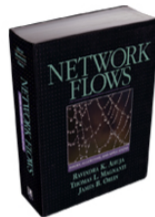
Application Example: Modeling

- n number of stops including the starting port and the terminal port.

- $N = (V, A, \mathbf{l} \equiv \mathbf{0}, \mathbf{u}, \mathbf{c})$ be the network defined as follows:

  - $V = \{v_1, v_2, ..., v_n\} \cup \{v_{ij} : 1 \le i < j \le n\}$

  - $A = \{v_1 v_2, v_2 v_3, ... v_{n-1} v_n\} \cup \{v_{ij} v_i, v_{ij} v_j : 1 \le i < j \le n\}$

  - capacity: $u_{v_i v_{i+1}} = r$ for $i = 1, 2, ..., n-1$ and all other arcs have capacity ∞.

  - cost: $c_{v_{ij} v_i} = -f_{ij}$ for $1 \le i < j \le n$ and all other arcs have cost zero (including those of the form $v_{ij} v_j$)

  - balance vector: $b(v_{ij}) = b_{ij}$ for $1 \le i < j \le n$ and the balance vector of $b(v_i) = -b_{1i} - b_{2i} - ... - b_{i-1,i}$ for $i = 1, 2, ..., n$

Application Example: Modeling
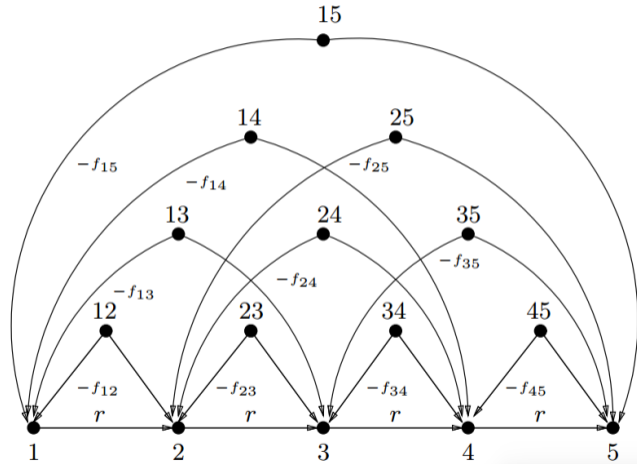
Application Example: Modeling

Claim: the network models the ship loading problem.

- suppose that $t_{12}, t_{13}, ..., t_{1n}, t_{23}, ..., t_{n-1,n}$ are cargo numbers, where t_{ij} $(\leq b_{ij})$ is the amount of cargo the ship will transport from port i to port j and that the ship is never loaded above capacity.

- total income is

$$I = \sum_{1 \leq i < j \leq n} t_{ij} f_{ij}$$

- Let x be the flow in N defined as follows:
  - flow on an arc of the form $v_{ij} v_i$ is t_{ij}
  - flow on an arc of the form $v_{ij} v_j$ is $b_{ij} - t_{ij}$
  - flow on an arc of the form $v_i v_{i+1}$, $i = 1, 2, ..., n-1$, is the sum of those t_{ab} for which $a \leq i$ and $b \geq i + 1$.

- since t_{ij}, $1 \leq i < j \leq n$, are legal cargo numbers then x is feasible with respect to the balance vector and the capacity restriction.

- the cost of x is $-I$.

Application Example: Modeling

- Conversely, suppose that x is a feasible flow in N of cost J.

- we construct a feasible cargo assignment $s_{ij}, 1 \leq i < j \leq n$ as follows:

    - let s_{ij} be the value of x on the arc $v_{ij}v_i$.

- income $-J$

Outline

Assignment Problem

Input: a set of persons $P_1, P_2, ..., P_n$, a set of jobs $J_1, J_2, ..., J_n$ and an $n \times n$ matrix $M = [M_{ij}]$ whose entries are non-negative integers. Here M_{ij} is a measure for the skill of person P_i in performing job J_j (the lower the number the better P_i performs job J_j).
Goal is to find an assignment π of persons to jobs so that each person gets exactly one job and the sum $\sum_{i=1}^{n} M_{i\pi(i)}$ is minimized.

Matching Algorithms

Matching: $M \subseteq E$ of pairwise non adjacent edges

- bipartite graphs
- arbitrary graphs

- cardinality (max or perfect)
- weighted

Assignment problem $\equiv$ min weighted perfect bipartite matching $\equiv$ special case of min cost flow

Transportation Problem

Given: a set of production plants $S_1, S_2, ..., S_m$ that produce a certain product to be shipped to a set of re-tailers $T_1, T_2, ..., T_n$. For each pair (Si, Tj) there is a real-valued cost c_{ij} of transporting one unit of the product from S_i to T_j. Each plant produces $a_i, i = 1, 2, ..., m$, units per time unit and each retailer needs $b_j, j = 1, 2, ..., n$, units of the product per time unit.

Goal: find a transportation schedule for the whole production (i.e., how many units to send from S_i to T_j for $i = 1, 2, ..., m$, $j = 1, 2, ..., n$) in order to minimize the total transportation cost.

We assume that $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Summary

1. Well Solved Problems

2. (Minimum Cost) Network Flows

3. Assignment and Transportation