Optimization Michael Tehranchi
Lecture 12 - Easter 2009

The transportation problem and transportation algorithm

- m suppliers, supplying $S_1, \dots, S_m$.
- n destinations, demanding $D_1, \dots, D_n$.
- Supplier i and destination j there is a unit cost d_{ij} of transport.

Assumption. Total supply equals total demand:

$$\sum_{i=1}^{m} S_i = \sum_{i=1}^{n} D_j$$

The *transportation problem* is the linear programming problem:

$$\text{minimize } \textstyle\sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij} \quad \text{subject to} \quad \begin{array}{rcl} \sum_{j=1}^n x_{ij} & = & S_i \text{ for all } i = 1, \dots, m \\ \sum_{i=1}^m x_{ij} & = & D_j \text{ for all } j = 1, \dots, n \\ x_{ij} & \geq & 0 \text{ for all } i, j. \end{array}$$

Theorem. The flow $(x^*_{ij})_{i,j}$ is optimal if there exists $\lambda^* \in \mathbb{R}^m$ and $\mu^* \in \mathbb{R}^n$ such that

- $(x^*_{ij})_{i,j}$ is feasible, (primal feasibility)
- $\lambda^*_i + \mu^*_j \leq d_{ij}$ for all i, j (dual feasibility)
- $(\lambda^*_i + \mu^*_j - d_{ij}) x^*_{ij} = 0$ (complementary slackness)

Proof. If x is feasible

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij} &= \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij} + \sum_{i=1}^m \lambda^*_i \left(S_i - \sum_{j=1}^n x_{ij} \right) + \sum_{j=1}^n \mu^*_j \left(D_j - \sum_{i=1}^m x_{ij} \right) \\ &= \sum_{i=1}^m \sum_{j=1}^n (d_{ij} - \lambda^*_i - \mu^*_j) x_{ij} + \sum_{i=1}^m \lambda^*_i S_i + \sum_{j=1}^n \mu^*_j D_j \\ &\geq \sum_{i=1}^m \lambda^*_i S_i + \sum_{j=1}^n \mu^*_j D_j \end{aligned}$$

with equality if $x = x^*$. $\square$

The *transportation algorithm.*

(0) *Find an initial feasible assignment.* Note that there are $m + n - 1$ linear relations contraining the feasible set ($\sum_j x_{ij} = S_i$ and $\sum_i x_{ij} = D_j$ but $\sum_i S_i = \sum_j D_j$.) Hence, any b.f.s. will have $m + n - 1$ basic variables. Popular ways to find an initial b.f.s.:

- The *North-West method* is to put as much flow as feasible in the top left corner. Then move either down or to the right, and put as much as feasible in this cell, and so forth.
- The *greedy algorithm* is to put as much flow as feasible in the cell with the smallest cost. After adjusting the remaining capacities (which in effect deletes a row or column from the table), put as much flow as feasible into the cell with the second smallest cost, and so forth.

(1) *Assign Lagrange multipliers.* We may take $\lambda_1 = 0$. We enforce complementary slackness by choosing λ_i and ν_j so that $\lambda_i + \mu_j = d_{ij}$ for each basic cell.

(2) *Test for optimality.* The Lagrange multipliers are dual feasible if $\lambda_i + \mu_j \leq d_{ij}$ for each i, j. If all cells satisfy this inequality, then STOP!

Otherwise, if $\lambda_i + \mu_j > d_{ij}$, put the difference $\lambda_i + \mu_j - d_{ij}$ in the cell.

(3) *Pivot.* Pick one of the cells such that $\lambda_i + \mu_j > d_{ij}$ to add flow. The rule of thumb is to pick the cell with the largest difference $\lambda_i + \mu_j - d_{ij}$.

Put an amount $\varepsilon > 0$ units of flow into the pivot cell. At the same time, add or subtract ε from the basic cells to maintain feasibility.

Now choose the largest ε possible such that the flow is feasible.

(4) Go to step (1).

Remark. If the problem is originally posed such that total supply exceeds total demand $\sum_{i=1}^m S_i > \sum_{j=1}^n D_j$ then we can reformulate the problem by adding another destination with demand $D_{n+1} = \sum_{i=1}^m S_i - \sum_{j=1}^n D_j$ and set the unit cost $d_{i,n+1}$ of transport to zero for all suppliers i.