ICGTA08
Coimbatore, december 10-13, 2008

An IO–Modification of Potential Method

Lavoslav Čaklović

University of Zagreb, Croatia

caklovic@math.hr

Contents

1 Preference flow (one criterion)

Let $G = (V, \mathcal{A})$ (directed) preference graph. A function $\mathcal{F} : \mathcal{A} \to \mathbb{R}_+ \cup \{0\}$ which assigns to each arc $\alpha \in \mathcal{A}$ its weight of preference is called a preference flow. $\mathcal{F}$ can be considered as a function on arcs-space. Then, $\mathcal{F}(-\alpha) := -\mathcal{F}(\alpha)$ takes sense.

2 Consensus flow (more criteria)

Each criterion C_i generates its own preference flow $\mathcal{F}_i$. Consensus graph $(V, \mathcal{A})$ and consensus flow $\mathcal{F}$ are defined as follows:

$$\forall \alpha = (u, v), \quad \mathcal{F}_\alpha := \sum_{\substack{i=1 \\ \pm\alpha \in \mathcal{A}_i}}^{k} w_i \mathcal{F}_i(\alpha) \tag{1}$$

w_i – weight of C_i.

consensus graph – demo

3 From flow $\mathcal{F}$ to ranking X

Definition 1. A preference flow $\mathcal{F}$ is consistent if it is orthogonal (in arcs-space) on each cycle or, equivalently, if the sum of flow components along each cycle equals zero.

Theorem 1. $\mathcal{F}$ is consistent iff it is an element of the columns-space of the icindence matrix of the preference graph.

For consistent flow ranking of nodes is obtained as follows: take any node and add (or substract) to it's neighbour the $\mathcal{F}$-value of the common[a] arc, i.e.

$$\alpha = (u, v) \implies X(v) - X(u) = \mathcal{F}_\alpha. \tag{2}$$

If $\mathcal{F}$ is not consistent we take it's consistent approx. $\mathcal{F}'$ and apply (2). Measure of inconsistency is defined as the angle between $\mathcal{F}$ and $\mathcal{F}'$.

[a] Obviously, graph should be connected.

In fact X is a solution of the Laplace equation

$$B^\tau B X = B^\tau \mathcal{F},$$

where B is the incidence matrix of the graph. To have uniqueness we put a condition

$$\sum_i X_i = 0.$$

4 Output/Input

In Data Envelopment Analysis (DEA) efficiency of a k-th Decision Making Unit (DMU) is defined as

$$\text{Eff}_k := \frac{\sum_{i=1}^{m} u_k^i}{\sum_{j=1}^{n} v_k^j} \qquad u_k^i - \text{outputs}, v_k^j - \text{inputs}.$$

Let's take $u_k^i = \log U_k^i, v_k^j = \log V_k^j$ and consider logarithm of Eff_k

$$\log \frac{\sum_{i=1}^{m} \log U_k^i}{\sum_{j=1}^{n} \log V_k^j} = \log\log \prod_{i=1}^{m} U_k^i - \log\log \prod_{j=1}^{n} \log V_k^j$$

or 'equivalently'

$$\Delta_k := \sum_{i=1}^{m} \log U_k^i - \sum_{j=1}^{n} \log V_k^j.$$

which is 'output-input' of the k-th node in decision graph.

In the presense of weights, say $u_i,\ i = 1, \ldots, m$, for outputs, and $v_j,\ j = 1, \ldots, n$, for inputs, we demand $\sum_{i=1}^{m} u_i = 1, \sum_{j=1}^{n} v_j = 1$

$$\begin{aligned}\Delta_k &= \sum_i u_i \log U_k^i - \sum_j v_j \log V_k^j \\ &= \sum_i \log (U_k^i)^{u_i} - \sum_j \log (V_k^j)^{v_j}\end{aligned}$$

and the corresponding consistent flow is

$$\begin{aligned}\mathcal{F}_{(k,l)} &= \log \prod_i \left(\frac{U_l^i}{U_k^i}\right)^{u_i} - \log \prod_j \left(\frac{V_l^j}{V_k^j}\right)^{v_j} \\ &= \log \left(\prod_i \left(\frac{U_l^i}{U_k^i}\right)^{u_i} \Big/ \prod_j \left(\frac{V_l^j}{V_k^j}\right)^{v_j} \right).\end{aligned}$$

the logarithm of relative weighted geometric mean of outputs divided by relative weighted geometric mean of intputs.

5 An example

Bank	Branch	Empl.	Assets**	Profits**	DEA	PM
Sanwa	395	13,998	52,999,340	399,398	1	0.155
Dai-ichi K.	407	17,837	53,438,938	411,149	1	0.129
Tokyo M.	419	20,235	78,170,694	504,422	1	0.127
Fuji	363	15,188	51,368,976	371,364	0.981	0.142
Sumimoto	397	15,710	56,468,571	353,531	0.854	0.115
Daiwa	218	8,671	16,665,573	112,121	0.853	0.106
Tokai	305	11,546	31,376,180	186,640	0.766	0.098
Sakura	529	18,805	52,535,262	302,901	0.742	0.066
Asahi	436	13,149	29,531,193	170,835	0.728	0.061

** Unit = One million yen (8300 USD)

Table 1: City Banks of Japan

6 Online example

1. Japan city banks http://decision.math.hr/examples/japan.php
2. ranking efficient DMUs only (crosefficiency, selfranking)

Future project.

1. Triplets http://decision.math.hr/programs/triplets/
2. New algorithm on graphs