

## RAMモデルのためのDEAネットワークアルゴリズムの構築

\*02203080 東京理科大学 染谷 和広 SOMEYA Kazuhiro  
02501910 東京理科大学 本間 隆嗣 HOMMA Takashi

### 1 はじめに

DEA(Data Envelopment Analysis)の計算は、大きな計算時間を必要とするため、これまでさまざまなDEAアルゴリズムに関する研究がされてきた。Sueyoshi(1990)ではDMU(Decision Making Unit)集合をいくつかの部分集合に分類して効率化をはかっている。しかし、現在急速に発展するネットワーク環境のもとにしながら、それらの研究は単一のコンピュータを仮定したものばかりである。そこで、本研究では大規模DEA問題を効率よく解くためのDEAネットワークアルゴリズムの構築と実証を目的とする。

### 2 ネットワークの基本構造

DEAネットワークコンピューティングでは、研究室内の10BASE-Tによるスター構造LANを利用する。また各コンピュータは1台をサーバとし、残りをクライアントとする。

### 3 DEAネットワークアルゴリズム

#### 3.1 DEAモデル

本研究では加法モデルを発展させたRAMモデル(Cooper 他(1999))を用いる。

$$\begin{aligned} \max \quad & \frac{1}{m+s} \left( \sum_{i=1}^m \frac{s_i^-}{R_i^-} + \sum_{r=1}^s \frac{s_r^+}{R_r^+} \right) \\ \text{s.t.} \quad & \sum_{j \in J} x_{ij} \lambda_j + s_i^- = x_{ik}, \quad i = 1, \dots, m, \\ & \sum_{j \in J} y_{rj} \lambda_j - s_r^+ = y_{rk}, \quad r = 1, \dots, s, \\ & \sum_{j \in J} \lambda_j = 1, \\ & \lambda_j \geq 0 \quad (j \in J), \quad s_i^- \geq 0, \quad s_r^+ \geq 0. \end{aligned} \quad (1)$$

(1)式において、 $R_i^-, R_r^+$ は、それぞれ入出力値の上限と下限を表している。

$$R_i^- = \bar{x}_{ij} - \underline{x}_{ij}, \quad R_r^+ = \bar{y}_{ij} - \underline{y}_{ij} \quad (2)$$

各入出力値の上下限値は(3)式によって定式化される。

$$\begin{aligned} \bar{x}_i &= \max_j(x_{ij}), & \underline{x}_i &= \min_j(x_{ij}) \\ \bar{y}_r &= \max_j(y_{rj}), & \underline{y}_r &= \min_j(y_{rj}) \end{aligned} \quad (3)$$

RAM効率値 $\Gamma^*$ は以下のように表される。

$$\Gamma^* = 1 - \frac{1}{m+s} \left( \sum_{i=1}^m \frac{s_i^{-*}}{R_i^-} + \sum_{r=1}^s \frac{s_r^{+*}}{R_r^+} \right) \quad (4)$$

#### 3.2 アルゴリズムの概要

非効率なDMUを除外することによって効率化を図る。そのためにそれぞれのDMUをその性質ごとに、(5)式(6)式の分類を用いる。

準備段階としてSueyoshi(1990)により、すべてのDMU集合 $J$ を優位集合 $J_n$ と非優位集合 $J_d$ とに分ける。

Stage Iで $J_n$ の解析を行い、効率的フロンティアを形成する集合(E)のキャンディデイト(EC)を探す。

Stage IIではECのもとで、 $J_n$ の解析を行い、最終段階でEが明らかとなる。

Stage IIIでは、Eのもとで $J_d$ の解析を行う。

$$J_n = E \cup E' \cup IE' \quad \text{and} \quad J_d = IE \quad (5)$$

$$\begin{aligned} E &= \{k \in J_n \mid \lambda_k^* = 1, \lambda_j^* = 0 (\text{all } j \neq k \in J_n), \\ &\quad \text{and all slacks are zero}\} \\ E' &= \{k \in J_n \mid \lambda_k^* < 1, \lambda_j^* > 0 (\text{some } j \in E), \\ &\quad \text{and all slacks are zero}\} \\ IE' &= \{k \in J_n \mid \lambda_k^* = 0, \lambda_j^* > 0 (\text{some } j \in E), \\ &\quad \text{and at least one positive slack}\} \\ IE &= \{k \in J_d \mid \lambda_k^* = 0, \lambda_j^* > 0 (\text{some } j \in E), \\ &\quad \text{and at least one positive slack}\} \end{aligned} \quad (6)$$

#### 3.3 アルゴリズムの詳細

##### Stage I

##### $J_n$ へのDEAパラレルコンピューティング

サーバは、 $J_n$ をある一定の数(ブロックサイズ)で部分集合( $J_n = J_{n_1} \cup J_{n_2} \cup \dots \cup J_{n_z}$ )に分割する。クライアント $h(h = 1, \dots, z)$ は、 $J_{n_h}$ について次のプロセスを実行する。

ステップ $k$ :  $k$ 番目のDMUについて測定を行っていると、2つの部分集合を次のように定義する。

(a)  $EC_k^h = \{DMU_j \mid j \in EC^h \text{かつ } J_{n_h} \text{中の } j \text{番目の DMUはクライアント } h \text{が測定を完了している.}\}$

(b)  $B_k^h = \{DMU_j \mid j \in EC_k^h\} \cup \{DMU_j \mid J_{n_h} \text{の } j \text{番目の DMUはクライアント } h \text{が測定していない.}\}$

$EC_k^h$ はクライアント $h$ におけるEキャンディデイト( $EC^h$ )である。ここで、 $k = j(j \in B_k^h - E_k^h)$ とする。

k 番目の DMU は次の式で解かれる。

$$\begin{aligned} \max \quad & \frac{1}{m+s} \left( \sum_{i=1}^m \frac{s_i^-}{R_i^-} + \sum_{r=1}^s \frac{s_r^+}{R_r^+} \right) \\ \text{s.t.} \quad & \sum_{j \in B_k^h} x_{ij} \lambda_j + s_i^- = x_{ik}, \quad i = 1, \dots, m, \\ & \sum_{j \in B_k^h} y_{rj} \lambda_j - s_r^+ = y_{rk}, \quad r = 1, \dots, s, \\ & \sum_{j \in B_k^h} \lambda_j = 1, \\ & \lambda_j \geq 0 \quad (j \in B_k^h), \quad s_i^- \geq 0, \quad s_r^+ \geq 0. \end{aligned} \quad (7)$$

$B_k^h$  のサイズはアルゴリズムが進むにつれて減少し、Stage I の最終段階では  $B_k^h = EC_k^h = EC^h$  となる。ここで得られた結果により、次の分類を行う。

- (a)  $\lambda_k^* = 1, \lambda_j^* = 0$  (for all  $j \neq k$ ) なら、 $DMU_k$  は  $EC^h$  に属する。 $B_{k+1}^h = B_k^h, EC_{k+1}^h = EC_k^h \cup \{DMU_k\}$  とする。
- (b)  $DMU_k$  が  $EC^h$  に属さないとき、 $B_{k+1}^h = B_k^h - \{DMU_k\}, EC_{k+1}^h = EC_k^h$  とする。

#### Stage II

##### $J_n$ への DEA ネットワークコンピューティング

クライアント h は EC の情報を受信した後、サーバ、クライアント間で互いに  $J_n$  の情報をやり取りしながら DEA ネットワークコンピューティングを行う。

-サーバ-

ステップ k : クライアント h に  $DMU_k (k \in J_{nh})$  の情報を送る。 $DMU_k$  をクライアント h が解き、その情報をサーバが受信する。

-クライアント h-

ステップ k : (8) 式により  $DMU_k$  を解析する。

$$\begin{aligned} \max \quad & \frac{1}{m+s} \left( \sum_{i=1}^m \frac{s_i^-}{R_i^-} + \sum_{r=1}^s \frac{s_r^+}{R_r^+} \right) \\ \text{s.t.} \quad & \sum_{j \in EC_k} x_{ij} \lambda_j + s_i^- = x_{ik}, \quad i = 1, \dots, m, \\ & \sum_{j \in EC_k} y_{rj} \lambda_j - s_r^+ = y_{rk}, \quad r = 1, \dots, s, \\ & \sum_{j \in EC_k} \lambda_j = 1, \\ & \lambda_j \geq 0 \quad (j \in EC_k), \quad s_i^- \geq 0, \quad s_r^+ \geq 0. \end{aligned} \quad (8)$$

計算結果によって以下のような分類を行う。

- (a)  $\lambda_k^* = 1, \lambda_j^* = 0$  (for all  $j \neq k$ ) なら、 $DMU_k$  は E に属する。 $EC_{k+1} = EC_k, E_{k+1} = E_k \cup \{DMU_k\}$  とする。
- (b)  $DMU_k$  が  $EC^h$  に属さないとき、 $EC_{k+1} = EC_k - \{DMU_k\}, E_{k+1} = E_k$  とする。

$EC_k$  のサイズはアルゴリズムが進むにつれて減少し、Stage II の最終段階では  $EC_k = E_k = E$  となる。

#### Stage III

##### $J_d$ への DEA パラレルコンピューティング

サーバは  $J_d$  を部分集合 ( $J_d = J_{d_1} \cup J_{d_2} \cup \dots \cup J_{d_z}$ ) に分割する。クライアント h は (9) 式によって、 $DMU_k (k \in J_{d_h})$  を解析する。

$$\begin{aligned} \max \quad & \frac{1}{m+s} \left( \sum_{i=1}^m \frac{s_i^-}{R_i^-} + \sum_{r=1}^s \frac{s_r^+}{R_r^+} \right) \\ \text{s.t.} \quad & \sum_{j \in E} x_{ij} \lambda_j + s_i^- = x_{ik}, \quad i = 1, \dots, m, \\ & \sum_{j \in E} y_{rj} \lambda_j - s_r^+ = y_{rk}, \quad r = 1, \dots, s, \\ & \sum_{j \in E} \lambda_j = 1, \\ & \lambda_j \geq 0 \quad (j \in E), \quad s_i^- \geq 0, \quad s_r^+ \geq 0. \end{aligned} \quad (9)$$

(9) 式では、 $\lambda_j (j \in E)$  に関する列のみ持つため、特に DMU 数が大きくなったときに効率化が進む。

## 4 シミュレーション

コンピュータ数の変化に伴って CPU 時間がどのように変化するかを調べた。その結果の一部抜粋を表 1 に示す。表 1 は全 DMU に対する効率的な DMU の割合が 20 パーセントの場合のシミュレーションの結果である。

コンピュータ数	DMU 数				
	1000	2000	3000	4000	5000
1	12806	53567	127971	244178	374542
2	8877	33379	71348	122088	189292
3	7313	23539	49340	88875	134530
4	6822	19956	40262	69357	104113
5	5142	16288	32779	57934	86861

## 5 結論と展望

今回、大規模 DEA 問題に対応するため、RAM モデルにおける DEA ネットワークアルゴリズムの構築を試みた。その結果、これまでの単一のコンピュータで行う計算と比べてより大規模な問題をより速い計算速度で解くことができた。

今回は LAN を用いたが、インターネットへの拡張や、ほかの数理計画問題への適用など今後の展望としてあげられる。

#### 参考文献

- [1] T.Sueyoshi, "Special Algorithm for an Additive Model in Data Envelopment Analysis," *Journal of the Operational Research Society*, Vol.41, No.3, pp.249-257, 1990.