

NUOPTを用いた大規模 DEA プログラムの開発

01014780	(株) NTTデータ	*井階 美歩	IKAI Miho
	(株) 数理システム	逸見 宣博	HENMI Nobuhiro
01405830	(株) NTTデータ	中川慶一郎	NAKAGAWA Keiichiro
01405390	専修大学	生田目 崇	NAMATAME Takashi
01307380	(株) 数理システム	田辺 隆人	TANABE Takahito

1 はじめに

DEA (Data Envelopment Analysis) は多入力多出力システムの相対的効率評価のための手法であり、1978年に Charnes, Cooper and Rhodes により発表されて以来、数多くの理論・応用両面での研究がなされてきた。ただし、研究的側面からの分析例が多く、ここではせいぜい数千の DMU の問題が解かれている。多くの既存の DEA ソフトウェアも Excel などのスプレッドシート上で分析しており、大規模問題を解くことは必ずしも目指していない。DEA は線形計画法を解くことにより効率評価を行うので、個々の問題は比較的解きやすい反面、DMU ごとの反復計算が必要となるため、大規模データに対しては計算機上の工夫が必要となる。本稿では、アルゴリズムを工夫した大規模データに対する DEA ソフトウェアを紹介する。

2 DEA と計算機への実装上の諸問題

DEA は分数計画問題として定式化されるが、実際には等価な線形計画問題が解かれる。したがって、比較的簡単にさまざまな事前の評価基準や問題特有の制約条件を計算モデルに取り入れることができ、提唱されて以来さまざまな拡張モデルが提案されている。このように DEA は最適化モジュールがあれば手軽に分析できる反面、末吉 [2] が指摘しているように、

- DMU 数だけ問題の一部を入れ替えながら同じ計算プロセスを繰り返す必要がある。
- DMU 数は入出力項目数よりかなり大きく、シンプレックス表の工夫により計算時間が短縮できる。
- 効率評価は効率的な DMU により形成される効率的フロンティアとの相対比較によりなされるので、効率的フロンティアを速く見つける必要がある。

といった点において、特に大規模問題を分析する際には現実的な時間内で計算するために、計算機上の工夫が必要である。

3 アルゴリズム

DEA の計算にとって線形計画法のモジュールそのものの高速性と信頼性も非常に重要な要因である。そこで、本稿で紹介する DEA プログラムは (株) 数理システムの最適化ソフトウェア NUOPT をベースとして、DEA のためのチューニングをおこなっている。

問題の定式化は基本的な CCR モデルに対して、制御不能項目を考慮すると共に領域限定などさまざまな制約条件を加味できるよう、一般的な線形の制約条件を付加できるようになっている。

計算順序ならびに基底構成のアルゴリズムは Sueyoshi [1] に準拠している。J を DMU の集合、 J_d を FDH の意味で支配されている DMU の集合、 $J_n = J \setminus J_d$ とする。Sueyoshi によるアルゴリズムは以下の通りである。

Stage 0: 各 DMU について、

$$g_j = \frac{\sum_r Y_{rj}}{\sum_i X_{ij}}$$

を求める。そして、各 DMU について J_n, J_d のいずれかに属するかの判定を g_j の降順に行う。

Stage 1: J_n に属する DMU について g_j の昇順に効率性を計算する。ただし基底は初期の J_n から非効率な DMU を削除して構成する。

Stage 2: J_n に属する DMU について g_j の昇順に効率性を計算する。ただし基底は Stage 1 で効率的と判定された DMU のみから構成される。

Stage 1において、 $DMU_j \in J_n$ の最適解 λ^* について $\lambda_k^* > 0$ となる DMU_k は [1] によると優先的に計算している。しかし、基底を縮小することによる計算の

表 1: 計算実験の結果

項目数	$ J $	$ E $	$ IE' $	$ J_d $	T_{total}	T_{1st}	T_{2nd}	I_{1st}	I_{2nd}
6	100000	56	2626	97318	2.14E+01	1.60E-01	1.73E-04	39	5.11
6	1000000	84	5253	994663	2.44E+02	1.63E+00	1.76E-04	7	4.99
10	100000	751	24609	74640	1.32E+03	7.00E-01	1.25E-02	80	15.78
10	1000000	1415	100626	897959	2.61E+04	4.04E+00	2.48E-02	67	14.09
16	10000	1847	7560	593	2.89E+02	2.10E-01	2.79E-02	84	34.21
16	100000	6038	76699	17263	3.52E+04	3.38E+00	3.45E-01	121	41.65
20	10000	3541	6417	42	5.42E+02	5.40E-01	5.27E-02	213	45.00
20	100000	14363	82542	3095	7.89E+04	3.92E+00	7.80E-01	112	60.05
26	10000	6052	3947	1	8.79E+02	7.40E-01	8.64E-02	257	53.80

高速化には DMU_k を先に計算してメリットはない。したがって、本ソフトウェアでは Stage 1 と Stage 2 をまとめて計算している。

また、単体法を解く上でも以下のような工夫を行っている。まず、最初の DMU については単体法により通常どおり解く。2つ目の DMU 以降は上記のアルゴリズムにより、比較的近い座標の DMU が順に解かれる。したがって、前回解いた解を初期解とすることで、最適解付近までの反復計算を大幅に削減できることが期待できる。したがって、2回目以降の計算については前回の基底解を初期解とする双対単体法により求解をおこなっている。

4 数値実験

(株) 数理システムによる実装時の CCR モデルの計算結果を表 1 に示す。計算機は Windows マシンで CPU は Pentium IV 1.5GHz である。表中の T_{total} は計算にかかった全所要時間、 T_{1st} は初回の DMU に対する計算時間、 T_{2nd} は2回目以降の DMU に対する平均計算時間である。また I_{1st} は初回の DMU の単体法の反復回数であり、 I_{2nd} は2回目以降の DMU に対する平均反復回数である。また、 $|E|$ は効率的 DMU の数であり、 $|IE'|$ は J_n に属する非効率的 DMU 数である。

この表を見ると初回の DMU の求解時間に対して、その後の DMU の求解時間は半分以下である。また単体法の反復回数も、最初の DMU に比べて2回目以降の計算では半減しており、本ソフトウェアの工夫が計算効率をあげている。

また、著者に知る範囲で最大規模のデータ集合についても実用時間内で解くことができた。

5 おわりに

本稿では、大規模問題をとくための DEA プログラムを紹介した。プログラムは C++ により実装されており、単独のプログラムとしても実行可能であるが、DLL 化することにより、Excel などのスプレッドシートからのデータのやり取りも可能である。

本稿で紹介したプログラムは単一コンピュータ上で実行するものである。しかし末吉 [3] が指摘しているように、DEA では DMU ごとに問題の一部を入れ替えた共通の問題を解いているため、クライアント=サーバ型ネットワークやグリッド・コンピューティング技術によりさらに高速化することが期待でき、それによりさらに大規模な問題を対応することができると考えられる。

参考文献

- [1] Sueyoshi, T.: "Algorithmic Strategy for Assurance Region Analysis in DEA," *Journal of Operations Research Society of Japan*, Vol.35, pp.62-76 (1992).
- [2] 末吉俊幸: 「DEA - 経営効率分析法 -」, 朝倉書店 (2001).
- [3] Seuyoshi, T. and Homma, T.: "DEA Network Computing in Multi-Stage Prallel Processes," *Journal of the Operational Research Society*, (forthcoming).