Algorithms for Multidimensional Frontier Visualization Based on Optimization Methods Using Distributed Computations

Alexander P. Afanasiev[1,2,3,4]   Vladimir E. Krivonozhko[2,3,5]   Andrey V. Lychev[2]
Oleg V. Sukhoroslov[1,4]

[1]Centre for Distributed Computing, Institute for Information Transmission Problems,
Russian Academy of Sciences, Nakhimovsky Prospekt 36-1, Moscow 117218, Russia
[2]National University of Science and Technology MISiS, Leninskiy Prospekt 4, Moscow 119049, Russia
[3]Lomonosov Moscow State University, GSP-1, Vorobievy Gory, Moscow 119991, Russia
[4]National Research University Higher School of Economics, Myasnitskaya str., 20, Moscow 101000, Russia
[5]Federal Research Center "Computer Science and Control", Russian Academy of Sciences,
Vavilov st. 44-2, Moscow 119333, Russia
E-mail: apa@iitp.ru, krivonozhkove@mail.ru, lychev@misis.ru, sukhoroslov@iitp.ru

Abstract

In data envelopment analysis, methods for constructing sections of the frontier have been recently proposed to visualize the production possibility set. The aim of this paper is to develop, prove and test the methods for the visualization of production possibility sets using distributed computations. In this article a general scheme of the algorithm for constructing sections (visualization) of production possibility set is proposed. An algorithm for constructing a generalized production function is described in detail. Also, the possibilities of implementing certain schemes in a distributed computing environment are considered.

1   Introduction

Despite the rapid development of information technologies in the last decade, relatively simple methods (based on ratio analysis) are used for performance evaluation of large complex systems (regions, financial organizations, municipalities, universities, etc.). In this approach, for each production unit a set of ratios is calculated. Then, some additive function, which is usually called the evaluation of complex objects, or simply rating function, is constructed from these ratios. However, in the papers [Krivonozhko et al., 2011, Krivonozhko et al., 2008] it was proved, that this approach significantly distorts the evaluation of a complex object. This happens because a multidimensional space is projected many times onto the subspaces during calculation of rating function. While significant distortions take place in every projection.

Data envelopment analysis (DEA) originated as a generalization of simple efficiency ratios over multidimensional case, i. e. when an activity of production units is described by a number of input and output indicators. A set of homogeneous production units is considered in order to make the formulation of this problem correct and intentional. As a result, this approach is reduced to the solution of a large family of optimization problems.

The founders of the DEA approach were famous scientists A. Charnes, W. Cooper, E. Rhodes, R. Banker and some others [Charnes et al., 1978, Banker et al., 1984]. At present the number of publications on this approach makes up several thousand units in the international scientific journals [Emrouznejad & Yang, 2017].

The ideas contained in the DEA approach have turned out to be much more seminal and far-reaching than the simple computation of efficiency scores of complex systems. The DEA approach has close links with the neoclassical theoretical economics, systems analysis, and multicriteria optimization. This approach allows one to analyze the behavior of complex systems in the multidimensional space, to find optimal paths of development in it, to model various scenarios.

However, the whole process of calculation in the DEA approach is hidden from the user. Every mathematical model is just an approximation of the real-life processes and phenomena. For this reason some inadequacies may arise in models. In the DEA scientific literature, some reports appeared that DEA results do not always coincide with experts' opinion. In our previous papers an approach for visualization of multidimensional production possibility sets and investigation of the behavior of complex units is proposed. The visualization of production possibility sets allows us to correct the DEA models using the frontier improving methods, reliably calculate important indicators of complex units (scale elasticity, marginal rates, etc.). Moreover, the visualization can reveal previously unexplored relationships between the variables in the model.

Visualization methods have been currently applied to the models that have a convex production possibility set: BCC (Banker, Charnes, Cooper) model with variable returns to scale, IRS (increasing returns to scale) model, DRS (decreasing returns to scale) model, etc. For a wide class of production models with nonconvex production possibility set, such methods and visualization algorithms are not developed. This paper aims to develop, prove and test the methods for the visualization of production possibility sets using distributed computations.

2 Background

Consider a set of n observations of actual production units (X_j, Y_j), $j = 1, \ldots, n$, where the vector of outputs $Y_j = (y_{1j}, \ldots, y_{rj}) \geq 0$ is produced from the vector of inputs $X_j = (x_{1j}, \ldots, x_{mj}) \geq 0$. The input-oriented model [Banker et al., 1984] is written as follows

$$
\begin{aligned}
&\min \theta \\
&\sum_{j=1}^{n} X_j \lambda_j + S^- = \theta X_0, \\
&\sum_{j=1}^{n} Y_j \lambda_j - S^+ = Y_0, \\
&\sum_{j=1}^{n} \lambda_j = 1, \ \lambda_j \geq 0, \ j = 1, \ldots, n, \\
&s_k^- \geq 0, \ k = 1, \ldots, m, \ s_i^+ \geq 0, \ i = 1, \ldots, r,
\end{aligned}
\tag{1}
$$

where $S^- = (s_1^-, \ldots, s_m^-)$ and $S^+ = (s_1^+, \ldots, s_r^+)$ are slack variables. In model (1) the optimal value θ^* describes the efficiency score of unit (X_o, Y_o), where (X_o, Y_o) is a unit from the set of production units (X_j, Y_j), $j = 1, \ldots, n$.

Notice that we do not use an infinitesimal constant explicitly in the DEA models, since we suppose that each model is solved in two stages in order to separate efficient and weakly efficient units [Cooper et al., 2007].

Definition 1. [Cooper et al., 2007] Unit $(X_o, Y_o) \in T$ is called BCC-efficient with respect to the input-oriented BCC model if and only if any optimal solution of (1) satisfies: a) $\theta^* = 1$, b) all slacks s_k^-, $k = 1, \ldots, m$, s_i^+, $i = 1, \ldots, r$ are zero.

If condition (a) in Definition 1 is satisfied, then unit (X_o, Y_o) is called input weakly efficient with respect to the BCC model.

Definition 2. [Cooper et al., 2007] Unit $(X', Y') \in T$ is Pareto efficient if and only if there is no $(X, Y) \in T$ and $(X, Y) \neq (X', Y')$ such that $X \leq X'$ and $Y \geq Y'$.

The production possibility set T for BCC model is formulated as follows

$$
T^{BCC} = \left\{ (X, Y) \ \middle| \ \sum_{j=1}^{n} X_j, \lambda_j \leq X, \ \sum_{j=1}^{n} Y_j, \lambda_j \geq Y, \ \sum_{j=1}^{n} \lambda_j = 1, \ \lambda_j \geq 0, \ j = 1, \ldots, n \right\}.
\tag{2}
$$

It was proved in [Krivonozhko et al., 2009] that the BCC model generalize a wide class of DEA models. Therefore, in this paper, we dwell mainly on this model.

3 Main Results

Define the intersection of the frontier with two-dimensional plane [Krivonozhko et al., 2004]

$$\mathrm{Sec}(X_o, Y_o, d_1, d_2) = \big\{ (X, Y) \mid (X, Y) \in \mathrm{Pl}(X_o, Y_o, d_1, d_2) \cap \mathrm{WEff}_P\, T \big\}, \tag{3}$$

where $\mathrm{Pl}(X_o, Y_o, d_1, d_2)$ is two-dimensional plane going through point (X_o, Y_o) and spanned by vectors $d_1, d_2 \in E^{m+r}$, $\mathrm{WEff}_P\, T$ is a set of weakly Pareto efficient points of set T. It is proved in [Krivonozhko et al., 2005; JORS] that $\mathrm{WEff}_P\, T$ coincides with a set of boundary points of T.

Define three types of two-dimensional sections that we will use in this paper.

1. Input isoquant, section S_1. In this case, we take the following direction vectors $d_1 = (e_p, 0) \in E^{m+r}$, $d_2 = (e_s, 0) \in E^{m+r}$, where e_p and e_s are m-identity vectors with a one in positions p and s, respectively.

2. Output isoquant, section S_2. For this section the direction vectors are chosen as $d_1 = (0, e_p) \in E^{m+r}$, $d_2 = (0, e_s) \in E^{m+r}$, e_p and e_s are r-identity vectors with a one in positions p and s, respectively.

3. Section S_3 is a generalized production function for unit (X_o, Y_o). For this case we use the following directions: $d_1 = (X_o, 0) \in E^{m+r}$, $d_2 = (0, Y_o) \in E^{m+r}$.

Next, we describe the algorithm for constructing a generalized production function.

Algorithm

Step 1. Find a leftmost point on a curve.

Let $d_1 = (X_o, 0) \in E^{m+r}$, $d_2 = (0, Y_o) \in E^{m+r}$,

a) solve the following optimization problem

$$\begin{aligned} &\min \beta_1 \\ &(Z_o + \beta_1 d_1 + \tau d_2) \in T, \end{aligned} \tag{4}$$

where τ is a free variable.

Set $Z_1 = Z_o + \beta_1^* d_1 + \tau^* d_2$, where β_1^* and τ^* are optimal variables in problem (4).

b) Find a leftmost vertex on a curve. Solve the following optimization problem

$$\begin{aligned} &\max \beta_2 \\ &(Z_1 + \beta_2 d_2) \in T \end{aligned} \tag{5}$$

Set $Z_1^1 = Z_1 + \beta_2^* d_2$, where β_2^* is the optimal value of the objective function in (5).

Step 2. Find a topmost point on a curve. Solve two following optimization problems.

a) Solve

$$\begin{aligned} &\max \beta_2 \\ &(Z_o + \tau d_1 + \beta_2 d_2) \in T \end{aligned} \tag{6}$$

Set $Z_2 = Z_o + \tau^* d_1 + \beta_2^* d_2$, where β_2^* and τ^* are optimal variables in (6).

b) Solve

$$\begin{aligned} &\min \beta_1 \\ &(Z_2 + \beta_1 d_1) \in T \end{aligned} \tag{7}$$

Set $Z_2^1 = Z_2 + \beta_1^* d_1$, where β_1^* is the optimal value of the objective function in (7).

Step 3. Set $l = 1$, $k = 1$, $i_1 = 1$, $i_2 = 2$. Create flow F_k^l, containing points $Z_{i_1}^l = Z_1^1$, $Z_{i_2}^l = Z_2^1$ of production possibility set T.

Define set $M = \{Z_1^1, Z_2^1\}$.

Step 4. While exist unprocessed flows F_k^l, perform the following computations.

For each flow solve optimization problem of the following type

$$\begin{aligned} &\max \beta_1 \\ &(G + \beta_1 d_1 + \tau d_2) \in T, \end{aligned} \tag{8}$$

where τ is a free variable,

$$G = \frac{1}{2}\Big(Z_{i_1}^l + Z_{i_2}^l \Big),$$

vector d_1 is perpendicular to the vector d_2, it lies in the plane of the section, and is directed to the upper left corner of a two-dimensional section, vector $d_2 = Z_{i_2}^l - Z_{i_1}^l$.

If optimal objective value of problem (8) $\beta_1^* > 0$, then start new flows $F_{k_1}^{l+1}$ and $F_{k_2}^{l+1}$ to solve optimization sub-problems.

Flow $F_{k_1}^{l+1}$ contains points

$$Z_{i_1}^{l+1} = Z_{i_1}^l, \quad Z_{i_2}^{l+1} = \frac{1}{2}(Z_{i_1}^l + Z_{i_2}^l) + \beta_1^* d_1 + \tau^* d_2, \tag{9}$$

where β_1^* and τ^* are optimal values of variables in problem (8).

Flow $F_{k_2}^{l+1}$ contains points

$$Z_{i_3}^{l+1} = Z_{i_2}^{l+1}, \quad Z_{i_4}^{l+1} = Z_{i_2}^l. \tag{10}$$

Set $l = l + 1$.

If optimal objective value in problem (8) $\beta_1^* \leq 0$, then points $Z_{i_1}^l$ and $Z_{i_2}^l$ are angular adjacent points of the segment of generalized production function. Add these points to the set of corner points of the production function M. Flow F_k^l is deleted from the list of flow tasks.

Step 5. Points of set M are angular points of generalized production function. Connect each adjacent pair of points by a straight line segment. Finally, add a horizontal line after the last point and a vertical line starting from the first point downwards. This completes construction of the generalized production function.
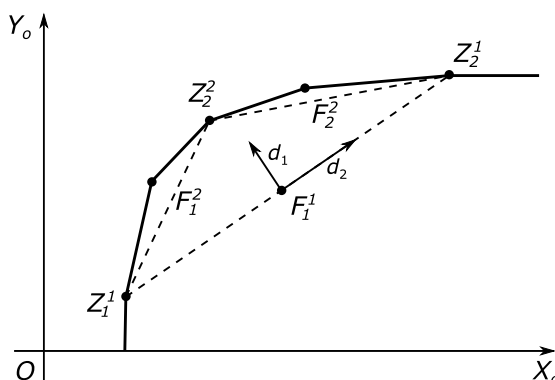

Figure 1: Graphical illustration of algorithm for construction of generalized production function

Figure 1 illustrates the steps of the algorithm for constructing a section of the generalized production function. At the first steps, the extreme points Z_1^1 and Z_2^1 are determined, then the flow F_1^1 containing these two points is started. At the next steps, as a result of solving problem (8), point Z_2^2 will be found; and the flow F_1^1 will be split into two flows F_1^2 and F_2^2 that will have vertices Z_1^1 and Z_2^2 for flow F_1^2, and vertices Z_2^2 and Z_2^1 for flow F_2^2.

Then the calculations are repeated until all segments of the generalized production function are found.

For the algorithm described above the following proposition holds.

Proposition. *Algorithm constructs section S_3 for the BCC model in a finite number of steps.*

In our computational experiments we used the software FrontierVision. This program allows us to visualize the multidimensional production possibility set by means of constructing two- and three-dimensional sections of the frontier. This software was specifically developed by authors for DEA models. The point is that the present-day optimization software are not suitable for the DEA models due to some specific features of these models. The reason of this is explained in detail in paper [Krivonozhko et al., 2004]. The FrontierVision is based on the algorithms developed by authors [Krivonozhko et al., 2004, Volodin et al., 2004, Krivonozhko et al., 2005; ITCS].

4 Implementation of the Algorithm in a Distributed Computing Environment

The described algorithm for constructing sections reduces the solution of the original problem to the solution of recursively generated sub-problems, similarly to the known "divide and conquer" scheme [Dasgupta, 2006]. Since the sub-problems available at each moment can be solved independently, it is possible to speed up the algorithm by parallel execution of these problems. In the case of a large amount of computations required, it is promising to use the resources of many computers with the help of distributed computing technologies.

This divide and conquer technique is the basis of many efficient algorithms in optimization theory. For example, Branch-and-Bound (B&B) algorithm use this scheme. The implementation of B&B method on desktop grid systems is considered in paper [Posypkin et al., 2017]. This distributed B&B implementation relies on BNB-Solver [Evtushenko et al., 2009]. A tool for simulating parallel B&B method and different load balancing strategies are considered in [Golubeva et al., 2016].

Consider the general scheme of algorithm implementation in the distributed computing environment. The implementation of the algorithm consists of two parts: the control part and the working part. The first part controls the calculation process, implementing the entire internal logic of the algorithm, except the solution of flow problems. The problems generated at the iterations of the algorithm are sent to solve the working part. After this the results are returned back to the algorithm and then combined to give a solution to the original problem. The working part implements the solution of flow problems. While the control part is a single process, the working part can consist of a many workflows running on various distributed computational resources.

An important part of the implementation of the control process is the strategy of distributing flow tasks to workers. This strategy largely determines the efficiency of the entire algorithm in a distributed environment. The strategy should minimize overhead, for example, not send workers small tasks, which can be solved locally more quickly. It is also necessary to ensure a balanced load of work processes, for example, based on an assessment of the complexity of tasks or the dynamic distribution of tasks.

To implement the algorithm for constructing sections in a distributed computing environment, it is planned to use the Everest platform [10]. Everest is a cloud-based software platform that supports the publication, execution, and composition of computing applications in a distributed environment. The advantage of using the Everest platform to implement the described algorithm is the availability of tools for creating distributed computing applications and integration with computing resources that do not require additional platform installation. In particular, the platform supports the implementation of multi-task applications that dynamically generate computational tasks that are processed on remote resources using the platform.

5 Conclusions

In this article one algorithm for constructing section, a generalized production function, is described in detail. The algorithms for constructing other sections differ mainly in the way the direction vectors d_1 and d_2 are selected at each iteration of the algorithm. Computational experiments using real-life datasets confirmed that the algorithm works reliably and construct sections correctly.

The general scheme of the algorithm for constructing sections (visualization) of a set T is described. Also, the possibilities of implementing certain schemes in a distributed computing environment are considered.

References

[Krivonozhko et al., 2011] Krivonozhko, V. E., Piskunov, A. A., & Lychev, A. V. (2011). On comparison of ratio analysis and data envelopment analysis as performance assessment tools. *IMA Journal of Management Mathematics 22(4)*, 357-370.

[Krivonozhko et al., 2008] Krivonozhko, V. E., Utkin, O. B., Safin, M. M., & Lychev, A. V. (2008). Transformation of the frontier in the efficiency analysis of complex systems. In S. V. Emel'yanov, & S. K. Korovin (Eds.). *Nonlinear dynamics and control*. Vol. 6. (pp. 279-304). Moscow: Fizmatlit. (in Russian)

[Charnes et al., 1978] Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring of efficiency of decision making units. *European Journal of Operational Research 2(6)*, 429-444.

[Banker et al., 1984] Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiency in data envelopment analysis. *Management Science 30(9)*, 1078-1092.

[Emrouznejad & Yang, 2017] Emrouznejad, A., & Yang, G. (2017). A survey and analysis of the first 40 years of scholarly literature in DEA: 1978-2016. *Socio-Economic Planning Sciences*. (in press) DOI: 10.1016/j.seps.2017.01.008

[Cooper et al., 2007] Cooper, W. W., Seiford, L. M., & Tone, K. (2007). *Data Envelopment Analysis. A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. Second edition. New York: Springer Science and Business Media.

[Krivonozhko et al., 2009] Krivonozhko, V. E., Utkin, O. B., Safin, M. M., & Lychev, A. V. (2009). On some generalization of the DEA models. *Journal of the Operational Research Society 60(11)*, 1518-1527.

[Krivonozhko et al., 2004] Krivonozhko, V. E., Utkin, O. B., Volodin, A. V., Sablin, I. A., & Patrin, M. (2004). Constructions of economic functions and calculations of marginal rates in DEA using parametric optimization methods. *Journal of the Operational Research Society, 55(10)*, 1049-1058.

[Krivonozhko et al., 2005; JORS] Krivonozhko, V. E., Utkin, O. B., Volodin, A. V., & Sablin, I. A. (2005). About the structure of boundary points in DEA. *Journal of the Operational Research Society 56(12)*, 1373-1378.

[Volodin et al., 2004] Volodin, A. V., Krivonozhko, V. E., Ryzhikh, D. A., & Utkin, O. B. (2004). Construction of Three-Dimensional Sections in Data Envelopment Analysis by Using Parametric Optimization Algorithms. *Computational Mathematics and Mathematical Physics 44(4)*, 589-603.

[Krivonozhko et al., 2005; ITCS] Krivonozhko, V. E., Utkin, O. B., Safin, M. M., & Lychev, A. V. (2005). Software EffiVision for analysis of complex systems. *Information technologies and computational systems 3*, 85-95. (In Russian)

[Dasgupta, 2006] Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2006). *Algorithms*. New York: McGraw-Hill Science/Engineering/Math.

[Evtushenko et al., 2009] Evtushenko, Y., Posypkin, M., & Sigal, I. (2009). A framework for parallel large-scale global optimization. *Computer Science-Research and Development 23(3-4)*, 211-215.

[Golubeva et al., 2016] Golubeva, Y., Orlov, Y., & Posypkin, M. (2016). A tool for simulating parallel branch-and-bound methods. *Open Engineering 6(1)*, 219-224.

[Posypkin et al., 2017] Posypkin, M., & Khrapov, N. (2017). Branch and bound method on desktop grid systems. *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 526-528.

[Sukhoroslov et al., 2015] Sukhoroslov, O., Volkov, S., & Afanasiev, A. (2015). A Web-Based Platform for Publication and Distributed Execution of Computing Applications. *14th International Symposium on Parallel and Distributed Computing (ISPDC)*, 175-184.