

COMPUTATIONS IN DEA

José H. Dulá

School of Business Administration
The University of Mississippi
University – MS 38677
E-mail: jdula@olemiss.edu

Received November 2001; accepted October 2002 after one revision.

Abstract

DEA is a well-established, widely used, and powerful analytical resource in the toolbox of the OR/MS analyst. It is used to assess relative efficiency of many, functionally similar, entities. It has applications in diverse areas including finance and banking, education, and healthcare. DEA is computationally intensive and, as the scale of applications grows, this intensity rapidly becomes one of the limiting factors in its utility. In this paper, we explore computations in DEA. We investigate the theory behind schemes, procedures and algorithms used in performing a DEA study and we report on current practices ranging from the basic and standard to the advanced and sophisticated. Our objective is to give researchers and practitioners an appreciation for the computational aspects of DEA that will permit them to understand the performance, problems, complications, limitations, as well as the potential of this technique.

Keywords: data envelopment analysis (DEA), linear programming, computational geometry.

1. Introduction

Data Envelopment Analysis (DEA), as originally proposed by Charnes *et al.* in 1978, is a non-parametric frontier estimation methodology for evaluating relative efficiencies and performance of a collection of related comparable entities (called Decision Making Units or DMUs) in transforming inputs into outputs. DEA's domain can be any group of many entities characterized by the same set of multiple attributes. DEA is a powerful quantitative tool that provides a means to obtain useful information about efficiency and performance of firms, organizations, and all sorts of functionally similar, somewhat autonomous, operating units. This methodology is nonparametric in the sense that it does not require an assumption about a functional form of the efficient frontier and, therefore, no parameter estimation, making it useful in a wide variety of applications. DEA clusters the entities as "efficient" or "inefficient" depending on their relative geometric location with respect to an empirical efficient frontier. The comparison is strictly in relation to the members of the subject group. DEA provides decision makers with information about how well subordinate units transform the resources they manage locally into the outputs that are necessary to achieve the operation's mission.

Modern DEA encompasses diverse areas and motivates work from researchers with sundry backgrounds. The field traces its origins to the seminal works in the economics literature of Debreu (1951), Koopmans (1951) and Farrell (1957) and, to this day, issues such as returns to scale of transformation processes motivate many new works with an economics perspective. DEA's relevance and impact is broad as witnessed by the eclectic array of applications where this methodology has been validated. DEA has been applied in education, finance, agriculture, sports, marketing, manufacturing; to name just a few. Implementation and application research frequently involves the participation of researchers in classical MS/OR. MS/OR can be said to be the home of DEA because of this and because of the methodology's inextricable relation with linear programming (LP). Other formal areas from which DEA extracts and contributes knowledge is convex analysis and computational geometry. All this can make works on DEA quite mathematical. Related to these areas, and of special interest here, are the algorithmic and computational aspects of DEA. DEA presents rich challenges in algorithm design and computational performance. The reader may begin to become familiar with a vast body of literature behind these different lines of research in DEA by turning to Seiford's (1996) work. This paper will be a contribution to the last of the topics above; namely, convex analysis, computational geometry, and algorithms.

This article is a comprehensive presentation of the theory behind computations in DEA. We consider the elements of DEA that impact computations; make connections with other areas that contribute and help understand these computations; explore the geometry of the objects behind DEA; and present the current procedures that involve computations when performing DEA analyses. We avoid facts, figures, and discussions that would date the paper; e.g., matters having to do with computational times since these depend so much on technology or commentaries on specialized commercial DEA software since this comes and goes and, as with technology, also evolves. We have elected to offer an expository narrative to make it enjoyable to read and easy to understand and to this end, we have kept notation and formulas to a minimum. However, this is not at the expense of accuracy and rigor.

2. DEA objectives and computations

The data set for a DEA study is a finite collection of points in multidimensional space. A DEA entity is defined by a vector of values; one for each attribute. To each entity, there corresponds a point and the dimension of each point is the number of attributes (e.g., inputs plus outputs) in the study. One assumption on technology can give rise to the four standard “convexified” returns to scale models in DEA (as opposed to the nonconvex “free disposal hull” model of, e.g., Deprins *et al.* (1984)). A full DEA analysis involves the individual “scoring” of each of the entities’ vector once the returns to scale assumption has been specified. This score depends on the values of the components of the vector somehow compared to the values of the rest of the data points in the study. The score is used to classify the entity as either efficient or inefficient.

The score of an entity in a DEA study depends on its relation to a constrained linear combination of the data set. The four different convexified returns to scale models are a result of different linear combinations of the same data set. In a *variable returns* transformation environment the comparison is made with convex combinations of the data. This is the, so called, “BCC” model of Banker *et al.* (1984). If the relative position with respect to the rest of the data is independent of any uniform scaling of the data points, the comparison is made with just nonnegative linear combinations of the data. This is the *constant returns* or “CCR” model of Charnes *et al.* (1978). In between variable and constant returns to scale assumptions we have *increasing* and *decreasing* returns. In the first, the linear combinations used in the comparison are allowed coefficients that add up to less than unity and in the other the coefficients can add up to values greater than one.

The four assumptions about returns to scale engender polyhedral sets that “envelop” the data in four different ways. Each of these sets is called a *production possibility set* in DEA. The polyhedral sets are defined by constrained linear operations of a finite list of points; therefore, they are an *external* representation in the sense of Rockafellar (1970), rather than an *internal* representation characterized by the intersection of halfspaces. In any case, as finitely generated polyhedral sets, they are always convex. The problem of scoring an entity under any one of these returns to scale assumptions requires locating the vector of the entity in the appropriate production possibility set with respect to a portion of its boundary. The specific portion of the boundary is referred to as the *efficient frontier*. The fundamental questions in DEA are: i) is the point in the interior or on the boundary? And, ii) if on the boundary, is it on the efficient frontier?

Whether a point is in the interior or on the boundary of a production possibility set can be ascertained using supporting hyperplanes. A point is on the boundary of the production possibility set if and only if there exists a supporting hyperplane with support set which includes the point. This relation between point and supporting hyperplane suggests two approaches. One is to start with a hyperplane and then proceed to identify data points in a support set of a translation of the hyperplane. Such points would necessarily be boundary points of the production possibility set. This idea is behind many preprocessors in DEA because of its computational simplicity. The second approach is to start with a point and then try to determine the existence of a supporting hyperplane there. This can be achieved by constructing a linear system such that a solution exists only if the point is on the boundary. The second approach, unlike the first, can be used systematically on each of the data points to establish conclusively their position relative to the boundary of the production possibility set. One practical way to establish the feasibility of a linear system is to use linear programming.

The second fundamental question posed above requires a difficult distinction. Determining whether a boundary point is on the efficient frontier of the production possibility set is not as straight forward as a determination about whether the point is on the boundary or not. The distinction as to where on the boundary a point is located causes theoretical and computational complications that many would rather, and actually do, ignore. The boundary of the production possibility set can be partitioned into two regions: the efficient frontier and the “free disposability” region. Part of the problem is that neither of the two regions is convex. The necessary and sufficient conditions to classify points in these regions exist; namely, a point is on the efficient frontier of the production possibility set if and only if there exists a supporting hyperplane at the point such that all the coefficients of the attributes are non-zero. The problem is that the existence of a supporting hyperplane at a point such that one or more of the coefficients of the attributes is zero is not sufficient to conclude that the point is not on the efficient frontier. Note that the condition requires that some hyperplane exist with the required characteristic. Data points on the free disposability region of the production possibility set are called *weak efficient*; a misnomer in terms of DEA since these are not points in the efficient frontier. A commentary to the original DEA LP proposed by Charnes *et al.* (1978, 1979) addresses this issue directly. Their LP provides necessary and sufficient conditions for a point to be on the efficient frontier. The implementation of this LP, however, is problematic because it requires an explicit numerical value for an arbitrarily small constant ϵ , so called, *nonArchimedean* constant. We will say more about the role of weak efficiency in DEA computations later.

Modern DEA and linear programming have been intimately and inextricably related since its origins in 1978. DEA was originally presented, interpreted, and understood in terms of the solution to a linear program in Charnes *et al.* (1978). Since then, many LP formulations have been proposed for DEA. All of them can be viewed as devices to identify the location of a point with respect to the boundary of the production possibility set. Different LPs, however, have different purposes and provide different information. For example, the original LP in Charnes *et al.* (1978) provides information about extending interior points to the boundary to provide *benchmarks* (suggested boundary counterparts that can be used to make recommendations for attaining efficiency) and its optimal solution conclusively classifies DMUs as efficient or inefficient. Other LPs are simplifications of this original formulation with the same benchmarking capabilities but without the assurance of a conclusive distinction between efficient and weak efficient DMUs. Yet others forego entirely benchmarking considerations and are formulated specifically to provide conclusive classifications. An example of such LPs are the *additive* forms of Charnes *et al.* (1985) and the *slack-based measures* generalizations of Pastor *et al.* (1999) and Tone (2001). So, depending on the returns to scale assumption, the information requirement of the analysis, and the approach to address the complications of weak efficiency, there is a multitude of LPs from which to choose and this selection is an important part of the modeling process.

We may, of course, solve either one of a primal-dual pair of LPs to obtain the information needed for a DEA classification. One of the problems in this pair will be the *multiplier* formulation and the other the *envelopment* formulation. The multiplier LP is in the space of the attributes plus one more structural variable for returns to scale when it is not constant. The optimal solution to a multiplier LP will be the parameters of a hyperplane that supports the production possibility set. Structural variables in envelopment formulations are associated with DEA data points. The optimal solution will be information on a subset of the data points that define a facet of the production possibility set where a benchmark is located

(efficient DMUs are their own benchmarks). This LP has a strong intuitive appeal and is typically the one use to explain, understand, and visualize DEA.

The solution to an LP will conclusively classify a point as interior or boundary with respect to the production possibility set. An obvious algorithm for DEA emerges from this property of these special LP formulations. It consists simply of iteratively solving an LP for every point in the data set. This is the “standard” basic algorithm for DEA and it is widely used in actual practice. It is easy to apply manually for small problems and can be coded directly in a language such as Visual Basic for Applications (VBA) when the data is in a spreadsheet. This algorithm requires the solution of as many linear programs as there are points in the data set. This means, however, that a DEA analysis on a large data set using the standard approaches is computationally intensive.

We can begin to appreciate the computational demands of a DEA analysis using the standard, LP-based, algorithm. Let us use the familiar linear regression procedure as a standard since the data sets used in the two methodologies are quite similar: a dense long and narrow rectangular matrix. Regression requires a sequence of elementary operations on dense matrices and the solution of a linear system. The solution of one LP is many times this amount work, as many times as the iterations required to attain optimality which, according to practical experience and in ideal circumstances, is roughly three times the number of rows of the technology matrix. A DEA analysis requires the solution of as many LPs as there are data points and, as it turns out, the circumstances are not always ideal. This means that DEA is orders of magnitude more computationally demanding and much more susceptible to the impact in the increase in the number of attributes and data points than regression.

The heavy computational demands of a DEA analysis have motivated considerable work on improving efficiency and performance. This work can be classified into three categories:

- 1) Preprocessors. There are several “quick and dirty” ways to identify boundary and/or interior DMUs. These procedures may be used to preprocess DEA data. They reveal the status of some (possibly all) DMUs quickly and efficiently without paying the full computational price of an LP. In general, preprocessing schemes consist of clever observations, quick assessments, and opportunistic calculations, which reveal the status of DMUs. The price paid for potential computational reductions is unpredictability and the absence of a guarantee of a conclusive resolution of the final status of all DMUs.
- 2) Enhancements and alterations to standard procedures. Besides all that can be done to speed up LPs, there are specific actions as the algorithm proceeds or to the sequencing of the instructions that have been shown to improve substantially the performance.
- 3) New algorithms. After all the improvements and modifications to the standard approach, there are still performance limits as the scale of the problem increases. New algorithms are required to solve problems that exceed these limits.

We now proceed to discuss technical aspects of DEA computations and the different ideas to improve performance. In the next section we see how DEA is related to problems in a seemingly unrelated area, computational geometry.

3. DEA and Computational Geometry

To understand the role of computations in DEA it is helpful to establish a connection between it and a well-known problem in computational geometry: the problem of identifying

the extreme points of the polyhedral hull of a finite collection of points. All we need in terms of notation to move ahead with our discussion is that there are n DMUs and m attributes per DMU.

A finitely generated polyhedral set is a convex polyhedron constructed using constrained linear operations on a finite list of data points called the set's *generators*. We will refer to these sets as *polyhedral hulls* of the points. The shape of a polyhedral hull depends on the nature of the linear operations and coefficient's constraints on the generators. Production possibility sets in DEA are the polyhedral hulls of the n data points corresponding to the DMUs in the model. The different returns to scale assumptions define the operations and constraints applied to the n generators which, in turn, define different polyhedral hulls. All these objects reside in \mathfrak{R}^m . The constant returns to scale assumption generates a cone with multiple extreme rays. The other returns to scale assumptions all define polyhedral hulls that are polyhedrons with (possibly) multiple extreme points. The problem of distinguishing efficient from inefficient in DEA and that of finding the extreme elements (points, for variable, increasing and decreasing returns to scale and rays, for constant returns to scale) of a polyhedral hull have much in common.

A familiar polyhedral hull is the convex hull. The polyhedral hulls generated in DEA are supersets of the convex hull of the data. Unlike convex hulls, DEA polyhedral hulls are unbounded and the recession cones depend on the returns to scale. All recession cones in DEA, however, contain a full orthant of \mathfrak{R}^m . For a rigorous description of DEA polyhedral sets refer to Dulá *et al.* (1995, 2001) and Dulá (1997).

Given a finite set of points in \mathfrak{R}^m and a polyhedral hull generated by them, we may ask two basic questions: i) What intersection of halfspaces define the hull, and ii) what are the extreme elements of the hull. The first question is notoriously hard. There is no escaping the fact that it will take a combinatorial number of operations to present the hull as a solution to a system of linear inequalities and, as the size of the problems increase, any such procedure will eventually explode. Any attempt at finding an efficient algorithm to attain this representation is doomed to failure. There is however, considerable amount of work on this question simply because there are many applications in computational geometry; e.g., Chand *et al.* (1970), Wets (1990); and many others, and Yu *et al.* (1996) and Olesen & Petersen (2001) for attempts to address this question in the context of DEA.

Fortunately for DEA, the second question above is not as difficult. Finding the extreme elements of the polyhedral hull of a finite list of points is a much simpler proposition. This problem has also been intensively studied because it has many applications in computational geometry, optimization, statistics, and, as we are about to see, DEA. Procedures and applications appear in Wets *et al.* (1967), Dulá *et al.* (1996, 1998). The extreme elements of a polyhedral hull will be a subset of the data points known as the *frame*. An important observation about frames of polyhedral hulls is that they are sufficient to express the original full hull; that is, the polyhedral hull of the frame is the same as the polyhedral hull of the entire data set. The question of finding a frame can be reduced to one of feasibility of linear systems since a point is extreme in a polyhedral hull if and only if it is not in the same sort of hull of the remaining points. When the question is one of feasibility of a linear system, the indicated approach is linear programming. The naïve algorithm that emerges from this observation is simply to apply a linear program to each of the data points, one at a time, to establish whether it is an element of the frame.

The naïve algorithm for finding the frame of a hull of a finite list of points sounds very similar to the standard procedure for DEA described in the previous section; namely, solve an LP for each of the data points. The similarity, actually, goes deeper. It turns out that the extreme elements of a DEA production possibility set will necessarily correspond to efficient DMUs (Dulá *et al.*, 2001). Therefore, finding the frame of a DEA production possibility set solves part of the problem of identifying the efficient DMUs. Moreover, the author's experience with large, randomly generated, data sets has shown that the complement of the frame is almost always the set of inefficient DMUs.

The connections between finding frames in polyhedral sets and DEA is a happy relation. Two collateral benefits are immediately evident: i) all previous work about frames of polyhedral hulls as in Wets *et al.* (1967), Dulá *et al.* (1992, 1996, 1998) bears directly on DEA, and ii) a natural algorithm for DEA is available based on finding frames first and using them to score the rest of the DMUs.

In the next section we begin our presentation about computations in DEA. We start it off with a discussion of preprocessors.

4. Preprocessors for DEA

Several preprocessing ideas have been proposed for the frame problem in computational geometry and these have been successfully applied in DEA. A preprocessor is effective when it conclusively determines the status of one or more DMUs without having to pay the full computational price of an LP solution. All preprocessors discussed here reduce to finding a support set for specified family of hyperplanes. Preprocessors fall into the following categories:

- 1) Sortings. Unique maximum and minimum attribute values over the entire data set can correspond to extreme points of a polyhedral hull depending on its recession cone Dulá *et al.* (1992). Therefore, simple sortings based on each attribute can detect efficient entities depending on the type of production possibility set. A sorting based on maximum and minimum attribute values corresponds to translating a special hyperplane the coefficients of which are all zero except for one attribute. These hyperplanes are perpendicular to one of the axes of \mathfrak{R}^m and parallel to the rest of the space. Other kinds of sortings that work as preprocessors are possible (see Rosen *et al.*, 1992). The effort involved in sorting is minimal and for the variable returns model, as many as $2m$ different efficient DMUs may be identified this way.
- 2) Translating Hyperplanes. The simple sorting idea above is generalized by using arbitrary hyperplanes. Unique data points that yield maximum or minimum level values for families of arbitrary parallel hyperplanes are extreme-point supports of the production possibility set. Any arbitrary hyperplane has the potential to uncover an extreme point or ray. Complications can occur in the event of ties since then the assurance that the supports are extreme elements is lost. Procedures based on such translating hyperplanes are relatively inexpensive requiring only m -dimensional inner products and the identification of the largest or smallest value in a list (see Dulá *et al.*, 1992). Because of the flexibility in the choice of hyperplanes, these preprocessors can reveal any number of new efficient DMUs, and, in the limit, they will find them all. They cannot, however, guarantee the identification of all efficient DMUs in a finite number of attempts.

- 3) Rotating hyperplanes. In the same way that a hyperplane can be translated until it supports a polyhedral hull, a family of rotating hyperplanes can be characterized. The idea that a supporting hyperplane “anchored” at an extreme point can be rotated to obtain new supporting hyperplanes that reveal the status of previously unknown extreme points is presented in Dulá *et al.* (2001b). The computational price amounts to inner products and minimum ratio tests.

The preprocessors described above are limited to identifying efficient DMUs since they are based on supporting hyperplanes. Work on preprocessors for identifying interior points is in the development stages; see e.g., Shaheen (2001).

In the next section, we discuss the standard procedure along with its known ameliorations and important variations.

5. The Standard Procedures for DEA

Implementing the standard procedure for DEA requires a decision about the DEA LP formulation. This choice can have an impact on the computational requirements of an analysis. The original LP proposed by Charnes *et al.* (1978) is what has come to be known as a nonArchimedean oriented formulation. In theory, such formulations conclusively classify efficient and inefficient DMUs based on any optimal solution and provide meaningful benchmarks based on the uniform scaling of either the inputs or outputs. However, as mentioned earlier, they are problematic in their implementation and may even result in incorrect classifications depending on machine tolerances and precision (Ali, 1994). Therefore, LP formulations that do not require setting values for nonArchimedean constants are interesting and available. The price paid for this convenience is, in some cases, the unavailability of information from optimal solutions for conclusive classifications. An LP formulation that provides sufficient conditions for classification of DMUs with every solution without the device of nonArchimedean constants was proposed by Charnes *et al.* (1985). It is the, so-called, additive LP. This LP however, is not suitable for many of the reasons that motivate a DEA analysis; efficiency studies to provide oriented benchmarking recommendations for inefficient DMUs. If the LP conclusively classifies efficient and inefficient DMUs then the standard procedure for DEA proceeds as follows:

Standard DEA Procedure:

FOR $J=1$ TO n DO:
 STEP 1. $J^* \leftarrow J$.
 STEP 2. SOLVE APPROPRIATE LP TO SCORE DMU J^* .
 STEP 3. CLASSIFY DMU J^* .

In the case of analyses with LPs that do not always provide sufficient conditions for efficiency or inefficiency classification, the standard procedure above needs to be modified. Such LPs are less discriminating and typically distinguish only between interior and boundary points. These LP arise naturally as when all nonArchimedean constants are set to zero in the original formulations or they may be especially constructed to provide specific benchmarking recommendations, e.g., the “unoriented” formulations of Bognol (2001). Ambiguity may arise when a point is on the boundary without satisfying the sufficient

condition for efficiency or inefficiency. In an envelopment formulation this typically is manifested by an appropriate optimal objective function value that places the point on the boundary but all slacks are zero. Positive slacks are sufficient to conclude weak efficiency but absent this, the point may or may not be efficient. In a multiplier formulation the ambiguous situation arises when at least one of the attributes' multiplier is zero for a point on the boundary. Strictly positive multipliers for all the attributes conclusively establish that the point is efficient. One zero multiplier precludes this conclusion. Such LPs require a two-stage approach for conclusive classification of all the DMUs.

Two-Stage Standard DEA Procedure:

```

FOR  $J=1$  TO  $n$  DO:
  STEP 1.  $J^* \leftarrow J$ .
  STEP 2. SOLVE APPROPRIATE LP TO SCORE DMU  $J^*$ .
  STEP 3. IF OBJECTIVE FUNCTION VALUE SATISFIES SUFFICIENT
          CONDITION FOR INTERIOR POINT:
          THEN.
              CLASSIFY DMU  $J^*$  AS INEFFICIENT.
          ELSE.
              IF SUFFICIENT CONDITION FOR
                EFFICIENCY/INEFFICIENT CLASSIFICATION
                SATISFIED
                THEN.
                    CLASSIFY DMU  $J^*$ .
              ELSE
                SOLVE "ADDITIVE" LP FORMULATION
                AND MAKE CONCLUSIVE
                CLASSIFICATION.

```

The problem of having to solve two LPs for every DMU for which no sufficient condition for classification is available may be somewhat mitigated by the use of deleted domain techniques.

The idea behind this technique consists of excluding the DMU being scored from the coefficient matrix of the LPs. This idea has been around since early in DEA computations and was probably practiced by many. It is mentioned in Banker & Gifford (1991), Charnes *et al.* (1992). It was not formally presented and discussed as an alternative to the conventional LPs until the papers by Andersen & Petersen (1993) and Bogetoft (1994). A rigorous study of the impact of domain deletion for the case of constant returns model appears in Thrall (1996), Dulá *et al.* (1997) and Seiford *et al.* (1999). The technique essentially provides all the same information as a complete LP plus additional insights about efficient DMUs including sufficient conditions to identify DMUs that correspond to extreme elements of the production possibility set. Such "extreme-efficient" DMUs are by far the most common type of efficient DMUs and are never weakly efficient; this is useful information when the LP is not nonArchimedean. The fact that LPs can be infeasible with this technique actually adds to its value since this is sufficient to conclude that the DMU being scored is extreme-efficient.

6. Enhancements to standard approaches

Enhancements and improvements for the standard approach are available. Besides all that can be done by applying what is known outside DEA about improving LP performance (e.g., multiple pricing, product forms, hot starts, etc.), there are techniques that exploit the special attributes of DEA LPs. Perhaps the two best known are Reduced Basis Entry (RBE) and Early Identification of Efficient DMUs (EIE) (see Ali, 1994). Both ideas are a consequence of the same result about DEA LPs; namely, that a DEA LP optimal solution provides information about a supporting hyperplane for the production possibility set and its support set. This translates to the fact that only variables associated with boundary DMUs can be basic at any (dual feasible) optimal solution in an envelopment form or, conversely, that only constraints associated with boundary entities can be equalities at optimal solutions of multiplier forms. RBE takes advantage of the fact that an inefficient entity is never in a support set and therefore, once identified as such, its data point can be omitted on any subsequent LPs to be solved. This idea is easy to implement as LPs are iteratively formulated and solved. The systematic application of this approach progressively reduces the size of the LPs that need to be tested. EIE simply states that if a DMU's variable appears in a basis of an optimal solution of an envelopment LP, or as its constraint is an equality at optimality in a multiplier form, then we have advance knowledge that it is a boundary DMU. EIE appears to have much less of an impact on improving performance than RBE. This is due to the fact that the proportion of efficient DMUs is relatively small compared to the number of data points, especially in large data sets. Another factor is that the list of efficient DMUs the variables for which appear as basic is frequently a subset of the full set of efficient DMUs. Moreover, if the LP is formulated to provide sufficient conditions for efficiency as with nonArchimedean or additive formulations, our knowledge is that the DMU is efficient under those circumstances. Durchholtz (1994) tested these two techniques together and reports a dramatic impact on reducing computation times. He also concludes that most of the impact on improvement is due to RBE.

An idea to speed up DEA computations is based on combining RBE with data partitioning schemes. The idea applies the principle that if an entity is inefficient with respect to a subset of the entities, it will be inefficient with respect to any superset. An implementation consists of partitioning the data set into uniformly sized "blocks" and independently applying standard DEA procedures to them to identify the inefficient entities within blocks. At this stage, the idea can be considered as simply a preprocessing scheme. Since the procedure can be repeated with new blocks of entities with unknown status, however, it becomes something more elaborate and effective. Repeating the process until a final block is created will result in eventually culling the bulk of inefficient entities. All entities are scored in a second phase using LPs composed of the entities which survived the culling; hopefully, a much smaller LP than would otherwise be used in a standard approach. The idea originates in the paper by Barr *et al.* (1997). They observe that the performance of such "hierarchical decomposition" techniques is affected by the size of the initial and intermediary blocks. This poses a problem since this decision has to be made explicitly by the analyst. Their experiments suggest that performance is quasiconvex in that times are monotonically nondecreasing as the block size deviates from an optimal value. Their tests indicate substantial time reductions in ideal implementations.

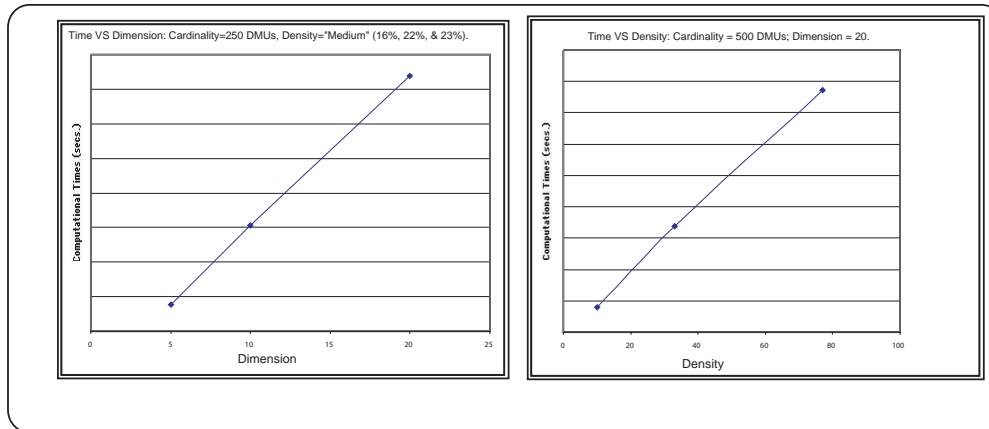


Figure 1.

Implementations of the standard procedures for DEA reveal that three factors affect computational performance: “Dimension” (number of inputs plus outputs), “Density” (proportion of DMUs that are efficient), and “Cardinality” (number of DMUs). All three have an adverse effect; that is, the greater their magnitude the more time is required to perform a DEA run. This effect is evident in the graphs in Figures 1 and 2. In Figure 1 we can see that, in a particular experiment we would consider typical, as either the dimension of the data set or its density increases (all else approximately equal), the effect is a clear increase in computational times. These experiments were executed with a single phase standard implementation enhanced with RBE.

Researchers have speculated on the nature of the relation between the cardinality, n , of a DEA problem and the time to solve it. In some papers (e.g., Barr *et al.*, 1997) this relation has been reported as exponential. Our tests confirm that the time required to solve DEA problems using the standard approach can be approximated by a quadratic function of the cardinality of the data set. Figure 2 depicts this relation for the actual case of five attributes and 4% efficiency. The results correspond to a standard implementation enhanced with RBE.

The actual relation between cardinality, n , and time, t , is approximated by the relation: $t = cn^2$, where the factor, c , depends on the cardinality and number of attributes. The factor would be unaffected by density in a pure (unenhanced) standard implementation. Efficiency density affects standard implementations enhanced with RBE. The impact is to favor lower densities since this means that more entities are excluded from LPs as the procedure iterates. Clearly, the hardware and software used will also affect this factor.

These days analysts who wish to program standard DEA procedures directly are likely to turn to a commercial spreadsheet. An excellent candidate for this would be Microsoft’s ‘Excel’ since it offers as part of the package an LP solver that is adequate for the task. It is a simple matter to program the spreadsheet to perform the single stage standard procedure. A complete macro using no more than 15 to 20 instructions is possible and quite effective for small problems. More sophisticated programming would be needed to implement the two-stage approach. Almost certainly, this has already been done and macros may be available through contacts or commercially. It is reasonable to expect that, one day, a macro for DEA will be bundled with the spreadsheet package in the same way as for, say, least-squares regression.

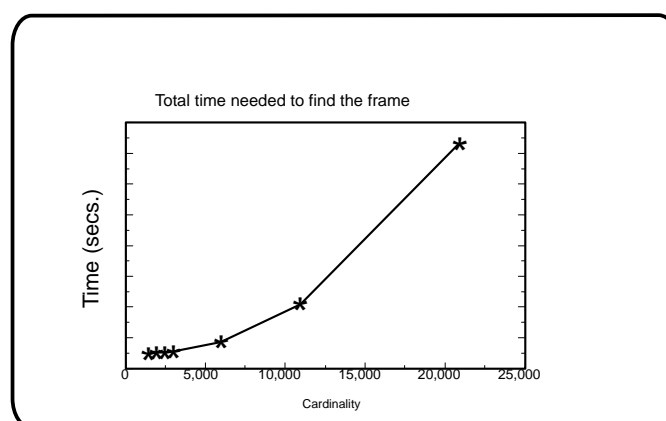


Figure 2.

7. Notes about Degeneracy and Parallel Implementations

An implementation of the standard procedure must contend with the possibility of numerical complications. A potentially serious one is connected to degeneracy and cycling. Ali (1994) observes that “As the number of input and output measures increases, the potential for encountering a very large number of degenerate pivots also increases. Barr *et al.* (1997) add: “our experience indicates that cycling in DEA codes is not only prevalent but likely, in the absence of anti-cycling procedures.” This concern has led to works such as Charnes *et al.* (1993) where an anti-degeneracy/cycling linear programming method especially for data envelopment analysis is introduced. A factor inducing degeneracy in DEA codes may be the inclusion of the column of the DMU being scored in the constraint matrix of envelopment LPs. This is a problem because the same column appears in the left and right hand sides of the linear system. One way to avoid this “induced degeneracy” is to use deleted domain formulations.

There is no natural sequence for the solution of the LPs in any of the procedures to solve DEA problems described so far. Therefore, one way to accelerate performance of these procedures is to distribute the load of solving individual LPs among several processors. In all procedures, including decomposition approaches, the LPs can be solved separately and the solutions independently advance the progress of the procedure. There is limited experience on parallelization schemes but indications are that substantial gains in time will be achieved through them. Durchholz (1994) reports that speedups with parallelization are nearly linear. Today's network architectures suggest obvious master/slave distribution scheme with the master assigning LPs to the subordinates and managing the information as it arrives. The coarse grained nature of such a parallelization is a good indication that substantial speed-ups are possible. Since all algorithms for DEA rely in the same way on the solution of linear programs, we might expect that the impact will be comparable across all the procedures. Certainly more work is needed to learn exactly how much can be gained.

8. Frame Algorithms and DEA

The frame is composed of the generators of the polyhedral hull that are its extreme elements. Therefore, a frame is a minimal cardinality subset of entities that generate the same polyhedral hull as the full data set.

The frame of a DEA production possibility set is composed of a subset of the data set that corresponds to extreme-efficient DMUs. Since the polyhedral hull of a frame is the same as the original polyhedral hull of the entire data set, the LP used to score DMUs can be reduced to include only the points in the frame. The optimal solution to such a reduced LP will provide all the same information as the original whole LP. From this result emerges the basic two-stage procedure for DEA using frames (Dulá *et al.*, 2001).

DEA Procedure using Frames:

STAGE 1. FIND FRAME FOR SPECIFIED RETURNS TO SCALE.

STEP 2. FORMULATE SCORING LP USING ONLY FRAME ELEMENTS AND SCORE ALL DMUS NOT IN THE FRAME.

In the first stage of the DEA procedure using frames, the frame is identified for the specific DEA model. Routines for identifying the frame of the data for the four returns to scale models may be based on any number of procedures. For example, results in Dulá *et al.* (1997) provide necessary and/or sufficient conditions for a DMU to be extreme-efficient based on the solution of deleted domain envelopment LPs. Specialized algorithms for finding frames have been proposed (Dulá *et al.*, 1996, 1998; López, 1999). These new frame algorithms build the frame sequentially, one element at a time. The process of erecting the frame begins with a single extreme point (or ray, in a polyhedral cone). A special linear program and inner products reveal a new, previously unidentified, extreme element. The process is repeated until all extreme elements are identified. These algorithms begin with small LPs the dimension of which grows one column (or row, in a dual) at a time. The final size of the LPs will be determined by the number of extreme elements; i.e., the extreme element “density”. A discussion of frame algorithms for different polyhedral hulls can be found in López (1999).

In the second stage of the DEA procedure using frames, all DMUs are scored using LPs composed of the frame elements. Therefore, when the frame is a small subset of the data set, the DMUs can be processed using much smaller LPs. Another interesting advantage of using this two-stage method is that the first stage is independent of the type of LP that is to be used to score the DMUs. This means that when the decision is made about the sort of LP to be used in the second stage, the bulk of the calculations are behind. This extends flexibility and functionality to the process since analyses using different LP formulations can be explored with relatively little additional work.

The two stage, frame based, DEA procedure using modern frame algorithms for the first stage is computationally superior to standard procedures, even when enhanced, especially when the cardinality, n , of the problems is large and the extreme element density is low. In that case, the LPs remain small in the first stage and in the second stage the same small LP is used to score the remaining DMUs. This limit on the size of the LPs translates to substantial savings in computational times. A comparison between the two procedures appears in Figure 3. We may verify from this figure how the dominance of the frame approach becomes dramatic as the problem size increases.

There are other advantages to frame-based algorithms besides faster computer times. One addresses the problems of induced degeneracy directly. The reason why induced degeneracy occurs disappears when using frame based procedures since the data for the DMU being scored never appears in both sides of the LP's system of constraints. Another advantage is that since the frame is composed exclusively of extreme elements of the production

possibility set these necessarily are not ever weakly efficient. Therefore, they need not be tested in the second stage. Moreover, the author's experience with randomly generated, high cardinality, low density data sets is that the vast majority of efficient DMUs are extreme efficient. Although efficient DMUs that are not extreme efficient can occur, they are a measure zero event in natural data. Even if weak efficient DMUs occur, scoring them with LPs composed of the frame is less likely to generate the type of situation where ambiguity about their classification might occur.

9. Extensions of DEA computations using frames

The interrelation among the four DEA frames can be exploited to gain knowledge about returns to scales without paying the full computational price that individual analyses would require. It turns out that the frame for a variable returns to scale production possibility set is a superset for the frames of the increasing, decreasing, and constant returns to scale production possibility sets (Dulá *et al.*, 2001). Another set of useful results is that the union of the increasing and decreasing returns to scale frame is the frame of the variable returns model and their intersection is the frame of the constant returns model. If we label these four frames as f^1 , f^2 , f^3 , and f^4 , for the variable, increasing, decreasing, and constant returns to scales production possibility set, respectively, then the results above are that i) $f^1 = f^2 \cup f^3$; and ii) $f^4 = f^2 \cap f^3$ (Dulá *et al.*, 2001).

These results have immediate computational implications since, after calculating any two frames from the list, $\{f^2, f^3, f^4\}$, the third may be obtained through simple set operations. One of these ideas is as follows.

Consider three routines VRFRAME(), IRFRAME(), and DRFRAME() the input argument for which is a DEA data set, A , and outputs will be frames as follows:

$$\begin{aligned} f^1 &\leftarrow \text{VRFRAME}(A), \\ f^2 &\leftarrow \text{IRFRAME}(A), \\ f^3 &\leftarrow \text{DRFRAME}(A). \end{aligned}$$

An important realization is that routines IRFRAME() and DRFRAME() can be used as follows:

$$\begin{aligned} f^2 &\leftarrow \text{IRFRAME}(f^1), \\ f^3 &\leftarrow \text{DRFRAME}(f^1). \end{aligned}$$

With this, we have enough to design a procedure for DEA analysis to identify of all four frames (Dulá *et al.*, 2001):

DEA Procedure for All Returns to Scale.

STAGE 1. IDENTIFY ALL FOUR FRAMES OF THE DEA DATA SET A .

STEP 1. $f^1 \leftarrow \text{VRFRAME}(A)$.

STEP 2. $f^2 \leftarrow \text{IRFRAME}(f^1)$.

STEP 3. $f^3 \leftarrow \text{DRFRAME}(f^1)$.

STEP 4. $f^4 \leftarrow f^2 \cap f^3$.

STAGE 2. SELECT RETURN TO SCALE MODEL(S) AND DEA LP FORMULATION AND SCORE DMUS WITH APPROPRIATE FRAME.

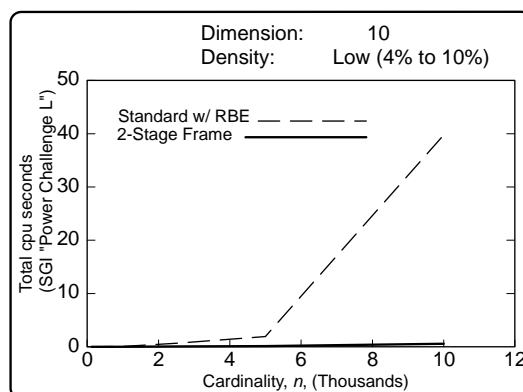


Figure 3.

Let us analyze some scenarios. In the worst case, the cardinalities of the frame and the entire data set are almost the same. In this case, the effort to find the frames of the data for the four models will require roughly three times the time to find the frame of the variable returns model. This since Step 4 is computationally inexpensive and effectively free. So, in a sense, we obtain four frames for the price of three. In real applications, however, such “dense” data sets do not seem to be the norm. A more realistic scenario is that the set of efficient DMUs is relatively small compared to the full DEA data set. In actual testing with this type of data, the four steps in Stage 1 are completed in little more than the time needed to execute Step 1 alone. This, combined with the fact that decisions about the DEA LP are left to the second stage where they can be performed using only frames, translates to significant increases in flexibility and speed especially when DEA analyses are over several models and multiple LP formulations.

These suppositions are actually borne out when tested with real data as we may see from Figure 4. The abscissa of the graph measures the ratio of the times it takes to calculate the three frames, f^2 , f^3 , f^4 , to just the frame f^1 . What is immediately apparent from Figure 4 is that the relation between frame density of the variable returns production possibility set and the time to find all four frames is linear and that the time to calculate the three frames f^2 , f^3 , and f^4 , is nearly negligible after f^1 has been found when the extreme point density is very low. It is important to reassert the fact that low extreme point density is indeed the case in actual applications, especially with large problems.

10. Conclusions

An important part of being an OR/MS practitioner who employs DEA is dealing with computations. There are examples of successful DEA applications with tens of thousands of entities. Massive data sets with hundreds of thousands, even millions, of entities are waiting to be analyzed. Studies with DEA can be predicted to go from the traditional one-time, cross-sectional, approach to dynamic, ‘instantaneous update’, tracking of DMUs. Such applications for DEA are not hard to conceive especially when we look around us and find already highly complicated financial, social, and technical systems and processes steadily growing and becoming more sophisticated. Contributions to reduce times and increase the information yield of a DEA study are needed to submit new problems to the understanding this quantitative tool provides. If these contributions are up to the task, DEA will emerge as one of the available tools for mining massive data sets.

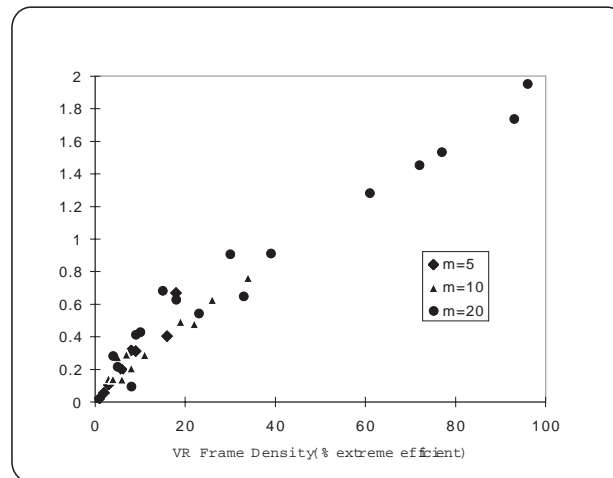


Figure 4.

Notes

Part of the work for this paper, especially the part about extensions of DEA computations using frames, was supported by ONR grant ONR 98342-0080. The author is grateful for this support.

References

- (1) Ali, A.I. (1994). Computational aspects of Data Envelopment Analysis. **In:** *DEA: Theory, Methodology and Applications* [edited by A. Charnes, W.W. Cooper, A.Y. Lewin, and L.M. Seiford], Kluwer Academic Publishers, Boston.
- (2) Andersen, P. & Petersen, N.C. (1993). A procedure for ranking efficient units in data envelopment analysis. *Management Science*, **39**, 1261-1264.
- (3) Banker, R.D. & Gifford, J.L. (1991). Relative efficiency analysis. Unpublished manuscript.
- (4) Banker, R.D.; Charnes, A. & Cooper, W.W. (1984). Some models for estimating technological and scale inefficiencies in Data Envelopment Analysis. *Management Science*, **30**, 1078-1092.
- (5) Barr, R.S. & Durchholz, M.L. (1997). Parallel and hierarchical decomposition approaches for solving large-scale Data Envelopment Analysis models. *Annals of Operations Research*, **73**, 339-372.
- (6) Bogetoft, P. (1994). Incentive Efficient Production Frontiers: An Agency Perspective on DEA. *Management Science*, **40**, 959-968.
- (7) Bougnol, M.-L. (2001). Nonparametric Frontier Analysis with Multiple Constituencies. Ph.D. Dissertation, The University of Mississippi, University, MS 38677.
- (8) Chand, D.R. & Kapur, S.S. (1970). An algorithm for convex polytopes. *J. of the ACM*, **17**(1).

-
- (9) Charnes, A.; Cooper, W.W. & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, **2**, 429-444.
- (10) Charnes, A.; Cooper, W.W. & Rhodes, E. (1979). Short Communication: Measuring the efficiency of decision making units. *European Journal of Operational Research*, **3**, p.339.
- (11) Charnes, A.; Cooper, W.W.; Golany, B.; Seiford, L. & Stutz, J. (1985). Foundations of Data Envelopment Analysis for Pareto-Koopmans efficient empirical production functions. *Journal of Econometrics*, **30**, 91-107.
- (12) Charnes, A.; Haag, S.; Jaska, P. & Semple, J. (1992). Sensitivity of efficiency classifications in the additive model of data envelopment analysis. *International Journal of Systems Science*, **23**, 789-798.
- (13) Charnes, A.; Rousseau, J. & Semple, J. (1993). An effective non-Archimedean anti-degeneracy/ cycling linear programming method especially for data envelopment analysis and like methods. *Annals of Operations Research*, **47**, 271-278.
- (14) Debreu, G. (1951). The Coefficient of Resource Allocation. *Econometrica*, **19**, 273-292.
- (15) Deprins, D.; Simar, L. & Tulkens, H. (1984). Measuring labour-efficiency in post offices. **In:** *The Performance of Public Enterprises* [edited by M. Marchand, P. Piestieau, and H. Tulkens], North-Holland, Amsterdam.
- (16) Dulá, J.H.; Helgason, R.V. & Hickman, B.L. (1992). Preprocessing schemes and a solution method for the convex hull problem in multidimensional space. **In:** *Computer Science and Operations Research New Developments in their Interfaces* [edited by O. Balci], Pergamon Press, Oxford, England, 59-70.
- (17) Dulá, J.H. & Venugopal, N. (1995). On characterizing the production possibility set for the CCR ratio model in DEA. *International Journal of Systems Science*, **26**, 2319-2325.
- (18) Dulá, J.H. & Helgason, R.V. (1996). A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space. *European Journal of Operational Research*, **92**, 352-367.
- (19) Dulá, J.H. (1997). Equivalences between Data Envelopment Analysis and the theory of redundancy in linear systems. *European Journal of Operational Research*, **101**, 51-64.
- (20) Dulá, J.H. & Hickman, B.L. (1997). Effects of excluding the column being scored from the DEA envelopment LP technology matrix. *Journal of the Operational Research Society*, **48**, 1001-1012.
- (21) Dulá, J.H.; Helgason, R.V. & Venugopal, N. (1998). An algorithm for identifying the frame of a pointed finite conical hull. *INFORMS Journal on Computing*, **10**, 323-330.
- (22) Dulá, J.H. & Thrall, R.M. (2001). A computational framework for accelerating DEA. *Journal of Productivity Analysis*, **16**, 63-78.
- (23) Dulá, J.H. & López, F.J. (2001b). Detecting the impact of including and omitting an attribute in DEA. Research Report HCES-05-01, Hearin Center for Enterprise Science, The University of Mississippi, University, MS 38677.
- (24) Durchholz, M.L. (1994). Large-scale Data Envelopment Analysis Models and Related Applications. Ph.D. Dissertation, Southern Methodist University, Dallas, TX 75275.

-
- (25) Farrell, M.J. (1957). The measurement of productive efficiency. *Journal of the Royal Statistical Society*, **120**, 253-290.
- (26) Koopmans, T. (1951). Analysis of production as an efficient combination of activities. **In:** *Activity Analysis of Production and Allocation* [edited by T.C. Koopmans], John Wiley & Sons, Inc., New York.
- (27) López, F.J. (1990). Algorithms to Obtain the Frame of a Finitely Generated Unbounded Polyhedron. Ph.D. Dissertation, The University of Mississippi, University, MS 38677.
- (28) Olesen, O.B. & Petersen, N.C. (2001). Identification and use of efficient faces and facets in DEA. To appear in *Journal of Productivity Analysis*.
- (29) Pastor, J.T.; Ruiz, J.L. & Sirvent, I. (1999). An enhanced DEA Russell graph efficiency Measure. *European Journal of Operational Research*, **115**, 596-607.
- (30) Rockafellar, R.T. (1970). *Convex Analysis*. Princeton University Press, Princeton, New Jersey.
- (31) Rosen, J.B.; Xue, G.L. & Phillips, A.T. (1992). Efficient computation of extreme points of convex hulls in \mathbb{R}^d . **In:** *Advances in Optimization and Parallel Computing* [edited by P.M. Pardalos], North Holland, 267-292.
- (32) Seiford, L.M. (1996). Data Envelopment Analysis: The evolution of the state of the art (1978-1995). *The Journal of Productivity Analysis*, **7**, 99-137.
- (33) Seiford, L. & Zhu, J. (1999). Infeasibility of Super-Efficiency Data Envelopment Analysis Models. *INFOR*, **37**, 174-187.
- (34) Shaheen, M. (2001). A Pre-Processor for the CCR Model in DEA. Presented at the INFORMS National Conference, Nov. 6, 2001. Miami, FL.
- (35) Tone, K. (2001). A slacks-based measure of efficiency in data envelopment analysis. *European Journal of Operational Research*, **130**, 498-509.
- (36) Thrall, R.M. (1996). Duality, classification and slacks in DEA. *Annals of Operations Research*, **66**, 109-138.
- (37) Wets, R.J.B. & Witzgall, C. (1967). Algorithms for frames and linearity spaces of cones. *Journal of Research of the National Bureau of Standards – B Mathematics and Mathematical Physics*, **71B**, 1-7.
- (38) Wets, R.J.-B. (1990). Elementary, constructive proofs of the theorems of Farkas, Minkowski and Weyl. **In:** *Economic Decision-Making: Games, Econometrics and Optimization* [edited by J.J. Gabszewicz, J.-F. Richard, and L.A. Wolsey].
- (39) Yu, G.; Wei, Q.; Brockett, P. & Zhou, L. (1996). Construction of all DEA efficient surfaces of the production possibility set under the generalized Data Envelopment Analysis model. *European Journal of Operational Research*, **95**, 491-510.