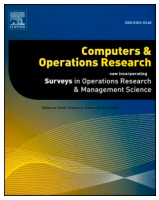
Computers & Operations Research 184 (2025) 107242

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor



Estimating production technologies using multi-output adaptive constrained enveloping splines

Victor J. España [a], Juan Aparicio [a,b,*], Xavier Barber [a]

[a] Center of Operations Research (CIO). Miguel Hernandez University of Elche. Avda. de la Universidad, s/n, 03202 Elche, Spain
[b] Senior Research Fellow. Valencian Graduate School and Research Network of Artificial Intelligence (valgrAI), Camino de Vera, s/n, 46022 Valencia, Spain

ARTICLE INFO

Keywords:
Data Envelopment Analysis
Multi-output technologies
Overfitting
Machine Learning

ABSTRACT

Data Envelopment Analysis (DEA) is a widely used method for evaluating the relative efficiency of decision-making units, but it often yields overly optimistic efficiency estimates, particularly with small sample sizes. To overcome this limitation, we introduce Adaptive Constrained Enveloping Splines (ACES), a non-parametric technique based on regression splines to accommodate multi-output, multi-input production contexts. ACES employs a three-stage estimation process. In the first stage, optimal output levels are estimated while incorporating essential envelope constraints, with optional monotonicity and/or concavity adjustments as needed. In the second stage, a refinement phase is carried out in which some of the estimates made are replaced by the observed values. Finally, a DEA-type technology is constructed using a new virtual data sample, ensuring adherence to usual shape constraints. Although ACES entails a higher computational cost, it achieves substantially lower mean squared error and bias than alternative methods of the literature across a wide range of simulated scenarios. This improvement is particularly pronounced in settings with complex production structures or heterogeneous returns to scale. This performance is consistent across both noise-free and noisy data environments, underscoring the method's robustness and accuracy.

1. Introduction

Efficiency analysis is the discipline focused on evaluating a set of observations, commonly referred as Decision-Making Units (DMUs), in terms of their transformation process from inputs to outputs. Understanding and improving efficiency is of paramount importance for organizations aiming to optimize their resource allocation and output generation in manufacturing, healthcare, finance, or any public sector.

Non-parametric frontier methods are widely used in measuring efficiency in production theory. These methodologies allow researchers to assess the performance of DMUs by comparing their actual output level against a frontier that represents the best achievable output for a given input profile. By capturing the gap between observed performance and the estimated frontier, these methods provide a valuable insight into the potential for improvement in areas where inefficiencies may be present. These types of frontiers are generally estimated through Data Envelopment Analysis (DEA) introduced by Charnes et al. (1978) and Banker et al. (1984), or Free Disposal Hull (FDH) proposed by Deprins et al. (1984). These non-parametric approaches present some benefits regarding their parametric counterparts, such as Stochastic Frontier Analysis (SFA), introduced by Aigner et al. (1977) and Meeusen and van Den Broeck (1977) . For example, they can estimate the frontier with greater flexibility, as they do not require assumptions about the functional form of the data, as well as they handle multi-input and multi-output scenarios without imposing prior weights on the dimensions considered.

Nevertheless, DEA (and FDH) has faced criticism due to its non-statistical nature, leading some authors to label it as a merely descriptive tool at a frontier level with limited inferential capabilities (Esteve et al., 2020; Tsionas, 2022; Valero-Carreras et al., 2022; and Molinos-Senante et al., 2023). Enveloping techniques such as DEA or FDH locate the efficient frontier as close as possible to the data sample, relying on the principle of minimal extrapolation. While these techniques accurately measure efficiency for a specific and known set of observations, they are prone to suffering from overfitting. In this context, Korostelev et al. (1995) demonstrated that when applying DEA to a finite sample of identically and independently distributed observations drawn from a Data Generation Process (DGP), the estimated frontier exhibits a downward bias relative to the true frontier underlying the DGP. The overfitting problem in DEA has a direct impact on the results, leading to

* Corresponding author.
E-mail address: j.aparicio@umh.es (J. Aparicio).

https://doi.org/10.1016/j.cor.2025.107242
Received 3 October 2024; Received in revised form 16 May 2025; Accepted 1 August 2025
Available online 6 August 2025

a significant portion of the evaluated DMUs being identified as technically efficient. This fact suggests that a DMU should not be regarded as (truly) efficient even if the DEA models indicate it as such. In general, DEA scores are particularly susceptible to overoptimism, which becomes even more pronounced when the dimensionality of the analysis increases. In fact, the curse of dimensionality is a common issue in DEA, occurring when the ratio of DMUs to variables (inputs and outputs) is not large enough. This phenomenon has been discussed in studies such as Adler and Golany (2007), Wilson (2018), Charles et al. (2019) and Chen et al. (2021).

To overcome these limitations in the non-parametric approach, particularly in estimating technical efficiency at an inferential level rather than a sample-specific based evaluation, several authors have striven over the past few decades to introduce alternative methodologies that complement (or replace) DEA. For instance, Simar and Wilson (1998, 2000) employed bootstrapping techniques to calculate confidence intervals for efficiency scores estimated through DEA. Aragon et al. (2005) proposed a non-parametric estimator of the efficient frontier based on conditional quantiles from a relevant production process distribution. Building upon these concepts, Daouia and Simar (2007) further expanded on this approach. Kuosmanen and Johnson (2010) introduced the Corrected Concave Nonparametric Least Squares (C^2NLS) regression as a reinterpretation of DEA, aiming to estimate the underlying theoretical production function that generated the observed data sample. Parmeter and Racine (2013) proposed smooth constrained nonparametric and semiparametric estimators for production frontiers while satisfying theoretical axioms of production theory. Finally, Daouia et al. (2016) presented a constrained estimation method for support frontiers combining edge estimation and quadratic or cubic spline smoothing techniques.

From a stochastic point of view, several authors have introduced methods incorporating both inefficiency and statistical noise into frontier estimation. An early contribution in this direction is the stochastic DEA formulation by Banker (1988), who introduced a linear programming-based approach to estimate a frontier that accounts explicitly for statistical noise, positioning it within the data rather than strictly enveloping it. This model was further extended by Banker and Maindiratta (1992) into a semiparametric framework, incorporating maximum likelihood estimation and specific distributional assumptions about inefficiency and noise. More recently, Kuosmanen and Kortelainen (2012) introduced the Stochastic Non-Smooth Envelopment of Data (StoNED) method, which combines the nonparametric DEA approach with the SFA framework, aiming to estimate production frontiers while accounting for both inefficiency and statistical noise. Finally, Kuosmanen et al. (2015) and Kuosmanen and Johnson (2017) developed a consistent nonparametric estimator of the Directional Distance Function (DDF) introduced by Chambers et al. (1998) using StoNED.

Additionally, the DEA community is increasingly exploring the relationship between efficiency analysis, production function estimation, and machine learning, particularly to address overfitting in traditional methods by improving the estimation of the Data Generating Process. For instance, Olesen and Ruggiero (2018) introduced weighted random hinge functions with parameter constraints as an alternative to Afriat-Diewert-Parkan (ADP) estimators. Esteve et al. (2020) developed Efficiency Analysis Trees (EAT) to estimate frontiers in a FDH fashion, using a modified version of the Classification and Regression Trees algorithm (Breiman et al., 1984). Building on these ideas, Esteve et al. (2023) and Guillen et al. (2023) further improved the robustness of the EAT results by incorporating adaptations of the Random Forest methodology (Breiman, 2001) and Gradient Tree Boosting (Friedman, 2001), respectively. Valero-Carreras et al. (2021) adapted Support Vector Regression (SVR), originally introduced by Drucker et al. (1997), for production function estimation, with a natural extension for the multi-output case presented in Valero-Carreras et al. (2022). In the same line, Guerrero et al. (2022) further extended SVR to estimate production frontiers, effectively mitigating the typical overfitting problem. Olesen and Ruggiero (2022) introduced Hinging Hyperplanes (HH) function approximation (Breiman, 1993) as a flexible estimator of production functions. Finally, España et al. (2024) adapted the additive version of Multivariate Adaptive Regression Splines (MARS), introduced by Friedman (1991), to standard production contexts with a single output through piece-wise linear functions.

Our approach relies on an adaptation of the additive version of the Multivariate Adaptive Regression Splines (MARS) algorithm by Friedman (1991). This technique is a powerful non-parametric method broadly used in supervised learning. MARS approximates a target function through an expansion of basis functions (BFs), which are mathematical transformation of variables (i.e., step functions, polynomials, splines, etc.). These BFs serve as the building blocks for approximating complex relationships and interactions between predictors. The MARS algorithm constructs the approximation by iteratively adding and removing BFs, which are defined as splines — piecewise linear functions that are connected at specific points called knots. These splines can be univariate or multivariate, depending on whether they involve one or multiple predictors. When MARS uses only univariate splines, it is referred to as an additive MARS model. The MARS technique faces the problem of optimal knot location using two main processes: forward selection and backward elimination. The forward selection creates a comprehensive set of BFs that may overfit the training data, while (afterwards) the backward elimination sequentially removes (unnecessary) BFs with minimal impact on the model's performance. MARS avoids the problem of data overfitting in this way.

Despite relying on different modeling strategies, the approaches by Olesen and Ruggiero (2022), España et al. (2024), and our current extension all follow a similar underlying principle: they build flexible piecewise linear approximations through input space partitioning and the combination of elementary BFs. However, the scope of each approach—and the type of technologies they are best suited to estimate—differs substantially. On the one hand, Olesen and Ruggiero (2022) reinterpreted Breiman's HH formulation as a nonparametric estimator of S-shaped production functions by assuming homotheticity. Their method uses fixed hinge locations and separates the estimation into a linear homogeneous core and a nonlinear scaling law, avoiding the need for Afriat-type inequalities (Afriat, 1972; Diewert and Parkan, 1983). On the other hand, España et al. (2024) proposed a constrained version of MARS which estimates concave production functions by selecting spline basis functions under shape constraints, through a fully data-driven forward–backward procedure. That work is now further extended to handle multi-output settings by constructing a DEA-type technology from refined predictions, enabling full compatibility with standard efficiency measurement tools. In future research, a similar separation into a linear core and nonlinear scaling law—following the structure proposed by Olesen and Ruggiero—could also be explored within our framework, potentially broadening its applicability to S-shaped technologies.

We now present our methodological contribution, which builds upon and extends the additive MARS approach proposed by España et al. (2024). Specifically, our proposed framework provides three key methodological advancements that address limitations of the original method and extend its applicability to a broader set of problems.

First, we relax the conditions for satisfying monotonicity and concavity assumptions. In the original approach, these constraints were enforced additively, meaning that each univariate function within the estimator was independently required to comply with the shape restrictions. In contrast, our new approach enforces these constraints dimensionally, applying them across the (dimensional) aggregated function rather than its individual components. This refinement eliminates the need for all component functions to be individually monotonic and concave, thereby providing greater flexibility in estimating the model's coefficients.

Second, we address a key limitation of the original additive framework: its inability to model interactions between variables. By

introducing the capability to account for these interactions, our approach captures intricate, non-linear relationships present in the data, overcoming the restrictive linear structure of the original method. This improvement directly resolves one of the critical shortcomings identified in the earlier work, where cross-variable effects were insufficiently represented. Together with the relaxation of monotonicity and concavity constraints described earlier, this advancement considerably enhances the model's ability to accurately estimate the underlying data structure and account for the complexities of real-worlds production processes.

Third, we expand upon the work of España et al. (2024) by extending their research on production functions (i.e., only one output was considered) to estimate multi-output, multi-input technologies. This extension is based on a novel procedure that unfolds in three main steps. First, we use a Machine Learning (ML) technique adapted to the context of efficiency analysis to approximate observed outputs to the data-generating process (DGP), ensuring that the observed sample is enveloped from above. Second, a refinement phase is performed, where some of the initial estimates, particularly those deemed inaccurate, are replaced by their observed values. Third, a DEA-VRS (Variable Returns to Scale) technology is constructed by replacing the observed outputs with a new vector of outputs predicted by the ML algorithm. To achieve this, we introduce an extension of the algorithm originally proposed by España et al. (2024), which we name Adaptive Constrained Enveloping Splines (ACES) to distinguish it from its earlier version.

The paper's structure is as follows. Section 2 provides the background. In Section 3, we introduce the new technique called Adaptive Constrained Enveloping Splines (ACES), detailing its theoretical underpinnings and how it builds on previous methodologies. Section 4 describes how some well-known measures of technical efficiency can be implemented through ACES. Section 5 introduces the set of available hyperparameters and conducts computational experiments using simulated data to evaluate the performance of the new approach. Section 6 shows an empirical illustration. Finally, Section 7 concludes the paper.

2. Background

This section provides a brief overview of important concepts related to Data Envelopment Analysis and the application of Multivariate Adaptive Regression Splines for estimating production functions (España et al., 2024). Additionally, we will introduce some notation.

2.1. Data envelopment analysis

Let us consider a sample of n DMUs, whose technical efficiency needs to be evaluated. Specifically, each DMU_i, $i = 1,\ldots,n$, consumes $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{im}) \in R_+^m$ inputs to produce $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ir}, \ldots, y_{is}) \in R_+^s$ outputs. To assess the (relative) efficiency of a DMU, a common technology set (φ), shared by all the DMUs within the sample, needs to be defined. From a broader viewpoint, the technology can be expressed as:

$$\varphi = \left\{(\boldsymbol{x},\boldsymbol{y}) \in R_+^{m+s} : \boldsymbol{x} \text{ can produce } \boldsymbol{y}\right\}. \tag{1}$$

This technology includes all $(\boldsymbol{x},\boldsymbol{y})$ combinations that are technically feasible. In the non-parametric approach, this technology is axiomatically established, adhering to principles outlined by Banker et al. (1984). Precisely, it upholds the free disposability of inputs and outputs, meaning that if $(\boldsymbol{x},\boldsymbol{y}) \in \varphi$, then $(\boldsymbol{x}',\boldsymbol{y}') \in \varphi$ for $\boldsymbol{x}' \geqslant \boldsymbol{x}$ and $\boldsymbol{y}' \leqslant \boldsymbol{y}$. It also guarantees the enveloping property, ensuring that $(\boldsymbol{x}_i,\boldsymbol{y}_i) \in \varphi, \forall i = 1,\ldots, n$. Convexity is also typically assumed, implying that if $(\boldsymbol{x},\boldsymbol{y}) \in \varphi$ and $(\boldsymbol{x}',\boldsymbol{y}') \in \varphi$, then $\lambda(\boldsymbol{x},\boldsymbol{y}) + (1-\lambda)(\boldsymbol{x}',\boldsymbol{y}') \in \varphi$, $\forall \lambda \in [0,1]$. Lastly, the technology meets minimal extrapolation, representing the smallest set satisfying prior axioms. This particular axiom is the cause of the overfitting problem by closely approximating the technology's boundary to the observed units (see, e.g., Esteve et al., 2020).

In the realm of Data Envelopment Analysis (DEA), Banker et al. (1984) proposed an estimation of the technology with variable returns to scale as:

$$\widehat{\varphi}_{DEA} = \left\{ (\boldsymbol{x},\boldsymbol{y}) \in R_+^{m+s} : x_j \geqslant \sum_{i=1}^{n} \lambda_i x_{ij}, j = 1,\ldots,m, y_r \leqslant \sum_{i=1}^{n} \lambda_i y_{ir}, r = 1,\ldots,s, \sum_{i=1}^{n} \lambda_i = 1, \lambda_i \geqslant 0, i = 1,\ldots,n \right\}. \tag{2}$$

Regarding the measurement of technical efficiency, the DMU being evaluated should be projected onto a certain part of the border of the technology. This part of the technology is called the efficient frontier or production frontier of φ, which is defined as:

$$\partial(\varphi) = \{(\boldsymbol{x},\boldsymbol{y}) \in \varphi : \widehat{\boldsymbol{x}} < \boldsymbol{x}, \widehat{\boldsymbol{y}} > \boldsymbol{y} \Rightarrow (\widehat{\boldsymbol{x}},\widehat{\boldsymbol{y}}) \notin \varphi\}. \tag{3}$$

In DEA, the estimation of the technology and the measurement of technical efficiency are achieved in a single step through a linear programming (LP) model. Typical measures for determining technical efficiency include the input-oriented and output-oriented radial measures (Banker et al., 1984), the input-oriented and output-oriented non-radial Russell measures (Färe and Lovell, 1978; Färe et al., 1985), the Directional Distance Function (Chambers et al., 1998) or the additive models such as the Measure of Inefficiency Proportions (Cooper et al., 1999), the Range Adjusted Measure (Cooper et al., 1999), the Bounded Adjusted Measure (Cooper et al., 2011) or the Normalized Weighted Additive Model (Lovell and Pastor, 1995).

Finally, we present a graphical example to illustrate the overfitting problem inherent in DEA. In Fig. 1, we display a sample of DMUs for efficiency evaluation using a DEA-VRS frontier. Simultaneously, we observe the underlying DGP, which measures the maximum output (y) achievable based on a given resource profile (x). It is interesting to note that DEA yields overoptimistic efficiency scores, potentially skewing efficiency assessments. We delve into the specific case of units A and B within this analysis. While DMU A is considered efficient, DMU B requires a slight increase in the level of the output produced, while maintaining a constant input level to achieve efficiency.[1] Nevertheless, both DMUs are significantly distant from the theoretical levels of efficiency, demonstrating the fact that DEA is solely based on sample-level assessments.

2.2. Multivariate Adaptive Regression splines for the estimation of production functions

In this section, we briefly introduce the main notion associated with the model by España et al. (2024) to estimate single-output multi-input

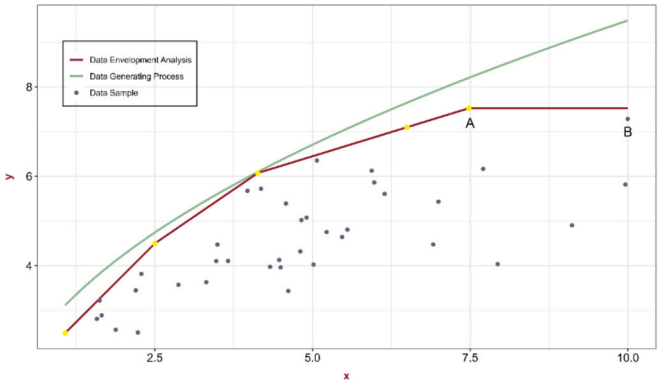


Fig. 1. An example of overoptimistic DEA scores.

[1] Under the output-oriented radial measure approach (Banker et al., 1984).

production contexts, that is, a production function. Estimating production functions can be considered a shape-restricted regression problem that focuses on unveiling the relationship between a single-output y and a set of inputs x_j, $j = 1,...,m$, at its extreme level. In the deterministic setting, the data-generating system is presumed to be described by the following expression:

$$y = f(x_1,...,x_m) - v. \tag{4}$$

Here, the first term $f(x_1,...,x_m)$ represents the joint predictive relationship between the output y and the set of inputs $\boldsymbol{x} = (x_1,...,x_m)$, while the term $v \geqslant 0$ assesses the level of technical inefficiency, with $v = 0$ indicating technical efficiency.

España et al. (2024) extended the additive version of the Multivariate Adaptive Regression Splines (MARS) algorithm (Friedman, 1991) by incorporating shape constraints to approximate y under a deterministic framework. This modified approach estimates a surface that envelops the data from above while ensuring non-decreasing monotonicity and concavity. The additive version of MARS is constructed through the linear combination of a set of basis functions (BFs). The set of BFs are created from an exhaustive search of knot locations by recursive partitioning of the input space. Friedman (1991) proposes the implementation of a strategy based on selecting two-sided truncated univariate splines of degree 1 as BFs:

$$\begin{aligned} B^{+}(\boldsymbol{x}) &= (x_j - k)_{+} = \max(0, x_j - k) \\ B^{-}(\boldsymbol{x}) &= (k - x_j)_{+} = \max(0, k - x_j). \end{aligned} \tag{5}$$

A reflected pair consists of two BFs that share a common knot location (k): a right-side spline that captures the relationship between the predictor x_j and the response variable to the right side of the knot, and a left-side spline that does so to the left side of the knot. When a (sibling) spline is removed from a reflected pair, we refer to it as an unpaired BF. This terminology allows us to distinguish between splines that remain part of a reflected pair and those that become unpaired during the model-fitting process.

The MARS algorithm involves two stepwise procedures: a forward selection and a backward elimination. In the forward selection step, the input space is divided into subspaces by searching for knots along the range of inputs. These knots are used to create a set of BFs through splines, which transform the original inputs into additional data. At each forward step, the reflected pair that minimizes the training error the most, is added as a new term in the model. The set of BFs for creating reflected pairs during the forward step is the following:

$$\aleph = \left\{ \left\{ (x_j - k)_{+}, (k - x_j)_{+} \right\} | k \in \{x_{1j}, x_{2j}, ..., x_{nj}\}, j = 1, ..., m \right\}. \tag{6}$$

This step continues until the desired number of BFs (univariate splines) predefined by the user is reached or when further error reduction is not significant. The backward elimination algorithm is then applied, sequentially removing less significant model terms. By combining these stepwise procedures, the MARS model achieves an effective balance between model complexity and predictive performance.

The new additive MARS approximation function within the production framework is formulated as:

$$\begin{aligned} \widehat{f}(\boldsymbol{x}) = \tau_0 + \sum_{j=1}^{m} \sum_{p \in P_j} \Bigg[\gamma_{j_p}^{+} \left(x_j - \widetilde{\kappa}_{j_p} \right)_{+} \\ + \gamma_{j_p}^{-} \left(\widetilde{\kappa}_{j_p} - x_j \right)_{+} \Bigg] + \sum_{j=1}^{m} \sum_{v \in V_j} \left[\omega_{j_v} \left(\widetilde{\kappa}_{j_v}^{(L)} - x_j \right)_{+} \right], \end{aligned} \tag{7}$$

where x_j is the $j-$th input, $j = 1,...,m$, τ_0 is the intercept term, $\gamma_{j_p}^{+}$ and $\gamma_{j_p}^{-}$ are the coefficients associated with the $p-$th reflected pair for the $j-$th input and $\widetilde{\kappa}_{j_p} \in \{x_{1j}, x_{2j}, ..., x_{nj}\}$ is the knot location that defines the $p-$th reflected pair for the $j-$th input. Furthermore, $P = \{P_j\}_{j=1}^{m}$ is a set of m elements, where P_j is the subset of indexes that enumerate the reflected pairs built through the $j-$th input. Then, γ_j^{+}, γ_j^{-} and $\widetilde{\kappa}_j$, $j = 1,...,m$, are the $j-$th subset of γ^{+}, γ^{-} and $\widetilde{\kappa}$, respectively. While the first two subsets hold the coefficients, the latter subset is made up of the knot locations in the input space for the reflected pairs. In the same way, we can define V, ω and $\widetilde{\kappa}^{(L)}$ for the left-side unpaired BFs. Finally, we define the set of selected BFs as $B = \left\{ 1, \left(x_j - \widetilde{\kappa}_{j_p} \right)_{+}, \left(\widetilde{\kappa}_{j_p} - x_j \right)_{+}, \left(\widetilde{\kappa}_{j_v}^{(L)} - x_j \right)_{+} \right\}, j = 1, ..., m, \forall p \in P_j, \forall v \in V_j$. Notice that during the forward algorithm, all the BF are paired, resulting in $V = \varnothing$.

The algorithm starts by incorporating the constant function $B_1(\boldsymbol{x}) = 1$ (τ_0) into the model to establish the initial region over the entire domain. Next, a new reflected pair from (6) is selected to be incorporated into the model as:

$$\widehat{f}(\boldsymbol{x}) = \tau_0 + \gamma_{j_1}^{+} \left(x_j - \widetilde{\kappa}_{j_1} \right)_{+} + \gamma_{j_1}^{-} \left(\widetilde{\kappa}_{j_1} - x_j \right)_{+}. \tag{8}$$

The fitting process involves generating a model for each possible (and available) combination of variable x_j and knot location $\widetilde{\kappa}_{j_{|P_j|+1}} = x_{ij}$ (an observed value), where $|P_j|$ denotes the cardinality of P_j. This procedure is computationally expensive, as nearly $n{\cdot}m$ models are fitted in each iteration.[2] From each of those models, a set of coefficients is estimated via the following LP model:

$$\begin{aligned} &\underset{\boldsymbol{\varepsilon}, \tau_0, \gamma^{+}, \gamma^{-}}{\text{minimize}} \sum_{i=1}^{n} \varepsilon_i \\ &\text{subject to} \\ &\tau_0 + \sum_{j=1}^{m} \sum_{p \in P_j} \left[\gamma_{j_p}^{+} \left(x_{ij} - \widetilde{\kappa}_{j_p} \right)_{+} + \gamma_{j_p}^{-} \left(\widetilde{\kappa}_{j_p} - x_{ij} \right)_{+} \right] - \varepsilon_i = y_i, \quad \forall i, \quad (9.1) \\ &\varepsilon_i \geqslant 0, \quad \forall i, \quad (9.2) \\ &-\gamma_{j_p}^{+} - \gamma_{j_p}^{-} \geqslant 0, \quad \forall j, \forall p \in P_j, \quad (9.3) \\ &\gamma_{j_p}^{+} \geqslant 0, \quad \forall j, \forall p \in P_j, \quad (9.4) \\ &-\gamma_{j_p}^{-} \geqslant 0, \quad \forall j, \forall p \in P_j. \quad (9.5) \end{aligned} \tag{9}$$

In model (9), ε_i measures the error term defined by constraint (9.1). Note that this variable must be restricted to be positive to envelop the

[2] Friedman (1991) and Zhang (1994) introduced methods to preserve spacing between successive knots, aiming to reduce overfitting while also lowering computational cost by limiting the number of fitted models.

observed data, as indicated by constraint (9.2). Constraint (9.3) ensures concavity, while constraints (9.4) and (9.5) guarantee the non-decreasing monotonicity of the estimator. Then, the reflected pair from (6) that yields the greatest reduction in a certain lack-of-fit (LOF) criterion when solving (9), is introduced in the model. Typically, the mean residual sum of squares is chosen as a LOF measure:

$$LOF = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}(\boldsymbol{x}_i))^2. \tag{10}$$

In this way, equation (8) expands iteratively until a stopping criterion holds: first, when the maximum number of BFs (η) has been reached, and second, when the reduction in training error becomes insignificant, as determined by a user-defined ratio (ξ) . Like DEA, the model associated with the forward step still exhibits overfitting. At this stage, the accuracy is relatively high due to the close approximation to the data sample by the piece-wise linear estimator, resulting in low bias. However, this estimator relies heavily on the training data, leading to high variance and makes the achievement of a good generalization performance somewhat challenging. To tackle this issue, a backward algorithm based on the generalized cross-validation (GCV) metric, initially proposed by Golub et al. (1979), is employed:

$$GCV(B) = \frac{\frac{1}{n}\sum_{i=1}^{n}[y_i - \widehat{f}_B(\boldsymbol{x}_i)]^2}{\left[1 - \frac{C(B)+d\cdot\chi}{n}\right]^2}, \tag{11}$$

In this context, $C(B)$ represents the number of parameters to be estimated in $\widehat{f}_B$, where $\widehat{f}_B$ is a model built from a specific set of BFs (B). The hyperparameter d penalizes model complexity. Finally, χ is the number of additional BF parameters being fit to the data in $\widehat{f}_B$. In the new ACES algorithm, χ represents the number of knots being placed to establish the set of BFs. It should be noted that removing a BF from a reflected pair reduces $C(B)$ by one unit since there is one less parameter to be fitted, but it does not affect χ.

The backward algorithm shown in España et al. (2024) follows the procedure outlined in Friedman (1991). The key difference between both approaches lies in the selection procedure for removing a BF. In contrast to standard (additive) MARS, where any BF is candidate for elimination, the approached described in España et al. (2024) introduces two conditions that must be considered before selecting a BF for removal:

1. Right-side BFs can only be removed from reflected pairs.
2. Left-side BFs can only be removed when appearing unpaired.

Then, the LP model to be solved at each stage of the backward algorithm is as follows:

$$\underset{\boldsymbol{\varepsilon},\tau_0,\gamma^+,\gamma^-,\boldsymbol{\omega}}{\text{minimize}} \sum_{i=1}^{n}\varepsilon_i$$

subject to

$$\begin{aligned}
&\tau_0 + \sum_{j=1}^{m}\sum_{p\in P_j}\left[\gamma_{j_p}^{+}\left(x_{ij} - \widetilde{\kappa}_{j_p}\right)_+ + \gamma_{j_p}^{-}\left(\widetilde{\kappa}_{j_p} - x_{ij}\right)_+\right] + \sum_{j=1}^{m}\sum_{v\in V_j}\left[\omega_{j_v}\left(\widetilde{\kappa}_{j_v}^{(L)} - x_{ij}\right)_+\right] - \varepsilon_i = y_i, \quad \forall i, && (12.1)\\
&\varepsilon_i \geqslant 0, \quad \forall i, && (12.2)\\
&-\gamma_{j_p}^{+} - \gamma_{j_p}^{-} \geqslant 0, \quad \forall j, \forall p \in P_j, && (12.3)\\
&\gamma_{j_p}^{+} \geqslant 0, \quad \forall j, \forall p \in P_j, && (12.4)\\
&-\gamma_{j_p}^{-} \geqslant 0, \quad \forall j, \forall p \in P_j, && (12.5)\\
&-\omega_{j_v} \geqslant 0, \quad \forall j, \forall v \in V_j. && (12.6)
\end{aligned} \tag{12}$$

Model (12) introduces only one additional constraint, (12.6), compared to the forward model (9), ensuring that the unpaired BFs on the left side also adheres to monotonicity and concavity properties. Hence, the backward algorithm creates a set of $|B|-1$ sub-models by removing BFs one by one and selects the model that minimizes (11).

3. Adaptive Constrained Enveloping splines

This section details the enhancements made to the additive MARS model introduced by España et al. (2024). First, we outline the modifications implemented to improve the model fit, including (i) the introduction of a relaxed optimization problem for coefficient estimation and (ii) the incorporation of variable interactions in the modeling process. Second, we describe how the model has been adapted for estimating production frontiers, which involve the analysis of multi-output multi-input production contexts.

3.1. A new and more relaxed approach

The additive MARS model adapted to the estimation of production functions presented in España et al. (2024) is based on the idea that the sum of non-decreasing monotonic and concave functions results in a function that is both non-decreasing monotonic and concave. Specifically, each reflected pair and unpaired left-side BF in (7) are forced to be non-decreasing monotonic and concave. However, it is worth noting that these constraints may be overly stringent.

We propose the following reformulation of (7), where the additive MARS model is expressed as a sum of (paired and unpaired) right-side and left-side BFs:

$$\widehat{f}^{ACES}(\boldsymbol{x}) = \tau_0 + \sum_{j=1}^{m}\sum_{h\in H_j}\left[\alpha_{j_h}\left(x_j - \kappa_{j_h}^{(R)}\right)_+\right] + \sum_{j=1}^{m}\sum_{u\in U_j}\left[\beta_{j_u}\left(\kappa_{j_u}^{(L)} - x_j\right)_+\right]. \tag{13}$$

Here, x_j is the $j-$th input, $j = 1,\ldots,m$, τ_0 is the intercept term, α_{j_h} is the coefficient associated with the $h-$th right-side BF for the $j-$th input and $\kappa_{j_h}^{(R)} \in \{x_{1j}, x_{2j}, \ldots, x_{nj}\}$ is the knot location that defines the $h-$th right-side BF for the $j-$th input. In addition to that, $H = \{H_j\}_{j=1}^{m}$ is a set of m elements, where H_j is the subset of indexes that enumerate the right-side BFs built through the $j-$th input. The standard format for BF enumeration is to list the paired BFs first, followed by the unpaired BFs (first right, then left). Besides, α_j and $\kappa_j^{(R)}$, $j = 1,\ldots,m$, are the $j-$th subset of α and $\kappa^{(R)}$, respectively. In the same way, we can define U, β and $\kappa^{(L)}$ for the left-side BFs. Within each type of BF, the order of enumeration is determined by the value of the knot location, from lowest to highest. Finally, remember that $\left(x_j - \kappa_{j_h}^{(R)}\right)_+$ and $\left(\kappa_{j_u}^{(L)} - x_j\right)_+$ form a reflected pair if $\exists h \in H_j$, $u \in U_j$ such that $\kappa_{j_h}^{(R)} = \kappa_{j_u}^{(L)}$. In particular, during the

forward algorithm, only reflected pairs are formed implying that $\kappa_{j_h}^{(R)} = \kappa_{j_u}^{(L)}$, $h = u, \forall h \in H_j, \forall u \in U_j$.

Next, the set of knots selected by the left-side BFs for the $j-$th input is specified as $\kappa_j^{(L)*} = \kappa_j^{(L)}$ such that $\kappa_{j_u}^{(L)} \in \kappa_j^{(L)}, \forall u \in U_j$. Besides, the set of selected knots associated with the right-side BFs for the $j-$th input is determined, excluding the knots already used by the left-side BFs, as $\kappa_j^{(R)*} = \kappa_j^{(R)} \setminus \left(\kappa_j^{(R)} \cap \kappa_j^{(L)}\right)$ such that $\kappa_{j_h}^{(R)} \in \kappa_j^{(R)}, \forall h \in H_j$. This omission of knots is necessary to avoid repeated values when the BFs form a reflected pair. Thus, the set of knots selected at any step of the algorithm is established as follows:

$$K = \left\{K_j : \kappa_j^{(R)*} \cup \kappa_j^{(L)*}\right\}_{j=1}^{m} = \{K_1, ..., K_m\} = \left\{\left\{k_{1_1}, ..., k_{1_{|K_1|}}\right\}, ..., \left\{k_{1_m}, ..., k_{1_{|K_m|}}\right\}\right\}, k_{j_1} < ... < k_{j_{|K_j|}}. \tag{14}$$

As an additional step, each set K_j is expanded by including both the minimum and maximum observed values of the variable x_j, with the goal of creating a collection of intervals $\left[k_{j_{t-1}}, k_{j_t}\right), t = 1, ..., |K_j| + 1$:

$$K^* = \left\{k_{j_0} \cup K_j \cup k_{j_{|K_j|+1}}\right\}_{j=1}^{m}, k_{j_0} = \min_{1 \leqslant i \leqslant n}(x_{ij}), k_{j_{|K_j|+1}} = \max_{1 \leqslant i \leqslant n}(x_{ij}). \tag{15}$$

Under this new approach, our objective is to guarantee non-decreasing monotonicity and concavity within each interval $\left[k_{j_{t-1}}, k_{j_t}\right)$. To achieve this, we use the closed-form expression of the first-order partial derivatives determined from expression (13) to establish estimation conditions on the coefficients that satisfy the shape requirements of the estimator. The idea is to gather all BFs that involve identical inputs, as expressed below:

$$\begin{aligned}\widehat{f}^{ACES}(\boldsymbol{x}) &= \tau_0 + \sum_{j=1}^{m} \widehat{f}_j^{ACES}(\boldsymbol{x}) \\ &= \tau_0 + \sum_{j=1}^{m} \left(\sum_{h \in H_j} \left[\alpha_{j_h} \left(x_j - \kappa_{j_h}^{(R)}\right)_+\right] + \sum_{u \in U_j} \left[\beta_{j_u} \left(\kappa_{j_u}^{(L)} - x_j\right)_+\right] \right).\end{aligned} \tag{16}$$

Hence, the $j-$th first-order partial derivative of $\widehat{f}^{ACES}(\boldsymbol{x})$ is defined as a piece-wise function:

$$\frac{\partial \widehat{f}^{ACES}(\boldsymbol{x})}{\partial x_j} = \sum_{h \in H_j} \alpha_{j_h} \cdot I\left(x_j > \kappa_{j_h}^{(R)}\right) - \sum_{u \in U_j} \beta_{j_u} \cdot I\left(x_j < \kappa_{j_u}^{(L)}\right), \tag{17}$$

intervals to the right-side of the knot $\kappa_{j_h}^{(R)}$, while $\left(\kappa_{j_u}^{(L)} - x_j\right)_+$ is activated in all intervals to the left-side of the knot $\kappa_{j_u}^{(L)}$. In view of this, we can define the $j-$th partial derivative of $\widehat{f}^{ACES}(\boldsymbol{x})$ for the $t-$th interval $\left[k_{j_{t-1}}, k_{j_t}\right)$ as:

$$\left.\frac{\partial \widehat{f}^{ACES}(\boldsymbol{x})}{\partial x_j}\right|_t = \sum_{h \in H_j} \alpha_{j_h} \cdot I\left(k_{j_0} \leqslant \kappa_{j_h}^{(R)} < k_{j_t}\right) - \sum_{u \in U_j} \beta_{j_u} \cdot I\left(k_{j_{t-1}} < \kappa_{j_u}^{(L)} \leqslant k_{j_{|K_j|+1}}\right), t = 1, ..., |K_j| + 1. \tag{18}$$

From this point onwards, we proceed to define the conditions required for ensuring both non-decreasing monotonicity and concavity properties of our estimator. Non-decreasing monotonicity is achieved by imposing that the estimated function increases in each interval $\left[k_{j_{t-1}}, k_{j_t}\right)$:

$$\left.\frac{\partial \widehat{f}^{ACES}(\boldsymbol{x})}{\partial x_j}\right|_t \geqslant 0, j = 1, ..., m, t = 1, ..., |K_j| + 1, \tag{19}$$

while concavity is guaranteed by imposing that the rate of growth decreases between two consecutive intervals $\left(\left[k_{j_{t-1}}, k_{j_t}\right), \left[k_{j_t}, k_{j_{t+1}}\right)\right)$:

$$\left.\frac{\partial \widehat{f}^{ACES}(\boldsymbol{x})}{\partial x_j}\right|_t \geqslant \left.\frac{\partial \widehat{f}^{ACES}(\boldsymbol{x})}{\partial x_j}\right|_{t+1}, j = 1, ..., m, t = 1, ..., |K_j| + 1, \tag{20}$$

which is equivalent to:

$$-\sum_{h \in H_j} \alpha_{j_h} \cdot I\left(k_{j_t} \leqslant \kappa_{j_h}^{(R)} < k_{j_{t+1}}\right) + \sum_{u \in U_j} \beta_{j_u} \cdot I\left(k_{j_{t-1}} < \kappa_{j_u}^{(L)} \leqslant k_{j_t}\right) \geqslant 0, j = 1, ..., m, t = 1, ..., |K_j|. \tag{21}$$

With this approach, the shape constraints are satisfied for each dimension individually, meaning that each $\widehat{f}_j^{ACES}(\boldsymbol{x})$ in (16) is a non-decreasing monotonic and concave function. This contrasts with the methodology introduced by España et al. (2024) where each reflected pair or left-side unpaired BF in (7) had to meet shape constraints. Then, the LP model for estimating the set of coefficients under this new approach is as follows:

$$\begin{aligned}
&\underset{\boldsymbol{\varepsilon}, \tau_0, \boldsymbol{\alpha}\boldsymbol{\beta}}{\text{minimize}} \sum_{i=1}^{n} \frac{1}{\phi_i} \cdot \varepsilon_i \\
&\text{subject to} \\
&\tau_0 + \sum_{j=1}^{m} \sum_{h \in H_j} \left[\alpha_{j_h} \left(x_{ij} - \kappa_{j_h}^{(R)}\right)_+\right] + \sum_{j=1}^{m} \sum_{u \in U_j} \left[\beta_{j_u} \left(\kappa_{j_u}^{(L)} - x_{ij}\right)_+\right] - \varepsilon_i = y_i, \quad \forall i, \quad (22.1) \\
&\varepsilon_i \geqslant 0, \quad \forall i, \quad (22.2) \\
&\sum_{h \in H_j} \alpha_{j_h} \cdot I\left(k_{j_0} \leqslant \kappa_{j_h}^{(R)} < k_{j_t}\right) - \sum_{u \in U_j} \beta_{j_u} \cdot I\left(k_{j_{t-1}} < \kappa_{j_u}^{(L)} \leqslant k_{j_{|K_j|+1}}\right) \geqslant 0, \quad \forall j, t = 1, ..., |K_j| + 1, \quad (22.3) \\
&-\sum_{h \in H_j} \alpha_{j_h} \cdot I\left(k_{j_t} \leqslant \kappa_{j_h}^{(R)} < k_{j_{t+1}}\right) + \sum_{u \in U_j} \beta_{j_u} \cdot I\left(k_{j_{t-1}} < \kappa_{j_u}^{(L)} \leqslant k_{j_t}\right) \geqslant 0, \quad \forall j, t = 1, ..., |K_j|. \quad (22.4)
\end{aligned} \tag{22}$$

where $I(.)$ is an indicator function that takes the value of 1 when its condition is met and 0 otherwise. These indicator functions determine the regions in the input space where a BF associated with some coefficient α_{j_h} or β_{j_u} is activated. Specifically, $\left(x_j - \kappa_{j_h}^{(R)}\right)_+$ is activated in all

In model (22), $\frac{1}{\phi_i}$ is included to weigh errors, where ϕ_i is the score of the $i-$th DMU obtained by the radial model under output orientation (Banker et al., 1984). This gives higher importance to errors near the frontier. Moreover, the relaxation of the problem also has an impact on

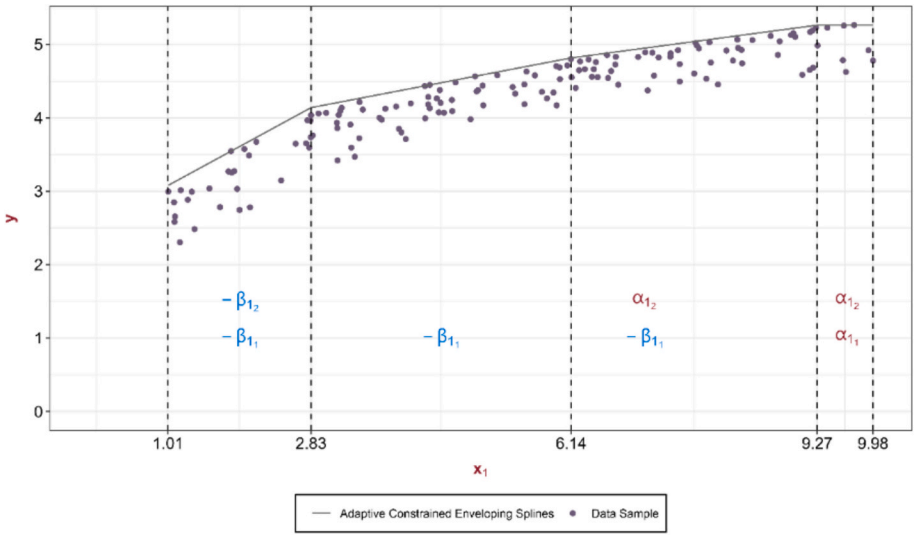


Fig. 2. Example of the partitioning of the input space for an ACES model.

the backward algorithm. Unlike the approach by España et al. (2024), which requires a careful selection of candidate BFs for removal at each iteration, the current approach allows any BF to be considered for elimination as long as (22) remains feasible.

Next, we provide an example to illustrate the aforementioned concepts using the following ACES model:

$$\widehat{f}^{ACES}(\boldsymbol{x}) = 5.4 + \alpha_{1_1}\cdot(x_1 - 9.27)_+ + \beta_{1_1}\cdot(9.27 - x_1)_+ + \alpha_{1_2}\cdot(x_1 - 6.14)_+ + \beta_{1_2}\cdot(2.83 - x_1)_+, \tag{23}$$

where $\alpha_{1_1} = 0.134$, $\beta_{1_1} = -0.192$, $\alpha_{1_2} = -0.055$ and $\beta_{1_2} = -0.417$.

Fig. 2 represents the partition of the input space into a collection of disjoint intervals derived from the ACES model defined in (23). The selected intervals arise naturally from the knot locations, which are determined by the optimization process in the ACES model. These knots are introduced where the data shows significant shifts in behavior, allowing the model to capture local variations and minimize error within each subregion. Specifically, the following intervals can be defined: $[1.01, 2.83)$, $[2.83,\ 6.14)$, $[6.14,\ 9.27)$ and $[9.27, 9.98)$. Moreover, $\left.\frac{\partial \widehat{f}^{ACES}(\boldsymbol{x})}{\partial x_1}\right|_t$ is formulated as the sum of the coefficients activated in each interval: 0.609 in $x_1 \in [1.01, 2.83)$, 0.192 in $x_1 \in [2.83, 6.14)$, 0.137 in $x_1 \in [6.14, 9.27)$ and 0.079 in $x_1 \in [9.27, 9.98)$. It can be observed that the first derivative takes a positive value in each interval and decreases as we move dimensionally to the right intervals. Finally, in terms of expression (15), the set of selected knots (including the minimum and the maximum data value) is $K^* = \{1.01,\ 2.83, 6.14,\ 9.27,\ 9.98\}$.

3.2. A model with interaction of variables

The prior version of ACES (Section 3.1) is a purely additive model, which may limit its performance when dealing with production functions involving interactions among input variables. Although its additive nature could initially be seen as a limitation, España et al. (2024) demonstrated through simulations from Kuosmanen and Johnson (2010) that the new method can outperform DEA and C^2NLS even in non-additive cases. However, the success of this approach depends on the predominance of variable interactions and the extent of non-additivity in the true production function. Specifically, España et al. (2024) highlight a threshold around 0.5 as the maximum interaction between variable for which the technique proposed continues to provide good results. Beyond this threshold, performance declines sharply, regardless of the sample size.

The standard MARS approach incorporates multivariate BFs of $q-$th degree. These high-order degree BFs are formed by taking the product of a new 1-degree (univariate) spline as in (5) and a previously entered (univariate or multivariate) BF. Some considerations must be made. Firstly, each factor in the multivariate BF must involve a different predictor variable to avoid high-power dependencies sensitive to extreme values. Secondly, a hierarchical structure prevails, as a $q-$th degree BF cannot be introduced in the model until a $(q-1)-$th degree BF is included. Lastly, these BFs act locally within the region of the input space where all the splines that comprise the multivariate BF are activated. As an example, consider the following 3-degree BF $(x_1 - 5)_+\cdot(x_3 - 9)_+\cdot(7 - x_2)_+$. This BF incorporates interactions between three predictor variables. The terms $(x_1 - 5)_+$, $(x_3 - 9)_+$ and $(7 - x_2)_+$ are the univariate splines associated with each predictor variable, and their product builds the multivariate BFs. Unfortunately, the product of a non-decreasing monotonic and concave function is not necessarily a non-decreasing monotonic and concave function. Then, as far as we are concerned, this strategy cannot be applied in our context due to the inability to impose the required shape constraints in these types of surfaces. In this regard, Shih et al. (2006) and Martinez et al. (2015) introduced two alternative convex (but non-monotonic) versions of MARS that allowed convex multivariate BFs.

To overcome the aforementioned limitations, we propose a simple yet effective procedure for incorporating variable interactions. We introduce a new hyperparameter, denoted as $q_{\max}$, that determines the maximum number of inputs that can interact to create a BF. Then, let $M = \{1, \ldots, m\}$ be the set of available inputs, and let $S =$

$\{M' \subseteq M : 2 < |M'| \leqslant q \leqslant m\}, S_g \in S, g = 1, ..., J, J = \sum_{a=2}^{q_{\max}} C_m^a = \binom{2}{m} + ... + \binom{q}{m}$ be the set of all possible interactions between the original inputs with a maximum degree $q_{\max}$. In this way, we can define J new additional variables as:

$$z_g = \prod_{j \in S_g} x_j, g = 1, ..., J. \tag{24}$$

Then, let $x_1, ..., x_m$ be the set of original input in R^m and let $z_1, ..., z_J$ be the set of interaction variables in R^J we can define the following transformation $\vartheta : R^m \rightarrow R^{m+J}$ on the form:

$$\vartheta(x_1, ..., x_m) = (x_1, ..., x_m, z_1, ..., z_J), \tag{25}$$

where each z_g is defined as the product of a subset of the original input variables x_j, with the subset containing at most $q_{\max}$ variables.

The introduction of this new set of artificial variables enables the modeling of interaction between variables while imposing necessary shape constraints, since these new variables are treated like the original inputs. Similar to the original MARS approach, interactions involving a variable with itself are avoided. On the other hand, these BFs will act locally but as a univariate BF, i.e., it will be activated on one side of the knot and deactivated on the other side. Additionally, while the hierarchical sense of interaction is not incorporated in exactly the same way as Friedman's method, the ACES algorithm does prioritize the inclusion of 1-degree BFs over higher degree BFs, following the approach shown in Tsai and Chen (2005). In this regard, we introduce a new hyperparameter $\xi^{(q)}$, $q = 2, ..., q_{\max}$, which determines the minimum percentage of improvement with respect to the best 1-degree BF required for the incorporation of a $q-$degree BF into the model. This ensures that only significantly beneficial higher-degree BFs are incorporated into the model.

Based on the outlined procedure, the dimension of the input space increases from m to $m + J$. Shape constraints are imposed in the new space of inputs, and, in consequence, the properties of non-decreasing monotonicity and concavity may not necessarily hold in the original space. Therefore, in a final step, a standard DEA-VRS technology is constructed using the new predicted output vector and the originally observed input vectors. In this way, the estimated frontier integrates information about variable interactions in the prediction of the optimal output, while satisfying all the original axioms in the m dimensional space.

3.3. A procedure for estimating a DEA-type technology using Adaptive Constrained Enveloping splines

In this section, we extend the previously described methodology, which estimates production functions under shape constraints like concavity and monotonicity, to accommodate a multi-output production framework. This extension defines an estimator for the production technology using a piecewise linear frontier, offering an alternative to traditional DEA in scenarios with multiple inputs and outputs, which are very usual in practice.

Consider a set of n DMUs, where each unit i is characterized by an observed input–output pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ randomly drawn from a true population. Our approach aims to identify optimal input–output combinations $(\boldsymbol{x}_i^*, \boldsymbol{y}_i^*)$, $i = 1, ..., n$, which define the true efficient frontier within the production space. This is carried out in a 3-stage procedure.

In the first stage, the objective is to estimate the optimal outputs $(\boldsymbol{y}_i^*)$ by determining approximations $(\widehat{\boldsymbol{y}}_i)$ that better reflect the true production capabilities of each DMU. This stage operates under the assumption that the observed inputs are optimal, i.e., $\boldsymbol{x}_i = \boldsymbol{x}_i^*$, indicating that there is no inefficiency or random error in input measurements. This step generates a new dataset $\{(\boldsymbol{x}_i^*, \widehat{\boldsymbol{y}}_i)\}_{i=1}^n$, representing a projection of the observed data close to the true efficient frontier and better align with the underlying production process.

In the second stage of the method, we address potential overestimations in the predicted output values. Specifically, if any component of the estimated output vector is suspected to be overestimated, i.e., $\widehat{y}_{ij} > y_{ij}^*$ for some j, it can be corrected by replacing $\widehat{y}_{ij}$ with the corresponding observed value y_{ij}. This conservative approach is proposed because is sufficient to project a few units to the true frontier to achieve a good characterization of the technology. Additionally, any estimate that falls between the observed value y_{ij} and the true (but unknown) optimal value y_{ij}^* improve the DEA's performance under no random error. However, an overestimation of $\widehat{y}_{ij}$ beyond y_{ij}^* could degrade the accuracy of the method. Therefore, correcting overestimations by retaining the observed values maintain the reliability of the resulting technology.

As a final step in the methodology, we construct a standard DEA-VRS technology using the refined version of $\{(\boldsymbol{x}_i^*, \widehat{\boldsymbol{y}}_i)\}_{i=1}^n$. This stage offers two significant advantages. First, it provides the flexibility to relax monotonicity and concavity constraints during the initial stage if needed. It is important to note that the first step is primarily focused on adjusting the observed data to better align with the true underlying production technology, rather than directly estimating it. By initially relaxing these constraints, the model can capture data complexities more effectively in some scenarios, while in others, maintaining the constraints may yield more accurate results. However, even if all the form constraints were imposed in the first stage, the correction phase could break with the proper form of the technology. For these reasons, constraints are rigorously re-imposed in the final step when constructing the DEA-VRS technology, ensuring that the final estimated technology satisfies all necessary shape restrictions. Thus, the first two stages effectively "pushes" the observed data toward the true DGP ensuring $\boldsymbol{y}_i \leqslant \widehat{\boldsymbol{y}}_i$, $\forall i = 1, ..., n$, with the full enforcement of form constraints occurring in this last step. Finally, the second advantage lies in that, resorting to the DEA-VRS technology, it is easy to measure technical inefficiency through any measure previously established within the standard DEA framework (see Section 4).

We begin by adapting the iterative procedure from the original single-output to a multi-output context. While the formal application of MARS for estimating multiple response variables is limited, Milborrow (2014) describes a method for handling multiple outputs within a standard MARS model, which is implemented in the R earth package (Milborrow, 2023). In this approach, while the same set of BFs is applied across all models, the coefficients differ for each response variable. Next, during the backward algorithm, the usual procedure is followed, but with the minimization of the overall GCV score across all response variables.

The first two phases of this method are detailed below, while the final phase is discussed in Section 4. To extend model (22) to the multi-output context, we need to consider all outputs simultaneously. This optimization imposes shape constraints like concavity and monotonicity while minimizing deviations from observed data. The result is a piecewise linear frontier that accurately captures the underlying production technology. The following LP model is used to produce an estimator that

Table 1
Illustrative example of the performance of our methodology for four selected units.

Index	x_1	x_2	y_1	y_2	y_1^*	y_2^*	$\widehat{y}_1$	$\widehat{y}_2$	$\phi^{(1)}$	$\phi^{(2)}$	ϕ
4	39.96	15.47	161.77	459.22	189.98	539.29	600.71	572.42	3.59	1.03	1.03
22	32.34	40.57	667.48	811.30	675.93	821.57	765.65	880.22	1.15	1.00	1.00
23	38.30	14.85	446.22	190.38	563.14	240.27	558.46	529.54	1.22	2.36	1.22
49	17.41	12.26	143.98	173.14	233.85	281.22	255.62	222.19	1.97	1.72	1.42

preserves the essential properties of production technologies while mitigating overfitting:

$$\underset{\boldsymbol{\varepsilon}^{(r)},\tau_0^{(r)},\boldsymbol{\alpha}^{(r)},\boldsymbol{\beta}^{(r)}}{\text{minimize}} \sum_{r=1}^{s}\sum_{i=1}^{n}\frac{1}{\phi_i}\cdot\varepsilon_{ir}$$

subject to

$$\begin{aligned}
&\tau_0^{(r)} + \sum_{j=1}^{m}\sum_{h\in H_j}\left[\alpha_{j_h}^{(r)}\left(x_{ij}-\kappa_{j_h}^{(R)}\right)_+\right] + \sum_{j=1}^{m}\sum_{u\in U_j}\left[\beta_{j_u}^{(r)}\left(\kappa_{j_u}^{(L)}-x_{ij}\right)_+\right] - \varepsilon_{ir} = y_{ir}, && \forall r,\forall i, && (26.1)\\
&\varepsilon_{ir} \geqslant 0, && \forall r,\forall i, && (26.2)\\
&\sum_{h\in H_j}\alpha_{j_h}^{(r)}\cdot I\left(k_{j_0}\leqslant\kappa_{j_h}^{(R)}<k_{j_t}\right) - \sum_{u\in U_j}\beta_{j_u}^{(r)}\cdot I\left(k_{j_{t-1}}<\kappa_{j_u}^{(L)}\leqslant k_{j_{|K_j|+1}}\right) \geqslant 0, && \forall r,\forall j,t=1,\ldots,|K_j|+1, && (26.3)\\
&-\sum_{h\in H_j}\alpha_{j_h}^{(r)}\cdot I\left(k_{j_t}\leqslant\kappa_{j_h}^{(R)}<k_{j_{t+1}}\right) + \sum_{u\in U_j}\beta_{j_u}^{(r)}\cdot I\left(k_{j_{t-1}}<\kappa_{j_u}^{(L)}\leqslant k_{j_t}\right) \geqslant 0, && \forall r,\forall j,t=1,\ldots,|K_j|. && (26.4)
\end{aligned} \tag{26}$$

The following discussion addresses key aspects of model (26). The shape constraints — monotonicity and concavity — are efficiently managed through constraints (26.3) and (26.4). While the grid of knots in the input space is shared across all outputs, the coefficients are estimated separately for each output. Alternatively, it is possible to estimate each output individually by applying model (22) independently, focusing on a dataset comprising all inputs and a single output. However, this separate estimation approach would imply not being able to determine a common set of knots as well as an increased computational burden on the process.

To continue, we define the estimation of the $r-$th output using the following expression:

$$\widehat{f}^{ACES^{(r)}}(\boldsymbol{x}) = \tau_0^{(r)} + \sum_{j=1}^{m}\sum_{h\in H_j}\left[\alpha_{j_h}^{(r)}\left(x_{ij}-\kappa_{j_h}^{(R)}\right)_+\right] + \sum_{j=1}^{m}\sum_{u\in U_j}\left[\beta_{j_u}^{(r)}\left(\kappa_{j_u}^{(L)}-x_{ij}\right)_+\right], \tag{27}$$

where, using our vector notation, we have $\widehat{f}^{ACES}(\boldsymbol{x}) = \left(\widehat{f}^{ACES^{(1)}}(\boldsymbol{x}),\ldots,\widehat{f}^{ACES^{(s)}}(\boldsymbol{x})\right)$. It is important to note that constraints (28.3) and (28.4) guarantee monotonicity and concavity for each function $\widehat{f}^{ACES^{(r)}}(\boldsymbol{x})$, $r=1,\ldots,s$. This implies that applying or omitting these constraints will result in each $\widehat{f}^{ACES^{(r)}}(\boldsymbol{x})$ being non-decreasing, concave, both, or neither, thereby effectively enveloping the observed data by constraints (28.1) and (28.2). Additionally, this approach is flexible enough to allow different shape constraints to be applied to each output r.

Model (26) is separable, meaning that given a common set of BFs, the coefficients α and β for each output are estimated independently. Consequently, the overall error is minimized by addressing each output's error separately. However, this separability can lead to certain issues. To illustrate this situation, consider a dataset of 50 DMUs, each characterized by two inputs (x_1,x_2) and two outputs (y_1,y_2), generated according to the methodology outlined by Perelman and Santín (2009) that meets usual microeconomic postulates. Additionally, the maximum output producible for a given set of inputs is denoted as (y_1^*,y_2^*). Data are generated without random error. Subsequently, we estimate these optimal outputs, yielding $(\widehat{y}_1,\widehat{y}_2)$. To further evaluate the method, three additional columns are included in the table below, representing the output-oriented radial score calculated by standard DEA, as defined by Banker et al. (1984). Specifically, $\phi^{(r)}$ is the output-oriented radial score utilizing all inputs and only the $r-$th output, while ϕ is the output-oriented radial score utilizing all the available variables. The following table presents data for four representative units, highlighting the performance of our approach:

Table 1 revels that for the resource levels of $x_1=39.96$ and $x_2=15.47$, unit 4 needs to produce $y_1^*=189.98$ and $y_2^*=539.29$ to achieve technical efficiency. Similarly, unit 23, with resources $x_1=38.30$ and $x_2=14.85$, should aim to produce $y_1^*=563.14$ and $y_2^*=240.27$. In these cases, production should be mainly concentrated in y_2 or y_1, respectively. Conversely, for resource levels presented in units 22 and 49, the optimal production levels are similar for both outputs. Additionally, we can evaluate each unit's relative position with respect to single-output and multi-output analyses by using the scores $\phi^{(1)}$, $\phi^{(2)}$ and ϕ. For example, units 22 and 49 have similar positions relative to the frontier in both single-output and multi-output contexts. However, unit 4 is significantly farther from the frontier when only output y_1 is considered compared to when both outputs are accounted for. A similar situation occurs for output y_2 and unit 23. The ACES model estimates $(\widehat{y}_1,\widehat{y}_2)$ were generated using model (26), incorporating monotonicity (26.3) and concavity (26.4) constraints. The results show that prediction for both outputs are similar across all four units, exhibiting poor performance in estimating y_1^* for unit 4 and y_2^* for unit 23. Precisely, these units exhibit greater discrepancies when comparing single-output and multi-output analyses. This situation underscores the importance of the refinement step, where replacing the predicted values with observed values could significantly enhance the technique's performance.

When applying an ACES model in a multi-output context, the model's accuracy is highly dependent on its ability to predict each output variable effectively. Poor predictions, even for a single output, can significantly distort the overall efficiency assessment of DMUs. In

contrast, projecting only a few units onto the frontier can achieve a good characterization of the technology. Adopting an approach where only a portion of the observed output is substituted, while preserving the remainder, has proven beneficial in improving the model performance, as we will show. As illustrated in Table 1, the ACES model performs well when the DMU's relative position remains consistent across both single-output and multi-output analyses, thereby accurately capturing the DMU's maximum production capacity. Considering these factors, and to mitigate the negative effects of inaccurate predictions, we implement the following strategy to refine the ACES estimates:

$$\widehat{y}_{ir} = \begin{cases} \widehat{f}^{ACES^{(r)}}(\boldsymbol{x}_i), & |\phi^{(r)} - \phi| < \rho \\ y_{ir}, & |\phi^{(r)} - \phi| \geqslant \rho \end{cases}, i = 1, ..., n, r = 1, ..., s, \tag{28}$$

where $\phi^{(r)}$ is the DEA-based output-oriented radial score utilizing all inputs and only the $r-$th output, ϕ is the DEA-based output-oriented radial score utilizing all the available variables, $\widehat{f}^{ACES^{(r)}}(\boldsymbol{x}_i)$ is the $r-$th output prediction for the $i-$th DMU by an ACES model, and ρ is a threshold that prioritizes either predicted output $\widehat{f}^{ACES^{(r)}}(\boldsymbol{x})$ or observed output y_{ir} to constitute $\{(\boldsymbol{x}_i^*, \widehat{\boldsymbol{y}}_i)\}_{i=1}^n$. Following our experience, a threshold value of $\rho = 0.05$ generally performs well across different scenarios, while a value of $\rho = 0$ corresponds to the standard DEA technology. Notably, in the case of considering a single output all observed units are replaced by the predicted units.

Finally, we can define the ACES technology as follows:

$$\widehat{\varphi}_{ACES} = \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in R_+^{m+s} : \boldsymbol{x}_j \geqslant \sum_{i=1}^n \lambda_i \boldsymbol{x}_{ij}, j = 1, \ldots, m, y_r \leqslant \sum_{i=1}^n \lambda_i \widehat{y}_{ir}, r = 1, \ldots, s, \sum_{i=1}^n \lambda_i = 1, \lambda_i \geqslant 0, i = 1, \ldots, n \right\}. \tag{29}$$

Several aspects merit consideration when comparing this new DEA-type technology with the well-established DEA-VRS technology ($\widehat{\varphi}_{DEA}$) defined in (2). In the standard DEA framework, technology construction hinges on historical data, where the observed input and output quantities for all DMUs form the "core" dataset. This data is then used to incorporate new virtual productions into the technology set, guided by assumptions like convexity, free disposability, and minimal extrapolation. Traditional DEA addresses the question of 'What other input–output combinations can be guaranteed as producible based on the observed units?' assuming that some efficient DMUs are always observed. However, the ACES framework shifts away from this assumption. Here, the observation of truly efficient DMUs is no longer assumed. Instead, the aim is to establish a (more realistic) technology by including input–output combinations that are better than those observed. This is done by varying the primary dataset composition. In this scenario, the primary data is no longer just the observed units but includes a virtual sample generated by the ACES model. This new sample contains the m original input vectors, as well as a set of r (new) "pushed-up" output vectors.

Convexity and free disposability of inputs and outputs in (29) are easily verified. It can also be proved that $\widehat{\varphi}_{DEA} \subseteq \widehat{\varphi}_{ACES}$. The minimal extrapolation assumption is not imposed; instead, our approach positions the frontier as closely as possible to the new virtual data sample. In this way, the issue of overfitting can be addressed.

4. Measuring technical efficiency through Adaptive Constrained Enveloping splines

Expression (29) defines a technology under variable returns to scale that is separated from the observed data set by eliminating the minimum extrapolation axiom. Next, we show how to measure the efficiency of a DMU with input–output bundle $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ using $\widehat{\varphi}_{ACES}$ depending on the type of measure considered (see, for example, Pastor et al., 2012).

The output-oriented radial measure (Banker et al., 1984):

$$\begin{aligned} \underset{\phi,\lambda}{\text{maximize}} \quad & \phi \\ \text{subject to} \quad & \\ & \sum_{i=1}^n \lambda_i \boldsymbol{x}_{ij} \leqslant \boldsymbol{x}_{0j}, \quad \forall j, \\ & \sum_{i=1}^n \lambda_i \widehat{y}_{ir} \geqslant \phi y_{0r}, \quad \forall r, \\ & \sum_{i=1}^n \lambda_i = 1, \\ & \lambda_i \geqslant 0, \quad \forall i. \end{aligned} \tag{30}$$

The input-oriented radial measure (Banker et al., 1984):

$$\begin{aligned} \underset{\theta,\lambda}{\text{minimize}} \quad & \theta \\ \text{subject to} \quad & \\ & \sum_{i=1}^n \lambda_i \boldsymbol{x}_{ij} \leqslant \theta \boldsymbol{x}_{0j}, \quad \forall j, \\ & \sum_{i=1}^n \lambda_i \widehat{y}_{ir} \geqslant y_{0r}, \quad \forall r, \\ & \sum_{i=1}^n \lambda_i = 1, \\ & \lambda_i \geqslant 0, \quad \forall i. \end{aligned} \tag{31}$$

The Directional Distance Function (Chambers et al., 1998):

$$\begin{aligned} \underset{\delta,\lambda}{\text{maximize}} \quad & \delta \\ \text{subject to} \quad & \\ & \sum_{i=1}^n \lambda_i \boldsymbol{x}_{ij} \leqslant \boldsymbol{x}_{0j} - \delta G_{x_j}, \quad \forall j, \\ & \sum_{i=1}^n \lambda_i \widehat{y}_{ir} \geqslant y_{0r} + \delta G_{y_r}, \quad \forall r, \\ & \sum_{i=1}^n \lambda_i = 1, \\ & \lambda_i \geqslant 0, \quad \forall i. \end{aligned} \tag{32}$$

Here $(\boldsymbol{G}_x, \boldsymbol{G}_y) = (G_{x_1}, \ldots, G_{x_m}, G_{y_1}, \ldots, G_{y_s})$ represents a directional projection vector to the frontier, where $(\boldsymbol{x}_0, \boldsymbol{y}_0) + \delta(\boldsymbol{G}_x, \boldsymbol{G}_y)$, with $\delta \geqslant 0$, intersects the frontier.

Additionally, other well-known efficiency measures can be determined in a similar manner. Additive models such as the Measure of Inefficiency Proportions (Cooper et al., 1999), the Normalized Weighted Additive Model (Lovell and Pastor, 1995), the Range Adjusted Measure (Cooper et al., 1999), and the Bounded Adjusted Measure (Cooper et al., 2011) can be applied. Likewise, input- and output-oriented non-radial Russell measures (Färe and Lovell, 1978; Färe et al., 1985) and the Enhanced Russell Graph Measure (Pastor et al., 1999) are also compatible with this approach, among others.

5. Computational experiments and hyperparameter tuning

This section is divided into three subsections. The first introduces two distinct sets of computational experiments, based on the frameworks proposed by Perelman and Santín (2009) and Fare et al. (1994), respectively. Each experimental scenario is defined by three key parameters: the sample size, the number of truly efficient units located on the underlying production frontier, and the presence or absence of random noise. In scenarios with noise, some observations may lie above the true frontier. These experiments are designed to evaluate the performance of the proposed ACES methodology in comparison with several well-established techniques in the field, including DEA by

Banker et al. (1984), StoNED by Kuosmanen and Johnson (2017), the convexified version of EAT (CEAT) by Esteve et al. (2020), and Bootstrap DEA (BDEA) by Simar and Wilson (1998, 2000).

Both experimental designs involve estimating the true radial output score (ϕ). To this end, we employ the available data to estimate ϕ using different techniques, denoted as $\widehat{\phi}^{ACES}$, $\widehat{\phi}^{DEA}$, $\widehat{\phi}^{StoNED}$, $\widehat{\phi}^{CEAT}$ and $\widehat{\phi}^{BDEA}$. The $\widehat{\phi}^{ACES}$ score is obtained using the methodology outlined in this paper, with model (29) used as the final step. The $\widehat{\phi}^{DEA}$ score is computed following the standard procedure described in Banker et al. (1984). The $\widehat{\phi}^{StoNED}$ score is derived using the StoNED method (Kuosmanen and Johnson, 2017), implemented via the Python package pyStoNED as described by Dai et al. (2021). Specifically, to perform StoNED, we use the directional projection vector to the frontier $(\boldsymbol{G}_x, \boldsymbol{G}_y) = (0, 0, \sigma_{y_1}, \sigma_{y_2})$, where σ_{y_1} and σ_{y_2} denote the standard deviation of outputs y_1 and y_2, respectively (see Kuosmanen and Johnson, 2017). The $\widehat{\phi}^{CEAT}$ score is calculated using the convexified version of EAT (Esteve et al. 2020), which replaced the original stepwise frontier with a piecewise lineal production function. To implement this, we rely on the eat R package (Esteve et al. 2022), using its default configuration of five units per terminal node to prevent excessive splitting. Finally, the $\widehat{\phi}^{BDEA}$ score is obtained by applying the Bootstrap DEA method (Simar and Wilson, 1998, 2000) using the Benchmarking R package (Bogetoft et al., 2015) under variable returns to scale and 1,000 bootstrap replications.

The second subsection introduces the set of hyperparameters available to configure the ACES methodology. It specifies the values adopted for each hyperparameter under the two experimental designs described previously. In addition, it provides a detailed procedure for configuring ACES in general, offering practical guidance on how to tune the method for any given application scenario.

The third subsection presents the results obtained from the simulations. For each combination of scenario, 100 independent trials were conducted to evaluate the relative performance of the methods. Three evaluation criteria are considered: Mean Squared Error (MSE), bias, and computational time. The MSE captures the average magnitude of the estimation error by measuring the squared differences between the estimated and true radial output scores. Bias quantifies the average direction of the error, indicating whether a method systematically overestimates or underestimates the true frontier. Computational time, in turn, provides a practical assessment of the algorithm's efficiency, reflecting its applicability to large-scale or time-sensitive problems. Together, these metrics offer a comprehensive assessment of each model's accuracy, estimation behavior, and computational performance.

Finally, we present the computational resources used for the estimations. Specifically, ACES and DEA evaluations were performed on the Dantzig Cluster at Miguel Hernández University (UMH), using a Supermicro SYS-120GQ-TNRT node equipped with two Intel® Xeon® Silver 4316 CPUs at 2.30 GHz, providing 80 cores and 768 GB of RAM. The simulations were executed under Rocky Linux 8.7, using R version 4.2.3. The complete ACES implementation is publicly available at https://github.com/Victor-Espana/aces. Optimization problems were solved using the Rglpk package (Theussl and Hornik, 2019). StoNED estimations were carried out in Python using the pystoned library (Dai et al., 2021), with models solved locally by using the default MOSEK solver (Mosek, 2021) .

5.1. Experimental setup

This section presents two complementary experimental designs, inspired by the methodologies outlined in Perelman and Santín (2009) and Fare et al. (1994) , which serve as the basis for evaluating the performance of the proposed approach.

5.1.1. Perelman and Santín (2009)

In the first design, following Perelman and Santín (2009), we simulate datasets with two inputs and two outputs that satisfy standard microeconomic postulates. Input variables are randomly drawn from a uniform distribution $U(5, 50)$, while the outputs in the production frontier are generated through the following formula:

$$\begin{aligned} -\ln(y_1^*) = \; & -1 + 0.5\cdot\ln\left(\frac{y_2}{y_1}\right) + 0.25\cdot\ln\left(\frac{y_2}{y_1}\right)^2 - 1.5\cdot\ln(x_1) - 0.6\cdot\ln(x_2) + 0.2\cdot\ln(x_1)^2 + 0.05\cdot\ln(x_2)^2 \\ & -0.1\cdot\ln(x_1)\cdot\ln(x_2) + 0.05\cdot\ln(x_1)\cdot\ln\left(\frac{y_2}{y_1}\right) - 0.05\cdot\ln(x_2)\cdot\ln\left(\frac{y_2}{y_1}\right), \\ -\ln(y_2^*) = \; & -\ln(y_1^*) - \ln\left(\frac{y_2}{y_1}\right), \end{aligned} \tag{33}$$

where $\ln\left(\frac{y_2}{y_1}\right) \sim U(-1.5, 1.5)$. To introduce the inefficiency term, a half normal distribution $u \sim |N(0, \sqrt{0.3})|$ was used. Random noise was incorporated through normal distributions $v_1, v_2 \sim N(0, \sqrt{0.01})$.

Consequently, observed outputs, which reflect technical inefficiency, are calculated as follows:

$$\begin{aligned} y_1 &= y_1^* \cdot \frac{1}{e^u}, \\ y_2 &= y_2^* \cdot \frac{1}{e^u}. \end{aligned} \tag{34}$$

Additionally, to incorporate random error, the following formulas were applied:

$$\begin{aligned} y_1 &= y_1^* \cdot \frac{1}{e^u} \cdot \frac{1}{e^{v_1}}, \\ y_2 &= y_2^* \cdot \frac{1}{e^u} \cdot \frac{1}{e^{v_2}}. \end{aligned} \tag{35}$$

In this context, the true radial score is defined as:

$$\phi = \frac{y_1^*}{y_1} = \frac{y_2^*}{y_2} \tag{36}$$

For each scenario, five distinct sample sizes are analyzed: 50, 100, 150, 200, and 300 observations. Additionally, we consider four different proportions of DMUs located on the true frontier: 0 %, 5 %, 10 %, and 20

Table 2
Configuration of producer groups by scale level and data generation parameters.

Producer size	$f(x) = h(y)$	ϖ	x_1	y_1^*
Small	25	0.898	[20, 60]	[10, 35]
Medium I	50	1.000	[30, 80]	[15, 70]
Medium II	75	1.000	[50, 100]	[25, 100]
Large	100	0.927	[90, 230]	[45, 135]

Table 3
Set of available hyperparameters to perform an ACES model.

Hyperparameter	Description
η	Maximum number of BFs allowed in the model after the forward algorithm.
ξ	Maximum error reduction rate required to add a BF to the model during the forward algorithm.
$q_{\max}$	Maximum degree of variable interaction during the forward algorithm.
$\xi^{(q)}$	Minimum improvement ratio over the best 1-degree BF required to include a $q-$th degree BF in the model.
minspan	Minimum number of observations required between two consecutive knots. This can be a fixed integer or one of the adaptations proposed by Friedman (1991) and Zhang (1994).
endspan	Minimum number of observations required between the extreme knots and the extremes of the data. This can be an integer or one of the adaptations proposed by Friedman (1991) and Zhang (1994).
grid	Set of possible locations for the knots. In Friedman (1991), the observed data points are used. However, this value can be varied to reduce the computational load of the forward algorithm.
LOF	Model coefficients are estimated using the LP models proposed throughout the text. However, for basis evaluation, Mean Squared Error or any other lack-of-fit metric, such as Mean Absolute Error, can be used.
d	Penalty factor for retaining knots during the backward algorithm.
ρ	Threshold that determines if the predicted output or the observed value is used in the refinement step.

%. For these units, equations (34) and (35) are not applied, and therefore the observed outputs coincide with the theoretical ones, i.e., $y_r = y_r^*$, $r = 1, 2$. Finally, each scenario is further classified into two variants—those with random noise and those without. In the noisy case, equation (35) is applied to introduce stochastic deviations, which may cause some observations to appear above the true frontier. In the noise-free variant, only technical inefficiency is considered through equation (34), unless the unit lies on the true frontier, in which case no distortion is applied.

5.1.2. Färe et al. (1994)

The second experimental design is adapted from Fare et al. (1994) and reflects a stylized production setting where technologies exhibit increasing, constant, and decreasing returns to scale. These simulations are constructed in a two-input, two-output setting. In this framework, the input side of the production function is modeled using a log-linear Cobb–Douglas specification, while the output side follows a Constant Elasticity of Transformation (CET) function. The input–output combinations are simulated to satisfy the identity:

$$f(x) = h(y), \tag{37}$$

Table 4
Performance comparison of different approaches in the first stage of ACES in scenarios without random noise.

border	n	Approach 1 Monotonicity and concavity	Approach 2 Only monotonicity	Approach 3 Only concavity	Approach 4 Only envelopment
0 %	50	0.095	0.105	0.116	0.133
	100	0.066	0.058	0.061	0.067
	150	0.053	0.048	0.045	0.044
5 %	50	0.073	0.085	0.098	0.104
	100	0.055	0.052	0.052	0.055
	150	0.057	0.045	0.046	0.041
10 %	50	0.063	0.080	0.096	0.094
	100	0.055	0.049	0.046	0.046
	150	0.053	0.046	0.041	0.040
20 %	50	0.063	0.069	0.064	0.064
	100	0.060	0.045	0.037	0.041
	150	0.061	0.053	0.049	0.037

Table 5
Performance comparison of different approaches in the first stage of ACES in scenarios with random noise.

border	n	Approach 1 Monotonicity and concavity	Approach 2 Only monotonicity	Approach 3 Only concavity	Approach 4 Only envelopment
0 %	50	0.115	0.128	0.131	0.163
	100	0.092	0.083	0.088	0.085
	150	0.100	0.079	0.081	0.067
5 %	50	0.102	0.120	0.122	0.118
	100	0.095	0.074	0.074	0.081
	150	0.089	0.080	0.072	0.064
10 %	50	0.101	0.101	0.101	0.133
	100	0.094	0.078	0.071	0.077
	150	0.089	0.080	0.074	0.067
20 %	50	0.093	0.098	0.078	0.103
	100	0.092	0.085	0.069	0.064
	150	0.098	0.080	0.076	0.061

where

$$f(x) = (\sqrt{x_1} \cdot \sqrt{x_2})^{\varpi}, \tag{38}$$

and

$$h(y) = \sqrt{\frac{1}{2}y_1^2 + \frac{1}{2}y_2^2}. \tag{39}$$

From equations (38) and (39), we define four different groups of producers. Once the sample size is fixed, each observation is randomly assigned to one of these groups—small, medium I, medium II, or large—with equal probability (i.e., each group receives close to 25 % of the total observations). This approach ensures a balanced representation across different production technologies and supports a realistic simulation of varying returns to scale. For each group, the first input (x_1) and the first output (y_1^*) are independently drawn from uniform distributions, specifically $x_1 \sim U(a_x, b_x)$ and $y_1^* \sim U(a_y, b_y)$, where (a_x, b_x) and (a_y, b_y) represent the lower and upper bounds for the input and output variables, respectively. These bounds vary across groups to reflect differences in scale and production capacity.

Therefore, each group is characterized by a common frontier identity $f(x) = h(y)$, a group-specific scale elasticity ϖ used in equation (38), and the specific data generation intervals for x_1 and y_1^*. Table 2 summarizes the configuration of each group:

Once the values of x_1 and y_1^* are generated, the second input (x_2) and the second output (y_2^*) are computed to ensure that each observation satisfies the pre-assigned values of $f(x)$ and $h(y)$. Regarding, inefficiency and random noise, these are introduced following the same procedure described in the Perelman and Santín (2009) design.

Finally, the true radial score is computed as:

$$\phi = \frac{f(x)}{h(y)} \tag{40}$$

In this case, each scenario considers five distinct sample sizes: 25, 75, 125, 175, and 250 observations. As before, we evaluate four different proportions of DMUs located on the true frontier: 0 %, 5 %, 10 %, and 20 %, under both noise-free and noisy conditions.

5.2. The set of available hyperparameters

The ACES algorithm offers a wide range of hyperparameters that allow for the customization of the model according to specific requirements and data characteristics. Throughout the text, several important hyperparameters have been described, which are outlined in Table 3:

For optimal hyperparameter selection, techniques like k-fold cross-

Table 6
Effect of hyperparameter configuration on estimation error and computation time.

Configuration	ξ	$\xi^{(q)}$	Monotonicity	Concavity	MSE	Time	Ranking_CV
1	0.005	0	TRUE	TRUE	0.02	88	4
2	0.010	0.50	FALSE	TRUE	0.02	61	1
3	0.005	0	TRUE	FALSE	0.04	158	13
4	0.010	0.10	TRUE	TRUE	0.05	52	3
5	0.010	0.20	TRUE	TRUE	0.05	52	7
36	0.010	0.05	FALSE	FALSE	0.12	128	36
37	0.005	0.10	FALSE	FALSE	0.12	128	23
38	0.010	0.10	FALSE	FALSE	0.12	128	35
39	0.005	0.20	FALSE	FALSE	0.12	128	34
40	0.010	0.20	FALSE	FALSE	0.12	128	40

validation (CV) are recommended. In our computational experiments, a 5-fold CV approach was used, where the data is divided into five subsets. The model is trained on four folds and tested on the fifth, repeating the process so each fold serves as a test set once. In order to improve the relevance of the evaluation in efficiency analysis, the predictive errors on the test fold are weighted using DEA-based radial output efficiency scores. Specifically, for each unit in the test fold, a DEA model is estimated using only the test data, and the inverse of the resulting efficiency score is used as a weight. This allows us to emphasize errors in more (out-of-sample) efficient DMUs, which are more relevant from a benchmarking perspective. The squared differences between the (optimal) predicted and observed outputs are computed and weighted accordingly, producing a weighted Mean Squared Error for each fold. These values are then averaged across folds to guide hyperparameter selection.

For standard scenarios, the following hyperparameter values are recommended for testing. For η, multiples of 10 should be considered, up to the total number of observations in the dataset. Regarding ξ, typical values could range from 0, 0.005, 0.01, 0.02 or even 0.05 if the sample size is particularly large. For $q_{\max}$, it is advisable to test values of 1 or 2, and in cases where the number of features allows, 3 can be considered—although higher values may lead to excessive expansion of the feature space and increased computational burden. For $\xi^{(q)}$, values such as 0, 0.05, 0.10, 0.20, or 0.50 could be tested. However, when $\xi^{(q)}$ is close to 0.50, the model often defaults to a first-degree approximation. As a good practice, if the goal is to explore higher-degree BFs, the maximum model degree should be set greater than one, and at least one relatively high value of $\xi^{(q)}$ should be included to ensure that only sufficiently effective higher-degree terms are retained. For both *minspan* and *endspan*, it is recommended to use the values proposed by Friedman (1991) and Zhang (1994), or, alternatively, to omit these parameters. As for the *grid*, unless computational issues arise, it is recommended to use the dataset's own values for the knots. Regarding, for the *LOF* criterion, it is suggested to keep the Mean Squared Error (MSE) as the default, while for the penalty factor *d*, reasonable values to test are 0, 1, and 2.

All the aforementioned hyperparameters directly affect the estimation of the optimal output vector $\widehat{\boldsymbol{y}}_i$, as they intervene during the model fitting process. In contrast, the threshold ρ, used in the refinement step (28), does not influence the estimation of $\widehat{\boldsymbol{y}}_i$, but rather determines how the virtual dataset $\left\{\left(\boldsymbol{x}_i^*, \widehat{\boldsymbol{y}}_i\right)\right\}_{i=1}^{n}$ is constructed by deciding when to replace a predicted output with its observed counterpart. Typical values include 0, 0.05, or 0.10, depending on how conservatively one wishes to correct potentially overestimated outputs. As a rule of thumb, $\rho = 0.05$ is a reasonable and robust choice across various scenarios.

Due to the large number of simulations carried out in this study, it was not feasible to perform an exhaustive search over the entire hyperparameter space. For this reason, some hyperparameter were fixed while others were varied. The maximum degree of variable interaction $q_{\max}$ was fixed at 2; the maximum number of BFs η was set to the number of DMUs in the analysis; the minimum error reduction rate ξ was set to 0.005; the knot penalty *d* was tested with values 1 and 2; the minimum span followed Friedman's (1991) recommendation, while the end span was not used; the knot grid was defined using the observed data points; and the lack-of-fit criterion was based on the Mean Squared Error. Additionally, the threshold ρ used in the refinement step was set to 0.05. Finally, the hyperparameter $\xi^{(q)}$ was tested across a range of values specific to each simulation setting. In the experiments based on Perelman and Santín (2009), we used $\xi^{(2)} \in \{0.10, 0.20, 0.50\}$, while in the simulations following Fare et al. (1994), we tested $\xi^{(2)} \in \{0, 0.05, 0.10\}$.

Beyond the hyperparameters listed in Table 2, model performance can also be influenced by the shape constraints imposed during the first stage of the method. These constraints—specifically, monotonicity and concavity—can be selectively applied depending on the application context. For instance, to construct a fully flexible envelopment model, both constraints (26.3) and (26.4) should be omitted. If only non-decreasing monotonicity is desired, constraint (26.3) should be enforced, whereas constraint (26.4) ensures concavity. When both properties are required, both constraints must be simultaneously applied.

It is important to note that both the introduction of interaction terms and the refinement phase can disrupt the shape constraints imposed during the first estimation stage. In particular, the interactions transform the input space in a way that may not preserve monotonicity and concavity in the original dimensions, while the refinement step—by selectively replacing predicted outputs with observed ones—can directly violate any previously enforced structural properties. This naturally leads to the following question: given that the final DEA-type technology is constructed in the last step by applying Equation (29), which inherently satisfies all shape axioms, is it truly necessary—or even beneficial—to enforce monotonicity and concavity during the initial estimation phase? In principle, these shape constraints could be treated as an additional hyperparameter of the model, selected adaptively to best suit the data at hand. However, instead of adopting this strategy, we aim to offer practical guidance by analyzing the conditions under which each configuration of shape constraints performs best. To this end, we conduct a simulation study based on the design of Perelman and Santín (2009), evaluating the predictive performance of four constraint configurations across multiple sample sizes and noise levels.

The results are presented in Table 4 for noise-free scenarios and in Table 5 for scenarios with random noise. In both cases, 100 independent trials were performed for every combination of sample size and proportion of units located on the true frontier. Four model configurations were considered: (i) both monotonicity and concavity imposed (approach 1); (ii) only monotonicity (approach 2); (iii) only concavity (approach 3); and (iv) no shape constraints (approach 4). In all settings, the envelopment condition was enforced to ensure that predicted outputs lie above the observed data.

In both scenarios, clear trends emerge that provide guidance for characterizing ACES. When dealing with a small sample size, such as 50 units or fewer, it is generally advisable to impose both shape constraints (approach 1) regardless of the proportion of units on the true frontier. Conversely, with a sample size of 100 units or more, it becomes preferable to impose only one of the shape constraints, regardless of which

Table 7
Computational experiments in scenarios without random noise in Perelman and Santín (2009).

% Eff. points	n	Mean Squared Error (ACES vs baseline model)					Bias					Computation time				
		ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED
0 %	50	0.115	0.206 (−44.2 %)	0.129 (−11.1 %)	1.596 (−92.8 %)	0.169 (−31.9 %)	−0,194	−0,333	−0,200	0,560	−0,169	10	0	5	4	1
	100	0.071	0.137 (−48.2 %)	0.078 (−9.7 %)	1.962 (−96.4 %)	0.132 (−46.3 %)	−0,148	−0,261	−0,131	0,690	−0,213	34	0	13	11	1
	150	0.063	0.103 (−38.9 %)	0.056 (+11.3 %)	2.207 (−97.2 %)	0.118 (−46.7 %)	−0,082	−0,223	−0,102	0,778	−0,229	323	0	24	22	3
	200	0.051	0.083 (−38.2 %)	0.043 (+18.3 %)	2.320 (−97.8 %)	0.105 (−51.3 %)	−0,063	−0,197	−0,081	0,801	−0,210	989	0	35	38	16
	300	0.038	0.056 (−31.7 %)	0.027 (+41.9 %)	3.343 (−98.8 %)	0.100 (−61.6 %)	−0,021	−0,163	−0,056	1,071	−0,212	2920	0	93	95	21
5 %	50	0.100	0.171 (−41.3 %)	0.106 (−4.9 %)	1.859 (−94.6 %)	0.172 (−41.5 %)	−0,142	−0,286	−0,146	0,647	−0,142	10	0	5	4	0
	100	0.056	0.109 (−48.6 %)	0.064 (−11.8 %)	2.512 (−97.8 %)	0.100 (−43.9 %)	−0,105	−0,214	−0,083	0,837	−0,184	40	0	12	10	1
	150	0.047	0.082 (−42.7 %)	0.048 (−0.8 %)	2.071 (−97.7 %)	0.099 (−52.3 %)	−0,041	−0,177	−0,051	0,739	−0,194	342	0	22	19	5
	200	0.043	0.056 (–23.6 %)	0.030 (+45.0 %)	2.557 (−98.3 %)	0.092 (−53.5 %)	−0,010	−0,150	−0,032	0,891	−0,189	909	0	34	31	12
	300	0.036	0.044 (−19.9 %)	0.025 (+44.7 %)	3.630 (−99.0 %)	0.092 (−61.4 %)	0,018	−0,122	−0,013	1,102	−0,201	3161	0	77	93	22
10 %	50	0.082	0.149 (−44.7 %)	0.092 (−10.5 %)	1.846 (−95.5 %)	0.144 (−42.7 %)	−0,105	−0,260	−0,123	0,697	−0,070	15	0	8	5	0
	100	0.049	0.091 (−46.6 %)	0.054 (−9.8 %)	1.846 (−97.4 %)	0.095 (−48.4 %)	−0,075	−0,185	−0,052	0,747	−0,159	51	0	19	17	1
	150	0.044	0.061 (−28.6 %)	0.035 (+23.8 %)	2.161 (−98.0 %)	0.085 (−48.6 %)	−0,015	−0,146	−0,021	0,788	−0,165	390	0	25	23	3
	200	0.041	0.047 (−12.3 %)	0.027 (+52.6 %)	2.595 (−98.4 %)	0.081 (−48.6 %)	0,015	−0,123	−0,005	0,899	−0,162	967	0	35	33	8
	300	0.037	0.033 (+13.8 %)	0.021 (+80.6 %)	3.249 (−98.8 %)	0.071 (−47.7 %)	0,047	−0,091	0,015	1,051	−0,147	3041	0	75	71	29
20 %	50	0.060	0.105 (−42.9 %)	0.067 (−10.5 %)	1.966 (−97.0 %)	0.146 (−59.1 %)	−0,052	−0,195	−0,057	0,757	−0,056	10	0	5	3	0
	100	0.035	0.070 (−49.5 %)	0.048 (−26.2 %)	2.288 (−98.5 %)	0.070 (−49.5 %)	−0,026	−0,134	−0,005	0,831	−0,100	37	0	12	10	1
	150	0.043	0.044 (−3.4 %)	0.029 (+47.8 %)	2.556 (−98.3 %)	0.063 (–32.4 %)	0,034	−0,103	0,014	0,905	−0,097	411	0	28	30	6
	200	0.042	0.038 (+9.5 %)	0.027 (+52.7 %)	2.626 (−98.4 %)	0.054 (–22.7 %)	0,053	−0,084	0,026	0,926	−0,098	883	0	34	37	6
	300	0.033	0.028 (+16.7 %)	0.020 (+58.8 %)	3.022 (−98.9 %)	0.051 (−36.8 %)	0,065	−0,065	0,033	1,022	−0,081	2522	0	77	73	33

one (approach 2 or 3). When the sample size reaches 150 units or more, it is most effective to impose the envelopment conditions (approach 4). This trend occurs because, as the sample size increases, the units naturally tend to align more closely with the true production frontier and its geometrical features. In contrast, with smaller sample sizes, the additional information from the shape of the DGP improves model performance.

Although the section provides practical guidance on how to select the hyperparameters of the ACES model, in practice—and whenever computational resources allow—it is recommended to perform a CV procedure to identify the most suitable configuration. To illustrate this, we present a minimal example based on the simulation framework proposed by Fare et al. (1994) , using a dataset of 100 DMUs. This experiment explores the sensitivity of the method to hyperparameter selection by applying a 5-fold CV under varying configurations, assessing its impact on both the prediction error and the computational time. In particular, we test two values for the minimum error reduction rate $\xi \in \{0.005, 0.01\}$; five values for the interaction threshold $\xi^{(2)} \in \{0, 0.05, 0.10, 0.20, 0.50\}$; and all possible combinations of

Table 8
Computational experiments in scenarios with random noise in Perelman and Santín (2009).

% Eff. points	n	Mean Squared Error (ACES vs baseline model)					Bias					Computation time				
		ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED
0 %	50	0,143	0,233 (−38.8 %)	0,158 (−9.4 %)	1,785 (−92.0 %)	0,234 (−39.0 %)	−0,153	−0,319	−0,172	0,658	−0,137	15	0	8	5	0
	100	0,085	0,127 (–32.8 %)	0,083 (+3.1 %)	2,004 (−95.8 %)	0,143 (−40.3 %)	−0,094	−0,217	−0,077	0,729	−0,214	36	0	13	11	1
	150	0,087	0,116 (−24.3 %)	0,083 (+5.8 %)	2,510 (−96.5 %)	0,141 (−38.0 %)	−0,043	−0,174	−0,036	0,822	−0,222	321	0	24	22	3
	200	0,077	0,098 (−21.5 %)	0,075 (+2.0 %)	3,058 (−97.5 %)	0,131 (−41.6 %)	0,003	−0,138	−0,002	1,001	−0,212	1036	0	30	49	10
	300	0,081	0,069 (+17.0 %)	0,061 (+32.3 %)	4,159 (−98.1 %)	0,128 (−37.1 %)	0,066	−0,089	0,038	1,216	−0,217	3659	0	55	111	14
5 %	50	0,130	0,188)-30.7 %)	0,125 (+4.4 %)	1,748 (−92.6 %)	0,198 (−34.2 %)	−0,115	−0,277	−0,132	0,651	−0,137	14	0	4	5	0
	100	0,079	0,126 (−37.0 %)	0,087 (−9.4 %)	2,111 (−96.2 %)	0,132 (−40.0 %)	−0,082	−0,193	−0,052	0,741	−0,202	50	0	19	17	1
	150	0,074	0,085 (−12.7 %)	0,065 (+14.4 %)	2,582 (−97.1 %)	0,115 (−35.4 %)	0,000	−0,133	0,005	0,862	−0,186	434	0	34	34	3
	200	0,078	0,081 (−3.20 %)	0,067 (+17.9 %)	3,248 (−97.6 %)	0,109 (−28.3 %)	0,039	−0,108	0,024	1,027	−0,179	1203	0	56	59	9
	300	0,085	0,063 (+35.5 %)	0,059 (+43.4 %)	4,111 (−97.9 %)	0,113 (−25.0 %)	0,084	−0,069	0,053	1,203	−0,197	3941	0	102	109	15
10 %	50	0,113	0,185 (−39.2 %)	0,129 (−12.7 %)	2,075 (−94.6 %)	0,172 (−34.3 %)	−0,095	−0,250	−0,102	0,711	−0,125	11	0	5	3	0
	100	0,069	0,102 (–32.7 %)	0,074 (−7.0 %)	2,317 (−97.0 %)	0,125 (−45.0 %)	−0,043	−0,161	−0,016	0,803	−0,150	38	0	12	10	2
	150	0,072	0,085 (−15.3 %)	0,069 (+5.0 %)	2,845 (−97.5 %)	0,101 (−28.7 %)	0,020	−0,116	0,019	0,948	−0,156	338	0	23	20	4
	200	0,073	0,071 (+2,6%)	0,062 (+17.9 %)	3,504 (−97.9 %)	0,103 (−29.2 %)	0,050	−0,089	0,038	1,016	−0,170	994	0	36	33	8
	300	0,080	0,057 (+41.6 %)	0,057 (+41.8 %)	4,292 (−98.1 %)	0,096 (−16.0 %)	0,091	−0,052	0,065	1,250	−0,164	3508	0	71	68	14
20 %	50	0,090	0,138 (−34.4 %)	0,100 (−9,9%)	1,994 (−95.5 %)	0,178 (−49.4 %)	−0,036	−0,186	−0,045	0,766	−0,068	11	0	4	3	0
	100	0,055	0,079 (−31.2 %)	0,062 (−11.5 %)	2,204 (−97.5 %)	0,083 (−34.2 %)	−0,013	−0,121	0,012	0,846	−0,092	37	0	11	9	2
	150	0,063	0,066 (−5.1 %)	0,056 (+11.6 %)	2,403 (−97.4 %)	0,082 (–23.9 %)	0,048	−0,085	0,037	0,890	−0,119	317	0	22	19	5
	200	0,071	0,054 (+31.8 %)	0,050 (+40.5 %)	2,857 (−97.5 %)	0,075 (−5.5 %)	0,077	−0,061	0,055	0,987	−0,110	925	0	35	32	9
	300	0,067	0,047 (+44.6 %)	0,047 (+42.1 %)	3,726 (98.2 %)	0,067 (+0.1 %)	0,100	−0,036	0,066	1,174	−0,083	2730	0	75	78	23

Table 9
Computational experiments in scenarios withrandom noise in Fare et al. (1994).

% Eff. points	n	Mean Squared Error (ACES vs baseline model)					Bias					Computation time				
		ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED
0 %	25	0,243	0,406 (−40.2 %)	0,308 (21.2 %)	0,749 (−67.6 %)		−0,327	−0,453	−0,330	0,321		9	0	2	2	
	75	0,094	0,310 (−69.7 %)	0,227 (−58.5 %)	1,407 (−93.3 %)	0,242 (−61.2 %)	−0,180	−0,337	−0,205	0,626	−0,164	40	0	13	11	1
	125	0,053	0,259 (−79.4 %)	0,184 (−71.0 %)	1,426 (−96.3 %)	0,168 (−68.3 %)	−0,114	−0,282	−0,148	0,675	−0,159	71	0	19	18	5
	175	0,062	0,236 (−73.6 %)	0,167 (−62.7 %)	1,607 (−96.1 %)	0,233 (−73.2 %)	−0,118	−0,251	−0,122	0,758	−0,130	834	0	26	39	5
	250	0,050	0,236 (−78.6 %)	0,169 (−70.2 %)	2,284 (−97.8 %)	0,124 (−59.4 %)	−0,088	−0,227	−0,102	0,966	−0,145	2360	0	75	78	19
5 %	25	0,184	0,372 (−50.5 %)	0,289 (−36.2 %)	0,850 (−78.3 %)		−0,247	−0,386	−0,264	0,366		8	0	2	2	
	75	0,068	0,251 (−72.9 %)	0,173 (−60.8 %)	1,316 (−94.8 %)	0,303 (−77.6 %)	−0,131	−0,292	−0,159	0,653	−0,084	38	0	7	10	2
	125	0,047	0,242 (−80.8 %)	0,174 (−73.2 %)	1,472 (−96.8 %)	0,245 (−81.0 %)	−0,088	−0,255	−0,121	0,701	−0,107	97	0	25	26	3
	175	0,049	0,224 (−78.3 %)	0,161 (−69.8 %)	2,077 (−97.7 %)	0,123 (−60.4 %)	−0,086	−0,224	−0,098	0,878	−0,132	961	0	44	48	6
	250	0,042	0,214 (−80.4 %)	0,155 (−72.9 %)	2,671 (−98.4 %)	0,110 (−61.8 %)	−0,056	−0,199	−0,075	1,071	−0,111	2611	0	81	85	23
10 %	25	0,168	0,308 (−45.4 %)	0,230 (−27.1 %)	0,872 (−80.8 %)		−0,224	−0,351	−0,220	0,402		9	0	4	2	
	75	0,066	0,272 (−75.7 %)	0,205 (−67.7 %)	1,287 (−94.9 %)	0,205 (−67.7 %)	−0,097	−0,269	−0,134	0,641	−0,085	30	0	8	6	1
	125	0,039	0,227 (−82.7 %)	0,168 (−76.5 %)	1,508 (−97.4 %)	0,136 (−71.1 %)	−0,057	−0,226	−0,097	0,728	−0,129	76	0	18	15	3
	175	0,046	0,222 (−79.2 %)	0,165 (−72.0 %)	1,653 (−97.2 %)	0,116 (−60.3 %)	−0,068	−0,205	−0,078	0,793	−0,101	999	0	45	45	7
	250	0,039	0,216 (−82.0 %)	0,161 (−75.8 %)	2,514 (−98.5 %)	0,106 (−63.2 %)	−0,039	−0,187	−0,068	1,050	−0,099	2934	0	84	87	25
20 %	25	0,126	0,245 (−48.7 %)	0,182 (−30.8 %)	0,765 (−83.6 %)		−0,178	−0,299	−0,168	0,407		6	0	2	1	
	75	0,045	0,215 (−78.9 %)	0,164 (−72.4 %)	1,550 (−97.1 %)	0,276 (−83.6 %)	−0,058	−0,217	−0,088	0,729	−0,010	29	0	9	7	2
	125	0,031	0,185 (−83.3 %)	0,139 (−77.7 %)	1,883 (−98.4 %)	0,120 (−74.2 %)	−0,027	−0,185	−0,058	0,822	−0,084	74	0	17	15	3
	175	0,033	0,172 (−80.6 %)	0,129 (−74.0 %)	1,704 (−98.0 %)	0,086 (−61.3 %)	−0,031	−0,164	−0,046	0,818	−0,087	853	0	32	33	7
	250	0,028	0,177 (−84.2 %)	0,134 (−79.0 %)	2,260 (−98.8 %)	0,080 (−65.2 %)	−0,006	−0,150	−0,037	0,964	−0,091	2334	0	57	56	17

enforcing or not enforcing monotonicity and concavity.

Table 6 presents the five best and five worst hyperparameter configurations out of the forty evaluated in this experiment. For each configuration, the table reports the value of the assessed hyperparameters, the true mean squared error (MSE), the computational time (in seconds), and the ranking position derived from CV (column "Ranking_CV").

Some considerations must be made regarding the influence of hyperparameters on model performance and computational cost. First, the results depend heavily on the selected configuration, as even small changes in the hyperparameter values can lead to noticeable variations in both error and runtime. While cross-validation does not always select the absolute best configuration, it consistently identifies high-performing ones, making it the preferred approach for tuning in practical applications. Additionally, the computational time is strongly driven by these few hyperparameters—particularly the error reduction threshold ξ and the shape constraints. While the former allows for a longer forward selection stage, the latter can slow down the convergence of the model if shape constraints are needed but not enforced, as the model may compensate by incorporating a larger number of BFs to

Table 10
Computational experiments in scenarios without random noise in Fare et al. (1994). (* Omitted due to anomalous results).

% Eff. points	n	Mean Squared Error (ACES vs baseline model)					Bias					Computation time				
		ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED	ACES	DEA	BDEA	CEAT	StoNED
0 %	25	0,229	0,388 (−40.9 %)	0,285 (−19.6 %)	0,734 (−68.7 %)	*	−0,336	−0,449	−0,324	0,298	*	6	0	2	1	*
	75	0,096	0,278 (−65.5 %)	0,194 (−50.7 %)	1,094 (−91.2 %)	0,379 (−74.7 %)	−0,211	−0,353	−0,230	0,543	−0,066	36	0	7	9	2
	125	0,068	0,269 (−74.7 %)	0,187 (−63.7 %)	1,116 (−93.9 %)	0,209 (−67.5 %)	−0,175	−0,329	−0,207	0,567	−0,144	73	0	17	15	11
	175	0,070	0,255 (−72.7 %)	0,178 (−60.8 %)	1,412 (−95.1 %)	0,136 (−48.9 %)	−0,170	−0,302	−0,185	0,636	−0,184	867	0	32	30	12
	250	0,055	0,240 (−77.3 %)	0,168 (−67.4 %)	1,932 (−97.2 %)	0,119 (−54.0 %)	−0,142	−0,277	−0,165	0,825	−0,142	2542	0	58	56	16
5 %	25	0,184	0,370 (−50.4 %)	0,284 (−35.4 %)	0,853 (−78.5 %)	*	−0,272	−0,404	−0,283	0,377	*	9	0	4	2	*
	75	0,073	0,282 (−74.0 %)	0,204 (−64.2 %)	1,249 (−94.1 %)	0,237 (−69.1 %)	−0,165	−0,320	−0,192	0,595	−0,071	38	0	13	11	3
	125	0,049	0,228 (−78.7 %)	0,161 (−69.7 %)	1,174 (−95.9 %)	0,158 (−69.3 %)	−0,122	−0,269	−0,147	0,595	−0,139	98	0	19	25	22
	175	0,055	0,231 (−76.4 %)	0,164 (−66.7 %)	1,591 (−96.6 %)	0,138 (−60.5 %)	−0,131	−0,260	−0,143	0,747	−0,151	599	0	26	22	9
	250	0,041	0,221 (−81.4 %)	0,156 (−73.5 %)	2,121 (−98.1 %)	0,121 (−66.0 %)	−0,099	−0,238	−0,122	0,880	−0,122	1553	0	42	35	19
10 %	25	0,147	0,324 (−54.6 %)	0,242 (−39.2 %)	0,721 (−79.6 %)	*	−0,232	−0,366	−0,234	0,341	*	6	0	2	1	*
	75	0,067	0,252 (−73.5 %)	0,180 (−62.8 %)	1,519 (−95.6 %)	0,409 (−83.6 %)	−0,136	−0,294	−0,164	0,693	0,006	27	0	7	6	3
	125	0,040	0,216 (−81.3 %)	0,154 (−73.8 %)	1,449 (−97.2 %)	0,139 (−71.0 %)	−0,092	−0,249	−0,125	0,691	−0,115	73	0	17	15	7
	175	0,047	0,214 (−78.2 %)	0,151 (−69.0 %)	1,978 (−97.6 %)	0,136 (−65.7 %)	−0,095	−0,231	−0,111	0,867	−0,115	587	0	34	33	7
	250	0,033	0,200 (−83.6 %)	0,143 (77.0 %)	2,294 (−98.6 %)	0,098 (−66.4 %)	−0,067	−0,209	−0,095	0,953	−0,115	2280	0	52	51	27
20 %	25	0,144	0,289 (−50.2 %)	0,223 (−35.2 %)	0,978 (−85.2 %)	*	−0,198	−0,324	−0,200	0,478	*	6	0	2	1	*
	75	0,048	0,199 (−75.7 %)	0,146 (−66.8 %)	1,559 (−96.9 %)	0,218 (−77.8 %)	−0,085	−0,231	−0,107	0,722	−0,046	29	0	8	6	3
	125	0,029	0,184 (−84.4 %)	0,135 (−78.8 %)	1,633 (−98.2 %)	0,135 (−78.7 %)	−0,048	−0,196	−0,073	0,785	−0,075	61	0	15	12	9
	175	0,031	0,178 (−82.3 %)	0,131 (−76.0 %)	1,696 (−98.1 %)	0,095 (−67.5 %)	−0,048	−0,183	−0,069	0,785	−0,103	522	0	22	18	9
	250	0,025	0,173 (−85.3 %)	0,127 (−80.0 %)	2,305 (−98.9 %)	0,081 (−68.6 %)	−0,031	−0,168	−0,059	0,953	−0,108	1486	0	43	35	18

Table 11
Aggregated results by number of DMUs on the true frontier (Fare et al., 1994).

border	Not noise				Noise			
	DEA	BDEA	CEAT	StoNED	DEA	BDEA	CEAT	StoNED
0 %	−66.2 %	−52.4 %	−89.2 %	−61.3 %	−68.3 %	−56.7 %	−90.2 %	−65.5 %
5 %	−72.2 %	−61.9 %	−92.6 %	−66.2 %	−72.6 %	−62.6 %	−93.2 %	−70.2 %
10 %	−74.2 %	−64.4 %	−93.7 %	−71.7 %	−73.0 %	−63.8 %	−93.7 %	−65.6 %
20 %	−75.6 %	−67.4 %	−95.5 %	−73.0 %	−75.1 %	−66.8 %	−95.2 %	−71.1 %

Table 12
Aggregated results by sample size (Fare et al., 1994).

border	Not noise				Noise			
	DEA	BDEA	CEAT	StoNED	DEA	BDEA	CEAT	StoNED
25	−49.0 %	−32.3 %	−78.0 %		−46.2 %	−28.8 %	−77.6 %	
75	−72.2 %	−61.1 %	−94.5 %	−76.3 %	−74.3 %	−64.9 %	−95.0 %	−72.5 %
125	−79.8 %	−71.5 %	−96.3 %	−71.6 %	−81.5 %	−74.6 %	−97.2 %	−73.6 %
175	−77.4 %	−68.1 %	−96.9 %	−60.5 %	−77.9 %	−69.6 %	−97.3 %	−63.8 %
250	−81.9 %	−74.5 %	−98.2 %	−63.7 %	−81.3 %	−74.5 %	−98.4 %	−62.4 %

Table 13
Aggregated results by number of DMUs on the true frontier (Perelman and Santín, 2009).

border	Not noise				Noise			
	DEA	BDEA	CEAT	StoNED	DEA	BDEA	CEAT	StoNED
0 %	−40.2 %	+10.1 %	−96.6 %	−47.6 %	−20.1 %	+6.8 %	−96.0 %	−39.2 %
5 %	−35.2 %	+14.4 %	−97.5 %	−50.5 %	−9.6 %	+14.0 %	−96.3 %	−32.6 %
10 %	−23.7 %	+27.3 %	−97.6 %	−47.2 %	−8.6 %	+9.0 %	−97.0 %	−30.7 %
20 %	−13.9 %	−24.5 %	−98.2 %	−40.1 %	+1.1 %	+14.5 %	−97.2 %	−22.6 %

Table 14
Aggregated results by sample size (Perelman and Santín, 2009).

border	Not noise				Noise			
	DEA	BDEA	CEAT	StoNED	DEA	BDEA	CEAT	StoNED
50	−43.3 %	−9.3 %	−95.0 %	−43.8 %	−35.8 %	−6.9 %	−93.6 %	−39.2 %
100	−48.2 %	−14.4 %	−97.5 %	−47.0 %	−33.4 %	−6.2 %	−96.6 %	−39.9 %
150	−28.4 %	+20.5 %	−97.8 %	−45.0 %	−14.3 %	+9.2 %	−97.1 %	−31.5 %
200	−16.1 %	+42.2 %	−98.2 %	−44.0 %	+2.4 %	+19.4 %	−97.6 %	−26.2 %
300	−5.3 %	+56.5 %	−98.9 %	−51.9 %	+34.7 %	+39.9 %	−98.1 %	−19.5 %

Table 15
Efficiency measures obtained from the empirical example.

Bak	Output-oriented radial model			Input-oriented radial model			Directional Distance Function		
	ACES 1	ACES 2	DEA	ACES 1	ACES 2	DEA	ACES 1	ACES 2	DEA
Bank SinoPac	1.14	1.17	1.11	0.86	0.83	0.89	0.07	0.09	0.05
Bank of Kaohsiung	1.38	1.40	1.36	0.63	0.62	0.65	0.19	0.19	0.18
Bank of Panhsin	1.76	1.75	1.75	0.37	0.37	0.37	0.34	0.34	0.34
Bank of Taiwan	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
Cathay United Bank	1.40	1.41	1.38	0.69	0.69	0.71	0.17	0.18	0.17
Chang Hwa Bank	1.07	1.11	1.06	0.93	0.90	0.94	0.04	0.05	0.03
China Development	1.00	1.18	1.00	1.00	0.56	1.00	0.00	0.13	0.00
China Trust Bank	1.01	1.00	1.00	0.98	1.00	1.00	0.01	0.00	0.00
Cooperative Bank	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
Cosmos Bank	2.26	2.25	2.24	0.23	0.23	0.23	0.52	0.52	0.52
Cota Bank	1.75	1.81	1.64	0.48	0.48	0.49	0.33	0.34	0.28
E. Sun Bank	1.12	1.12	1.12	0.84	0.84	0.84	0.08	0.08	0.08
Entie Bank	1.45	1.55	1.19	0.66	0.61	0.81	0.20	0.24	0.10
Export-Import Bank	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
Far Eastern Bank	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
First Bank	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
Hua Nan Bank	1.06	1.08	1.05	0.94	0.91	0.94	0.03	0.04	0.03
Hwatai Bank	1.74	1.73	1.73	0.34	0.33	0.35	0.36	0.36	0.35
Industrial Bank of Taiwan	1.58	1.58	1.00	0.72	0.72	1.00	0.26	0.26	0.00
Jih Sun Bank	1.73	1.72	1.63	0.43	0.43	0.48	0.32	0.32	0.28
Land Bank	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
Mega Bank	1.01	1.03	1.00	0.98	0.96	1.00	0.01	0.02	0.00
Shin Kong Bank	1.36	1.35	1.35	0.70	0.70	0.70	0.16	0.16	0.16
Sunny Bank	1.45	1.44	1.44	0.59	0.60	0.60	0.22	0.21	0.21
Ta Chong Bank	1.25	1.31	1.14	0.74	0.72	0.86	0.13	0.15	0.07
Taichung Bank	1.33	1.36	1.29	0.70	0.68	0.73	0.16	0.17	0.14
Taipei Fubon Bank	1.00	1.04	1.00	1.00	0.93	1.00	0.00	0.03	0.00
Taishin Bank	1.21	1.25	1.19	0.80	0.76	0.82	0.10	0.12	0.09
Taiwan Business Bank	1.05	1.05	1.03	0.95	0.95	0.97	0.03	0.03	0.02
The Shanghai Bank	1.15	1.18	1.11	0.84	0.80	0.89	0.07	0.09	0.06
Union Bank	1.71	1.70	1.69	0.49	0.50	0.50	0.30	0.29	0.29
Mean	1.29	1.31	1.24	0.77	0.75	0.80	0.13	0.14	0.11
Median	1.15	1.18	1.11	0.83	0.76	0.89	0.08	0.12	0.06

reduce the error, thus increasing the computational burden.

5.3. Comparative analysis

In this final section on computational experiments, we present a comparative analysis of the performance of ACES, DEA, StoNED, CEAT, and BDEA across the previously described scenarios. For the ACES model, and following the guidance provided in Tables 4 and 5, the results are reported using the following configurations: Approach 1 for sample sizes of 50 and 75; Approach 3 for sample sizes of 100 and 125; and Approach 4 for sample sizes of 150, 175, 200, 250, and 300. Accordingly, Tables 7 and 8 present the results corresponding to the Perelman and Santín (2009) design, while Tables 9 and 10 show those obtained under the framework of Fare et al. (1994).

The first two columns of Tables 7, 8, 9, and 10 indicate the proportion of DMUs located on the true frontier and the total number of DMUs evaluated in each scenario. The remaining columns compare the performance of the different methodologies in terms of estimation error, bias, and computation time. All percentage differences are expressed with respect to the competing methods. In this way, a negative value denotes that ACES outperforms the corresponding method (i.e., achieves lower error), while a positive value indicates a performance deficit. Finally, note that results for Fare et al. (1994) and StoNED with a sample size of 25 are not reported, as they led to abnormally large values that distorted the true performance of the methods and would have undermined the comparability of the results.

We now detail the performance differences observed across the two experimental settings: Perelman and Santín (2009) and Fare et al. (1994). In the latter, the results are particularly compelling. ACES outperforms all competing methods in every scenario tested—regardless of noise levels or the proportion of efficient units. Error reductions related to CEAT exceed 90 % in nearly all cases, while improvements over DEA, BDEA, and StoNED range from 19.6 % to 85.3 %. Overall, no systematic performance differences are observed between noisy and noise-free settings; however, a clear trend emerges whereby the relative advantage of ACES increases with both sample size and the proportion of DMUs located on the true frontier. Tables 11 and 12 summarize these results.

In the case of the Perelman and Santín (2009) scenarios, ACES maintains strong performance relative to most benchmark methods. Notably, it consistently achieves error reductions exceeding 90 % when compared to CEAT. ACES also outperforms StoNED across all configurations—regardless of noise levels, sample size, or the proportion of efficient units—with a larger margin of improvement observed in noise-free settings. In the presence of stochastic noise, however, the performance gap between StoNED and ACES narrows as both the sample size and the number of units on the true frontier increase, suggesting that StoNED benefits more from larger, noise-contaminated datasets where its stochastic structure becomes more effective. A similar pattern is observed when comparing ACES to standard DEA. In both noisy and noise-free settings, the performance of the two methods converges as the sample size and the proportion of efficient units increase. In particular, under noise-free conditions, DEA slightly outperforms ACES in scenarios with a sample size of 200 or more and when 20 % of the units lie on the true frontier. This suggests that DEA may benefit from its nonparametric envelopment structure in large, well-populated datasets where the frontier is densely represented. Finally, BDEA consistently delivers superior results in this class of scenarios. Its advantage over ACES becomes more pronounced as the sample size increases and a larger proportion of DMUs lie on the true frontier. These findings reinforce the strengths of bootstrap-based bias correction in well-populated and frontier-dense datasets, where resampling techniques can more effectively capture the underlying efficiency structure. Tables 13 and 14 summarize these results.

As a concluding remark, it is important to acknowledge two key limitations of the ACES methodology, both of which are closely interrelated: computational cost and hyperparameter tuning. Although ACES demonstrates robust empirical performance, its computational burden grows rapidly with the number of DMUs. As evidenced in Tables 7–10, the runtime increases at an exponential rate as the sample size expands. This effect becomes critical when the number of DMUs exceeds 300, even under moderate dimensionality (e.g., four input variables and second-degree interactions). The main computational bottleneck arises from the number of times model (26) must be solved during the forward selection stage, which is directly driven by the size of the candidate BF set defined in (6). Consequently, a promising direction for future research would be to design more efficient strategies to intelligently reduce the size of the candidate set—focusing only on potentially viable BFs—thereby improving scalability without compromising estimation quality. In addition, the high computational cost hinders thorough exploration of the hyperparameter space. Parameters such as the error reduction threshold, or the maximum degree of interaction play a critical role in balancing model flexibility and overfitting risk. However, the ability to systematically evaluate different configurations through cross-validation is severely limited by runtime constraints, particularly in large-scale settings. As such, the development of faster heuristics or adaptive tuning strategies would be essential to unlock the full potential of ACES in practice.

6. An empirical illustration

In this section, we apply a real dataset to demonstrate the performance of various technical efficiency measures using ACES. The dataset includes information on 31 Taiwanese banks for the year 2010, previously analyzed by Juo et al. (2015). The inputs considered are financial FUNDS (x_1), LABOR (x_2), and physical CAPITAL (x_3), while the outputs are financial INVESTMENTS (y_1) and LOANS (y_2). All monetary variables are measured in million TWD, with labor measured as the number of employees. To improve the numerical stability of the algorithm, all monetary variables were rescaled by dividing them by 1,000. A detailed discussion of the statistical sources and variable definitions is available in Juo et al. (2015).

Regarding the ACES configuration, we tuned three key hyperparameters: the error reduction threshold $\xi \in \{0.005, 0.01\}$, the required improvement of a 3-degree and 2-degree BF over the best 1-degree candidate $\xi^{(2)} = \xi^{(3)} \in \{0, 0.05, 0.10\}$; and the penalty per knot $d \in \{1, 2\}$. Monotonicity and concavity are both imposed in the initial stage due to the sample size lower than 50 DMUs. The two best results were obtained by the following configurations: $\xi = 0.005$, $\xi^{(2)} = \xi^{(3)} = 0.10$ and $d = 1$ (ACES 1) and (ii) $\xi = 0.010$, $\xi^{(2)} = \xi^{(3)} = 0.05$ and $d = 1$ (ACES 2).

Table 15 presents the results for different efficiency measures. The first column lists the assessed bank. The next three blocks, each with three columns, correspond to the efficiency models used in our study: the output-oriented radial model (30), the input-oriented radial model (31) and the Directional Distance Function (32). For the DDF, the directional vector $(\boldsymbol{G}_x, \boldsymbol{G}_y) = (x_{01}, x_{02}, x_{03}, y_{01}, y_{02})$ is used to evaluate each DMU. Finally, each column in these blocks represents a different approach: DEA, ACES 1 and ACES 2.

Table 7 demonstrates that DEA consistently produces more optimistic efficiency assessments across all evaluated cases. This outcome is primarily due to DEA's omission of the minimum extrapolation principle, which typically positions the production frontier as close as possible to the observed data points. Moreover, the results reveal that different configurations of ACES (ACES 1 and ACES 2) lead to notably different efficiency evaluations. For example, under the radial output approach, the mean score in ACES 1 is 1.29, in ACES 2 is 1.31, while in DEA it is 1.24. Consequently, the density of ACES scores tends to decrease around 1, indicating fewer units at the efficiency threshold, while it increases throughout the rest of the distribution, reflecting a more realistic estimation of efficiency.

7. Conclusions

This paper introduces the Adaptive Constrained Enveloping Splines (ACES) as an innovative approach to enhancing the accuracy of efficiency analysis in multi-output and multi-input production settings. Building on the foundational work of España et al. (2024), ACES advances this framework by implementing a three-stage process that delivers a more realistic and robust estimation of the production frontier compared to conventional methods, at least following out configurations of simulated scenarios.

In particular, our computational experiments show that ACES consistently outperforms DEA, StoNED, and CEAT across a variety of simulated scenarios, especially in terms of mean squared error and bias reduction. This advantage is robust across different sample sizes and noise conditions. Regarding Bootstrapped DEA, ACES only shows a clear improvement in the scenarios based on the design proposed by Fare et al. (1994), where the production technology exhibits varying returns to scale. These results suggest that while bootstrapping enhances the inferential capabilities of DEA, ACES offers a more accurate estimation in complex or heterogeneous production environments.

This study also provides guidance on configuring ACES to achieve optimal performance. While most hyperparameters can be tuned using k-fold cross-validation, as is common in Machine Learning, specific recommendations for shaping the estimator during the first stage of the method have been detailed. The results indicate that for small sample sizes (e.g., 50 units or fewer), it is advantageous to impose both monotonicity and concavity constraints to enhance model accuracy. As the sample size increases to 100 units or more, applying only one of these constraints suffices. For even larger samples (150 units or more), the best results are obtained by relying exclusively on the envelopment conditions. Furthermore, this paper includes an empirical case study demonstrating how to apply various efficiency measures using an ACES model.

In conclusion, ACES offers a significant advancement in the field of efficiency analysis. The method's ability to integrate shape constraints and handle noisy data makes it particularly valuable in real-world applications where sample sizes and data quality can vary. While ACES requires more computational resources than alternative approaches, the trade-off is justified by the substantial improvements in accuracy and robustness, making it a valuable tool for researchers and practitioners in efficiency analysis.

CRediT authorship contribution statement

Victor J. España: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Juan Aparicio:** Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Xavier Barber:** Writing – original draft, Validation, Supervision, Methodology, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

V. J. España y J. Aparicio thank the grant PID2022-136383NB-I00 funded by MICIU/AEI/ 10.13039/501100011033 and by ERDF/EU. This research was also partially supported by the CIPROM/2024/34 grant, funded by the Conselleria de Educación, Cultura, Universidades y Empleo, Generalitat Valenciana. V. J. España is grateful for the financial support from the Generalitat Valenciana under Grant ACIF/2021/135. Finally, X. Barber thanks the grant PID2022-136455NB-I00 funded by Ministerio de Ciencia e Innovación / Agencia Estatal de Investigación.

Data availability

Data will be made available on request.

References

Adler, N., Golany, B., 2007. PCA-DEA: reducing the curse of dimensionality. Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis 139–153.
Afriat, S.N., 1972. Efficiency estimation of production functions. Int. Econ. Rev. 568–598.
Aigner, D., Lovell, C.K., Schmidt, P., 1977. Formulation and estimation of stochastic frontier production function models. J. Econ. 6 (1), 21–37.
Aragon, Y., Daouia, A., Thomas-Agnan, C., 2005. Nonparametric Frontier Estimation: a Conditional Quantile-based Approach. Economet. Theor. 21 (2), 358–389.
Banker, R.D., 1988. Stochastic data envelopment analysis. Data Envelopment Analysis Journal 5 (2), 281–309.
Banker, R.D., Maindiratta, A., 1992. Maximum likelihood estimation of monotone and concave production frontiers. J. Prod. Anal. 3 (4), 401–415.
Banker, R.D., Charnes, A., Cooper, W.W., 1984. Some Models for estimating Technical and Scale Inefficiencies in Data Envelopment Analysis. Manag. Sci. 30 (9), 1078–1092.
Bogetoft, P., Otto, L., & Otto, M. L. (2015). Package 'benchmarking'. *Retrieved January*, *4*, 2016.
Breiman, L., 1993. Hinging hyperplanes for regression, classification, and function approximation. IEEE Trans. Inf. Theory 39 (3), 999–1013.
Breiman, L., 2001. Random Forests. Machine Learning 45, 5–32.
Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and regression trees. CRC Press.
Chambers, R.G., Chung, Y., Färe, R., 1998. Profit, directional distance functions, and Nerlovian efficiency. J. Optim. Theory Appl. 98, 351–364.
Charles, V., Aparicio, J., Zhu, J., 2019. The curse of dimensionality of decision-making units: a simple approach to increase the discriminatory power of data envelopment analysis. Eur. J. Oper. Res. 279 (3), 929–940.
Charnes, A., Cooper, W.W., Rhodes, E., 1978. Measuring the efficiency of decision making units. Eur. J. Oper. Res. 2 (6), 429–444.
Chen, Y., Tsionas, M.G., Zelenyuk, V., 2021. LASSO+ DEA for small and big wide data. Omega 102, 102419.
Cooper, W.W., Park, K.S., Pastor, J.T., 1999. RAM: a range adjusted measure of inefficiency for use with additive models, and relations to other models and measures in DEA. J. Prod. Anal. 11, 5–42.
Cooper, W.W., Pastor, J.T., Borras, F., Aparicio, J., Pastor, D., 2011. BAM: a bounded adjusted measure of efficiency for use with bounded additive models. J. Prod. Anal. 35, 85–94.
Dai, S., Fang, Y. H., Lee, C. Y., & Kuosmanen, T. (2021). pyStoNED: A Python package for convex regression and frontier estimation. arXiv preprint arXiv:2109.12962.
Daouia, A., Simar, L., 2007. Nonparametric efficiency analysis: a multivariate conditional quantile approach. J. Econ. 140 (2), 375–400.
Daouia, A., Noh, H., Park, B.U., 2016. Data envelope fitting with constrained polynomial splines. J. R. Stat. Soc. Ser. B Stat Methodol. 3–30.
Deprins, D., Simar, L., Tulkens, H., 1984. Measuring labor-efficiency in post offices. Public Goods, Environmental Externalities and Fiscal Competition 285–309.
Diewert, W.E., Parkan, C., 1983. *Linear programming tests of regularity conditions for production functions*. Physica-Verlag HD, pp. 131–158.
Drucker, H., Burges, C.J., Kaufman, L., Smola, A., Vapnik, V., 1997. Support vector regression machines. Adv. Neural Inf. Proces. Syst. 9, 155–161.
España, V.J., Aparicio, J., Barber, X., Esteve, M., 2024. Estimating production functions through additive models based on regression splines. Eur. J. Oper. Res. 312 (2), 684–699.
Esteve, M., Aparicio, J., Rabasa, A., Rodriguez-Sala, J.J., 2020. Efficiency analysis trees: a new methodology for estimating production frontiers through decision trees. Expert Syst. Appl. 162, 113783.
Esteve, M., Aparicio, J., Rodriguez-Sala, J.J., Zhu, J., 2023. Random Forests and the measurement of super-efficiency in the context of Free Disposal Hull. Eur. J. Oper. Res. 304 (2), 729–744.
Esteve, M., España, V., Aparicio, J., Barber, X., 2022. eat: an R Package for fitting Efficiency Analysis Trees. R J. 14 (3), 249–281.
Färe, R., Lovell, C.K., 1978. Measuring the technical efficiency of production. J. Econ. Theory 19 (1), 150–162.
Färe, R., Grosskopf, S., Lovell, C.K., 1985. The Measurement of Efficiency of Production, Vol. 6. Springer Science & Business Media.
Fare, R., Grosskopf, S., Lovell, C.K., 1994. *Production frontiers*. Cambridge University Press.
Friedman, J.H., 1991. Multivariate adaptive regression splines. Ann. Stat. 1–67.
Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 1189–1232.
Golub, G.H., Heath, M., Wahba, G., 1979. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21 (2), 215–223.
Guerrero, N.M., Aparicio, J., Valero-Carreras, D., 2022. Combining Data Envelopment Analysis and Machine Learning. Mathematics 10 (6), 909.

Guillen, M.D., Aparicio, J., Esteve, M., 2023. Gradient tree boosting and the estimation of production frontiers. Expert Syst. Appl. 214, 119134.
Juo, J.C., Fu, T.T., Yu, M.M., Lin, Y.H., 2015. Profit-oriented productivity change. Omega 57, 176–187.
Korostelev, A.P., Simar, L., Tsybakov, A.B., 1995. Efficient Estimation of Monotone Boundaries. Ann. Stat. 476–489.
Kuosmanen, T., Johnson, A., 2010. Data Envelopment Analysis as Nonparametric Least-Squares Regression. Oper. Res. 58 (1), 149–160.
Kuosmanen, T., Johnson, A., 2017. Modeling joint production of multiple outputs in StoNED: Directional distance function approach. Eur. J. Oper. Res. 262 (2), 792–801.
Kuosmanen, T., Kortelainen, M., 2012. Stochastic non-smooth envelopment of data: semi-parametric frontier estimation subject to shape constraints. J. Prod. Anal. 38, 11–28.
Kuosmanen, T., Johnson, A., Saastamoinen, A., 2015. Stochastic nonparametric approach to efficiency analysis: a unified framework. A Handbook of Models and Methods, Data Envelopment Analysis, pp. 191–244.
Lovell, C.K., Pastor, J.T., 1995. Units invariant and translation invariant DEA models. Oper. Res. Lett. 18 (3), 147–151.
Martinez, D.L., Shih, D.T., Chen, V.C., Kim, S.B., 2015. A convex version of multivariate adaptive regression splines. Comput. Stat. Data Anal. 81, 89–106.
Meeusen, W., van Den Broeck, J., 1977. Efficiency estimation from Cobb-Douglas production functions with composed error. Int. Econ. Rev. 435–444.
Milborrow, S., 2014. Notes on the earth package. Retrieved October, 31,. 2017.
Milborrow, S., 2023. earth: Multivariate Adaptive Regression Spline Models. R Package Version 5 (3), 2.
Molinos-Senante, M., Maziotis, A., Sala-Garrido, R., Mocholi-Arce, M., 2023. Assessing the influence of environmental variables on the performance of water companies: an efficiency analysis tree approach. Expert Syst. Appl. 212, 118844.
Olesen, O.B., Ruggiero, J., 2018. An improved Afriat–Diewert–Parkan nonparametric production function estimator. Eur. J. Oper. Res. 264 (3), 1172–1188.
Olesen, O.B., Ruggiero, J., 2022. The hinging hyperplanes: an alternative nonparametric representation of a production function. Eur. J. Oper. Res. 296 (1), 254–266.
Parmeter, C.F., Racine, J.S., 2013. Smooth constrained frontier analysis. In: Recent Advances and Future Directions in Causality, Prediction and Specification Analysis. Springer, pp. 463–488.
Pastor, J.T., Lovell, C.K., Aparicio, J., 2012. Families of linear efficiency programs based on Debreu's loss function. J. Prod. Anal. 38, 109–120.
Pastor, J.T., Ruiz, J.L., Sirvent, I., 1999. An enhanced DEA Russell graph efficiency measure. Eur. J. Oper. Res. 115 (3), 596–607.
Perelman, S., Santín, D., 2009. How to generate regularly behaved production data? a Monte Carlo experimentation on DEA scale efficiency measurement. Eur. J. Oper. Res. 199 (1), 303–310.
Shih, D.T., Chen, V.C.P., Kim, S.B., 2006. Convex version of multivariate adaptive regression splines. In Proceedings of the 2006 Industrial Engineering Research Conference.
Simar, L., Wilson, P.W., 1998. Sensitivity analysis of efficiency scores: how to bootstrap in nonparametric frontier models. Manag. Sci. 44 (1), 49–61.
Simar, L., Wilson, P.W., 2000. A general methodology for bootstrapping in non-parametric frontier models. J. Appl. Stat. 27 (6), 779–802.
Mosek, A. P. S. (2021). The MOSEK optimization toolbox for MATLAB manual. Version 9.0., 2019. URL http://docs. mosek. com/9.0/toolbox/index. html.
Theussl, S. & Hornik, K. (2019). Rglpk: R/GNU Linear Programming Kit Interface. R package version 0.6-4. https://CRAN.R-project.org/package=Rglpk.
Tsai, J.C., Chen, V.C., 2005. Flexible and robust implementations of multivariate adaptive regression splines within a wastewater treatment stochastic dynamic program. Qual. Reliab. Eng. Int. 21 (7), 689–699.
Tsionas, M.G., 2022. Efficiency estimation using probabilistic regression trees with an application to Chilean manufacturing industries. Int. J. Prod. Econ. 108492.
Valero-Carreras, D., Aparicio, J., Guerrero, N.M., 2021. Support Vector Frontiers: a New Approach for estimating Production Functions through support Vector Machines. Omega 104, 102490.
Valero-Carreras, D., Aparicio, J., Guerrero, N.M., 2022. Multi-output support Vector Frontiers. Comput. Oper. Res. 143, 105765.
Wilson, P.W., 2018. Dimension reduction in nonparametric models of production. Eur. J. Oper. Res. 267 (1), 349–367.
Zhang, H., 1994. Maximal correlation and adaptive splines. Technometrics 36 (2), 196–201.