

Second-order conic programming for data envelopment analysis models

Nahia Mourad

Computer Science, American University in the Emirates, United Arab Emirates

ABSTRACT

Data envelopment analysis (DEA) is a widely used benchmarking technique. Its strength stems from the fact that it can include several inputs and outputs of not necessarily the same type to evaluate efficiency scores. Indeed, the aforesaid method is based on mathematical optimization. This paper constructs a second-order conic optimization problem unifying several DEA models. Moreover, it presents an algorithm that solves the former problem, and provides a MATLAB function associated with it. As far as known, no MATLAB function solves DEA models. Among different types of DEA, this function can handle deterministic, Malmquist index, and stochastic models. In fact, DEA is involved in various practical applications, thus, this work will provide some possible future extensions, not only for MATLAB but also for any programming software in applications of decision science and efficiency analysis.

Keywords: Data envelopment analysis, Efficiency, Malmquist DEA, Stochastic DEA, MATLAB functions, Numerical simulations, Mathematical models

Corresponding Author:

Nahia Mourad
Computer Science
American University in the Emirates
United Arab Emirates
E-mail: nahia.mourad@aue.ae

1. Introduction

Data envelopment analysis (DEA) is a nonparametric mathematical tool for productive efficiency evaluation. This method produces what's known as the efficiency frontier based on the given data related to inputs and outputs of the decision-making units (DMUs) under study. Efficient DMUs are lying on what is known as "best-practice-frontier" [1], [2]. Its strength arises from not only being nonparametric with no predefined assumptions imposed on it but also because of the simplicity of the model. Even though it was initiated by maximizing the production efficiency ratio, it turns out to be a linear constrained optimization problem in absence of stochastic variables. This method is exponentially increasingly used in diversified fields [3], [4]. The main contribution of this method is in operations research and management science. Dutta et al. provided a literature review of DEA applications in supplier evaluation and selection [5]. Among the various application fields, DEA is extensively applied in various energy sectors like renewable energy and wastewater treatment plants e.g. [6]–[8], healthcare, where lately it was used for evaluating efficiencies of healthcare systems during COVID-19 pandemic e.g. [9]–[11], economic and Banking e.g. [12], [13], agriculture e.g. [14]–[16], education [17], [18], etc. Several software systems implement DEA models to collect information about relative productivity. For example, DEAP is an open-source program established by Tim Coelli. This program can calculate technical, cost efficiencies and Malmquist total factor productivity indices with the ability to handle different constraints on returns to scale in different orientations [19]. MDeap2, Open Source DEA, EMS, MaxDEA, GAMS and PIM-DEA are all available programs that calculate various DEA-related scores. MATLAB is a programming platform developed by MathWorks in 1994 [20]. It is widely used by scientists for data analysis, model creation and algorithm development. It comes out that DEA models are not implemented in this platform.

This paper provides an algorithm and an open-source MATLAB code, which can be transformed into any programming language, for implementing different DEA models and it is organized as follows: In Section 2, a

unified DEA model is constructed. Then, in Section 3, the algorithm to solve this last model together with the discretization are detailed. After, an illustrative example is given in Section 4. Finally, Section 5 concludes the paper.

2. DEA Models

Data envelopment analysis is a widely used efficiency evaluation method. It calculates efficiency scores for decision-making units (DMUs). The idea arises from the fact that the efficiency score is the ratio of outputs to inputs. Indeed, as these later variables are mostly different in nature, a linear combination is used to unify the unit for these variables. That is, for N DMUs to be evaluated based on m_I inputs denoted by $(x_{in})_{1 \leq i \leq m_I, 1 \leq n \leq N}$ and m_O outputs denoted by $(y_{jn})_{1 \leq j \leq m_O, 1 \leq n \leq N}$, the relative efficiency score for DMU p is then

$$e_p = \max_{(\beta, \nu) \in \mathbb{R}_+^{m_I} \times \mathbb{R}_+^{m_O}} \frac{\sum_{j=1}^{m_O} \nu_j y_{jp}}{\sum_{i=1}^{m_I} \beta_i x_{ip}}, \quad \text{where} \quad \frac{\sum_{j=1}^{m_O} \nu_j y_{jp}}{\sum_{i=1}^{m_I} \beta_i x_{ip}} \leq 1 \quad \text{for} \quad p = 1, \dots, N.$$

The real numbers β_i and ν_j are the weights associated with the inputs x_{in} and outputs y_{jn} , respectively.

Given the abovementioned, Charnes et al. [1] introduced the constant return to scale model (CCR), by linearizing the above fractional programming, with two different orientations, as follows

Input-oriented model:

$$\begin{aligned} & \max_{(\beta, \nu) \in \mathbb{R}_+^{m_I} \times \mathbb{R}_+^{m_O}} \sum_{j=1}^{m_O} \nu_j y_{jp} \\ & \sum_{i=1}^{m_I} \beta_i x_{ip} = 1 \\ & \sum_{j=1}^{m_O} \nu_j y_{jp} - \sum_{i=1}^{m_I} \beta_i x_{ip} \leq 0 \end{aligned}$$

Output-oriented model

$$\begin{aligned} & \min_{(\beta, \nu) \in \mathbb{R}_+^{m_I} \times \mathbb{R}_+^{m_O}} \sum_{i=1}^{m_I} \beta_i x_{ip} \\ & \sum_{j=1}^{m_O} \nu_j y_{jp} = 1 \\ & \sum_{j=1}^{m_O} \nu_j y_{jp} - \sum_{i=1}^{m_I} \beta_i x_{ip} \leq 0 \end{aligned}$$

These optimization problems, whether input or output-oriented, are known by the multiplier form. In practice the dual of these models is solved. The dual forms are known by the envelopment forms [21] and are given as follows

Input-oriented envelopment form model:

$$\begin{aligned} e_p &= \min_{\lambda \in \mathbb{R}_+^N} (\theta_p) \\ \sum_{n=1}^N \lambda_n x_{in} &\leq \theta_p x_{ip}, \quad i = 1, \dots, m_I \\ \sum_{n=1}^N \lambda_n y_{jn} &\geq y_{jp}, \quad j = 1, \dots, m_O \end{aligned}$$

Output-oriented envelopment form model:

$$\begin{aligned} \frac{1}{e_p} &= \max_{\lambda \in \mathbb{R}_+^N} (\theta_p) \\ \sum_{n=1}^N \lambda_n x_{in} &\leq x_{ip}, \quad i = 1, \dots, m_I \\ \sum_{n=1}^N \lambda_n y_{jn} &\geq \theta_p y_{jp}, \quad j = 1, \dots, m_O \end{aligned}$$

where λ_n and e_n are the weight and the efficiency score for the n -th DMU, respectively. A DMU with an efficiency score equal to 100% is relatively efficient, otherwise, it is inefficient. It is to be noted that the CRS input and output-oriented models produce the same efficiency score.

Following, in 1984, Banker Charnes and Cooper modified the CCR model and proposed what is known as the variable returns to scale model (VRS). It is worth mentioning that CCR and CRS refer to the same model which is the constant return to scale created by Charnes, Cooper, and Rhodes. This type of model reflects that the

change of output(s) relative to input(s) is constant and hence the efficiency score is not substantially related to the DMU size. Whereas the BCC named after Banker Charnes and Cooper is also labeled by VRS. This last waive the mentioned condition and, therefore, any variation in either input(s) or output(s) does not necessarily produce a proportional change in the other [22], [23].

It is worth mentioning that the choice of the orientation depends on the ability in increasing any of the outputs or decreasing any of the inputs. The input-oriented model minimizes the input at the given output level, whereas the output-oriented one maximizes the output at the given input level. The VRS model differs from the CRS model only by adding the below extra constraints on the weights associated with the DMUs

$$\sum_{n=1}^N \lambda_n = 1$$

Due to the embedding of this extra constraint, it can be concluded that the VRS efficiency scores are larger than those obtained from the CRS model.

After that, the stochastic data envelopment analysis model was initiated. It allows any variable, whether input or output, to be stochastic, i.e., it can be a random variable that follows a probability distribution [24]. The required constraint is preserved by enforcing the probability value of this constraint to be almost one. This is the so-called chance-constrained programming (CCP), more details can be found in [24], [25].

Define \mathcal{J}_S (resp. \mathcal{O}_S) to be the set of indices i (resp. r) for which $(x_{in})_{1 \leq n \leq N} \in \mathcal{S}^N$ (resp. $(y_{rn})_{1 \leq n \leq N} \in \mathcal{S}^N$), where \mathcal{S} is the set of random variables. On the other hand, let \mathcal{J}_D (resp. \mathcal{O}_D) collects the indices i (resp. j) for which $(x_{in})_{1 \leq n \leq N} \in \mathbb{R}^N$ (resp. $(y_{rn})_{1 \leq n \leq N} \in \mathbb{R}^N$) is a vector of deterministic inputs (resp. outputs). Then,

$$\mathcal{J}_D \cup \mathcal{J}_S = \{1, \dots, m_I\} \text{ and } \mathcal{O}_D \cup \mathcal{O}_S = \{1, \dots, m_O\}.$$

The envelopment form of the stochastic DEA model is then given by

Input-oriented model:

$$e_p = \min_{\lambda \in \mathbb{R}_+^N} (\theta_p)$$

$$P \left\{ \sum_{n=1}^N \lambda_n x_{in} \leq \theta_p x_{ip} \right\} \geq 1 - \epsilon, \quad i \in \mathcal{J}_S$$

$$\sum_{n=1}^N \lambda_n x_{in} \leq \theta_p x_{ip}, \quad i \in \mathcal{J}_D$$

$$P \left\{ \sum_{n=1}^N \lambda_n y_{jn} \geq y_{jp} \right\} \geq 1 - \epsilon, \quad j \in \mathcal{O}_S$$

$$\sum_{n=1}^N \lambda_n y_{jn} \geq y_{jp}, \quad j \in \mathcal{O}_D$$

$$\lambda_n \geq 0, \quad 1 \leq n \leq N$$

VRS constrained: $\sum_{n=1}^N \lambda_n = 1,$

Output-oriented model:

$$\frac{1}{e_p} = \max_{\lambda \in \mathbb{R}_+^N} (\theta_p)$$

$$P \left\{ \sum_{n=1}^N \lambda_n x_{in} \leq x_{ip} \right\} \geq 1 - \epsilon, \quad i \in \mathcal{J}_S$$

$$\sum_{n=1}^N \lambda_n x_{in} \leq x_{ip}, \quad i \in \mathcal{J}_D$$

$$P \left\{ \sum_{n=1}^N \lambda_n y_{jn} \geq \theta_p y_{jp} \right\} \geq 1 - \epsilon, \quad j \in \mathcal{O}_S$$

$$\sum_{n=1}^N \lambda_n y_{jn} \geq \theta_p y_{jp}, \quad j \in \mathcal{O}_D$$

$$\lambda_n \geq 0, \quad 1 \leq n \leq N$$

VRS constrained: $\sum_{n=1}^N \lambda_n = 1,$

where $\epsilon \in [0,1)$ is a small prescribed real number, and $e_p \in [0,1]$ is the relative efficiency score for the p -th DMU.

A unified DEA model can be written as follows:

$$\begin{aligned}
 & \min_{\lambda \in \mathbb{R}_+^N} ((-1)^{\delta_{0I}} \theta_p) \\
 & P \left\{ \sum_{n=1}^N \lambda_n x_{in} \leq \delta_{1I} \theta_p x_{ip} + \delta_{0I} x_{ip} \right\} \geq 1 - \epsilon, \quad i \in \mathcal{J}_S \\
 & \sum_{n=1}^N \lambda_n x_{in} \leq \delta_{1I} \theta_p x_{ip} + \delta_{0I} x_{ip}, \quad i \in \mathcal{J}_D \\
 & P \left\{ \sum_{n=1}^N \lambda_n y_{jn} \geq \delta_{1I} y_{jp} + \delta_{0I} \theta_p y_{jp} \right\} \geq 1 - \epsilon, \quad j \in \mathcal{J}_S \\
 & \sum_{n=1}^N \lambda_n y_{jn} \geq \delta_{1I} y_{jp} + \delta_{0I} \theta_p y_{jp}, \quad j \in \mathcal{J}_D \\
 & \lambda_n \geq 0, \quad 1 \leq n \leq N \\
 & \delta_{1V} \left(\sum_{n=1}^N \lambda_n - 1 \right) = 0
 \end{aligned} \tag{M-1}$$

where δ is the Kronecker delta, $I = 1$ (resp. $I = 0$) for the input (resp. output) oriented model, and $V = 1$ (resp. $V = 0$) for VRS (resp. CRS) model.

2.1. Malmquist DEA

The Malmquist index (MI) DEA analyses the efficiency of the DMUs over several periods of time. In fact, it measures the change in the productivity index. For each period t , and each DMU n , let

$$X_n^t = (x_{in}^t)_{1 \leq i \leq m_I} \text{ and } Y_n^t = (y_{jn}^t)_{1 \leq j \leq m_O}$$

be the vectors containing the values of inputs and outputs, respectively. Indeed, as elaborated by Lin & Fei [26] and Cooper et al. [22], the MI-DEA calculates the total factor productivity change over time (tfpch), and can be computed by

$$MI_n = \frac{\delta^2(X_n^2, Y_n^2)}{\delta^1(X_n^1, Y_n^1)} \times \sqrt{\frac{\delta^1(X_n^1, Y_n^1) \times \delta^1(X_n^2, Y_n^2)}{\delta^2(X_n^1, Y_n^1) \times \delta^2(X_n^2, Y_n^2)}} = \sqrt{\frac{\delta^1(X_n^2, Y_n^2) \times \delta^1(X_n^1, Y_n^1)}{\delta^1(X_n^1, Y_n^1) \times \delta^1(X_n^2, Y_n^2)}} \tag{1}$$

where $\delta^s(X_n^t, Y_n^t)$ represents the efficiency score of the n -th DMU observed in period s measured by frontier technology t [27], [28]. It is to be noted that, the values obtained from the MI-DEA model show the variation in productivity among two terms of production: the ‘‘Catch-up’’ and the ‘‘Frontier-shift’’. The former identifies the effect of growth in a DMU and is calculated by $\delta^2(X_n^2, Y_n^2)/\delta^1(X_n^1, Y_n^1)$, while the last verifies the change in the efficient frontiers and is equal to the square root of $(\delta^1(X_n^1, Y_n^1) \delta^1(X_n^2, Y_n^2))/(\delta^2(X_n^1, Y_n^1) \delta^2(X_n^2, Y_n^2))$.

Accordingly, the output and input-oriented radial MI-DEA models are

Input-oriented MI-DEA model:

$$\begin{aligned}
 \delta^s(X_p^t, Y_p^t) &= \min_{\lambda \in \mathbb{R}_+^N} (\theta_p) \\
 \sum_{n=1}^N \lambda_n x_{in}^s &\leq \theta_p x_{ip}^t, \quad i = 1, \dots, m_I
 \end{aligned}$$

Output-oriented MI-DEA model:

$$\begin{aligned}
 \frac{1}{\delta^s(X_p^t, Y_p^t)} &= \max_{\lambda \in \mathbb{R}_+^N} (\theta_p) \\
 \sum_{n=1}^N \lambda_n x_{in}^s &\leq x_{ip}^t, \quad i = 1, \dots, m_I
 \end{aligned}$$

$$\begin{aligned} \sum_{n=1}^N \lambda_n y_{jn}^s &\geq y_{jn}^t, & j = 1, \dots, m_o \\ \lambda_n &\geq 0, & 1 \leq n \leq N \end{aligned} \qquad \begin{aligned} \sum_{n=1}^N \lambda_n y_{jn}^s &\geq \theta_p y_{jp}^t, & j = 1, \dots, m_o \\ &= 1, \dots, m_o \\ \lambda_n &\geq 0, & 1 \leq n \leq N \end{aligned}$$

If the efficiency score $e_n^{(s,t)} = \delta_{1I} \delta^s(X_n^t, Y_n^t) + \delta_{0I} / \delta^s(X_n^t, Y_n^t)$ is identically one, then the DMU is said to be efficient in period s measured by frontier technology t , otherwise, it is inefficient. It can be observed that, for $t = s$, the $e_n^{(t,t)}$ is the (CCR) relative efficiency score for the n -th DMU in the period t .

Following the above methodology, a unified MI-DEA model is

$$\begin{aligned} &\min_{\lambda \in \mathbb{R}_+^N} ((-1)^{\delta_{0I}} \theta_p) \\ &\sum_{n=1}^N \lambda_n x_{in}^s \leq \delta_{1I} \theta_p x_{ip}^t + \delta_{0I} x_{ip}^t, \quad i = 1, \dots, m_I \\ &\sum_{n=1}^N \lambda_n y_{jn}^s \geq \delta_{1I} y_{jp}^t + \delta_{0I} \theta_p y_{jp}^t, \quad j = 1, \dots, m_o \\ &\lambda_n \geq 0, \quad 1 \leq n \leq N \\ &\delta_{1V} \left(\sum_{n=1}^N \lambda_n - 1 \right) = 0 \end{aligned} \tag{M-2}$$

It is also to be mentioned that the MI-DEA model is defined in the literature based on the CRS model only [28].

3. Discretization and algorithm

Multiple software systems implement the DEA model like DEAP, MDeap 2, Open Source DEA, EMS, and GAMS. None of the afore mentioned can handle all the above models at the same time. For example, DEAP is an open-source program that runs DEA and MI-DEA input/output-oriented and VRS/CRS models, with no prior installation, however, the stochastic DEA is not included [19]. In this section, an algorithm together with an open-source code [29], which solves any of the former DEA models, with no restrictions on the number of DMUs, are provided for MATLAB users. It is to be mentioned that the discretization and the obtained finite-dimensional optimization problem can be adapted to any programming language.

3.1. Discretization

Model (M-1) is a constrained optimization problem with equality linear constraints and inequality linear and nonlinear ones. Define the objective function $f(\lambda, \theta) = \theta$, and the matrices collecting the data associated with the deterministic input(s) and output(s)

$$\mathcal{M}_I = (x_{in})_{i \in I_D, n \in \{1, \dots, N\}} \text{ and } \mathcal{M}_O = (y_{jn})_{j \in O_D, n \in \{1, \dots, N\}} \tag{2}$$

The only existing equality constrained can be written using matrix multiplications as follows

$$A_{eq} \begin{bmatrix} \lambda \\ \theta \end{bmatrix} = b_{eq},$$

where

$$A_{eq} = \delta_{1V} [1 \dots 1 \quad 0] \text{ and } b_{eq} = \delta_{1V}. \tag{3}$$

While the inequality constraints are updated for each DMU p . Let

$$A_p = \begin{bmatrix} \mathcal{M}_j & -\delta_{1I}[\mathcal{M}_j]_{:p} \\ -\mathcal{M}_O & \delta_{0I}[\mathcal{M}_O]_{:p} \end{bmatrix} \quad (4)$$

where $[M]_{:p}$ is the p -th column in a matrix M , and

$$b_p = \begin{bmatrix} \delta_{0I}[\mathcal{M}_j]_{:p} \\ -\delta_{1I}[\mathcal{M}_O]_{:p} \end{bmatrix}. \quad (5)$$

Then the inequality constrained is given by

$$A_p \begin{bmatrix} \lambda \\ \theta \end{bmatrix} \leq b_p.$$

Indeed, the nonlinear constraints are a consequence of the chance constraints and are evaluated following the stochastic distribution variable (input or output). For $i \in \mathcal{J}_S$, $j \in \mathcal{O}_S$ and $p \in \{1, \dots, N\}$, let $x_{in} \sim \mathcal{N}(\mu_{in}, \sigma_{in})$ (resp. $y_{jn} \sim \mathcal{N}(\tilde{\mu}_{jn}, \tilde{\sigma}_{jn})$) for all $n \in \{1, \dots, N\}$. Then, the chance-constrained for normally distributed random variables

$$P \left\{ \sum_{n=1}^N \lambda_n x_{in} \leq \delta_{1I} \theta_p x_{ip} + \delta_{0I} x_{ip} \right\} \geq 1 - \epsilon$$

is equivalent to

$$\sum_{n=1}^N a_{np} \mu_{in} \leq e \sqrt{\sum_{n=1}^N a_{np}^2 \sigma_{in}^2 + 2 \sum_{k < n} a_{kp} a_{np} Cov(x_{in}, x_{ik})},$$

where

$$a_{np} = \lambda_n - \delta_{np} (\delta_{1I} \theta_p + \delta_{0I}),$$

and $\Phi_Z(e) = \epsilon$, Φ_Z is the cumulative density function for standard normal distribution.

While the chance-constrained of the output variable

$$P \left\{ \sum_{n=1}^N \lambda_n y_{jn} \geq \delta_{1I} y_{jp} + \delta_{0I} \theta_p y_{jp} \right\} \geq 1 - \epsilon$$

is equivalent to

$$\sum_{n=1}^N \tilde{a}_{np} \tilde{\mu}_{jn} \leq e \sqrt{\sum_{n=1}^N \tilde{a}_{np}^2 \tilde{\sigma}_{jn}^2 + 2 \sum_{k < n} \tilde{a}_{kp} \tilde{a}_{np} Cov(y_{jn}, y_{jk})},$$

where

$$\tilde{a}_{np} = -\lambda_n + \delta_{np} (\delta_{1I} + \delta_{0I} \theta_p).$$

Subsequently, define the nonlinear function

$$c(\lambda, \theta, p) = \begin{cases} \Lambda \mu_i - e \sqrt{\Lambda M_{cov}(i) \Lambda^T}, & i \in \mathcal{J}_S \\ \tilde{\Lambda} \tilde{\mu}_j - e \sqrt{\tilde{\Lambda} \tilde{M}_{cov}(j) \tilde{\Lambda}^T}, & j \in \mathcal{O}_S \end{cases} \quad (6)$$

where

$$\Lambda = \left(\lambda_n - \delta_{np} (\delta_{1I} \theta + \delta_{0I}) \right)_{1 \leq n \leq N}, \quad \tilde{\Lambda} = - \left(\lambda_n - \delta_{np} (\delta_{1I} + \delta_{0I} \theta) \right)_{1 \leq n \leq N}$$

$$\mu = (\mu_{in})_{i \in I_S, 1 \leq n \leq N}, \quad \tilde{\mu} = (\tilde{\mu}_{jn})_{j \in O_S, 1 \leq n \leq N'}$$

and

$$M_{cov}(i) = (Cov(x_{in}, x_{ik}))_{1 \leq n, k \leq N}, \quad \tilde{M}_{cov}(j) = (Cov(y_{jn}, y_{jk}))_{1 \leq n, k \leq N'}$$

Therefore, the optimization problem (M-1) is equivalent to the following second-order conic optimization problem

$$\begin{aligned} \min_{(\lambda, \theta) \in \mathbb{R}_+^N \times \mathbb{R}_+} & \quad ((-1)^{\delta_{0I}} f(\lambda, \theta)) \\ & \quad c(\lambda, \theta) \leq 0 \\ & \quad A_p \begin{bmatrix} \lambda \\ \theta \end{bmatrix} \leq b_p \\ & \quad A_{eq} \begin{bmatrix} \lambda \\ \theta \end{bmatrix} = b_{eq} \\ & \quad \lambda \geq 0 \\ & \quad \delta_{0I} \leq \theta \leq \delta(\delta_{1I}) + 1 \end{aligned} \tag{M-3}$$

where δ is the Dirac delta function [30] commonly known by

$$\lim_{t \rightarrow t_0} \delta(t - t_0) = \infty \text{ and } \delta(t) = 0 \text{ for } t \neq 0,$$

and loosely thought of to be

$$\delta(t) = \begin{cases} 0 & \text{if } t \neq 0 \\ \infty & \text{if } t = 0 \end{cases}$$

After solving the optimization problem (M-3) the relative efficiency score for the p -th DMU is calculated by

$$e_p = \delta_{1I} \theta_p + \delta_{0I} \left(\frac{1}{\theta_p} \right). \tag{9}$$

Moreover, the MI-DEA optimization problem (M-2) is equivalent to the following linear optimization problem

$$\begin{aligned} \min_{(\lambda, \theta) \in \mathbb{R}_+^N \times \mathbb{R}_+} & \quad ((-1)^{\delta_{0I}} f(\lambda, \theta)) \\ & \quad A_p^{(s,t)} \begin{bmatrix} \lambda \\ \theta \end{bmatrix} \leq b_p^{(t)} \\ & \quad \lambda \geq 0 \\ & \quad \delta_{0I} \leq \theta \leq \delta(\delta_{1I}) + 1 \end{aligned} \tag{M-4}$$

The matrices $A_p^{(s,t)}$ and $b_p^{(t)}$ are defined in the same manner as in (4) and (5) but over different periods

$$A_p^{(s,t)} = \begin{bmatrix} \mathcal{M}_j^{(s)} & -\delta_{1I} [\mathcal{M}_j^{(t)}]_{:p} \\ -\mathcal{M}_o^{(s)} & \delta_{0I} [\mathcal{M}_o^{(t)}]_{:p} \end{bmatrix}, \tag{7}$$

and

$$b_p^{(t)} = \begin{bmatrix} \delta_{0I} [\mathcal{M}_j^{(t)}]_{:p} \\ -\delta_{1I} [\mathcal{M}_o^{(t)}]_{:p} \end{bmatrix}. \tag{8}$$

3.2. Algorithm

The algorithm for solving various types of DEA is aligned in this section.

1. Initialize the parameters n : number of DMUs, m_I : number of inputs, m_O : number of outputs.
2. Determine the stochastic variables if any (i.e., determine $\mathcal{J}_S, \mathcal{J}_D, \mathcal{O}_S, \mathcal{O}_D$).
3. Choose the scale variation ($V = 1$ for VRS and $V = 0$ for CRS), and the orientation ($I = 1$ for input-oriented and $I = 0$ for output-oriented).
4. Load the data for the deterministic input(s) and output(s) of each DMU, and the parameters of the stochastic variables if exist.
5. Construct the matrices $\mathcal{M}_J, \mathcal{M}_O$ using (2).
6. If $\mathcal{J}_S \cup \mathcal{O}_S \neq \emptyset$, set the matrices M_{cov} and \tilde{M}_{cov} with the vectors μ and $\tilde{\mu}$, then define the nonlinear function $c(\lambda, \theta)$ as in (6).
7. Construct the matrices A_{eq} and b_{eq} using (3).
8. Determine the upper and the lower bounds to be imposed on θ , as described above $\delta_{0I} \leq \theta \leq \delta(\delta_{1I}) + 1$, and define the objective function $f(\lambda, \theta) = \theta$.
9. For $p = 1, \dots, N$, update the matrices A_p and b_p according to Equations (4) and (5) then solve the problem (M-3) to obtain the relative efficiency score for the p -th DMU as defined in (9).

In case the Malmquist Index DEA is to be used, then two periods should be defined and accordingly two sets of inputs and outputs for the two periods are loaded. Thus, the steps followed are as follows

1. Initialize the parameters n : number of DMUs, m_I : number of inputs, m_O : number of outputs.
2. Choose the orientation ($I=1$ for input-oriented and $I=0$ for output-oriented).
3. Load the data for the deterministic input(s) and output(s) of each DMU.
4. Construct the matrices $\mathcal{M}_J^{(t)}, \mathcal{M}_O^{(t)}$ for $t = 1, 2$, as in (2) but over the specified periods.
5. Construct the matrices A_{eq} and b_{eq} using (3).
6. Determine the upper and the lower bound to be used on θ and define the objective function.
7. Let $(s, t) \in \{1, 2\} \times \{1, 2\}$.
8. For $p = 1, \dots, N$, update the matrices $A_p^{(s,t)}$ and b_p^t defined in (7) and (8), then solve the problem (M-4) to obtain the relative efficiency score for the p -th DMU in period s measured by frontier technology t .
9. Go to step 6 and initialize (s, t) with a different value.
10. Calculate the productivity change over time for all the DMUs using equation (1).

4. Illustrative numerical example

This section provides a hypothetical example with three cases to show the performance of the given algorithm. The data are generated randomly and shown in Table 1.

Table 1. Hypothetical data for 20 DMUs

DMU	x_1	x_2	x_3	y_1	y_2	μ_1
DMU 1	157	7	20.599	2779	0.243	6.592
DMU 2	115	7	18.519	1655	0.561	9.399
DMU 3	131	8	33.889	2495	0.968	3.678
DMU 4	121	3	2.341	4175	0.420	4.095
DMU 5	141	6	25.309	1449	0.360	1.970
DMU 6	146	6	21.049	2438	0.372	5.170
DMU 7	114	3	28.900	1546	0.923	7.946
DMU 8	182	8	9.296	4515	0.834	9.407

DMU	x_1	x_2	x_3	y_1	y_2	μ_1
DMU 9	169	9	37.643	1588	0.420	0.080
DMU 10	109	8	17.227	4520	0.836	6.152
DMU 11	106	1	42.478	2879	0.895	0.668
DMU 12	142	5	31.022	1027	0.214	6.648
DMU 13	192	9	40.225	4079	0.605	0.282
DMU 14	142	1	18.773	1349	0.923	5.714
DMU 15	189	2	22.024	1047	0.712	3.187
DMU 16	192	5	26.883	3180	0.820	0.690
DMU 17	126	9	19.575	3136	0.586	1.862
DMU 18	103	5	17.574	3819	0.241	6.701
DMU 19	182	1	13.629	1605	0.639	0.201
DMU 20	103	6	15.762	3107	0.746	6.014

Case 1. In this case, only deterministic variables are involved with three inputs and 2 outputs. The efficiency scores for the CCR and VRS with both orientations are given in Table 2. DMUs 4, 7, 8, 10, 11, 14, and 19 are efficient in all the cases. DMUs 3, 18, and 20 are almost efficient with the CRS model and efficient with VRS. DMUs 2, 15, 16, and 17 are moderately efficient in all cases. Lastly, DMUs 1, 5, 6, 9, and 12 range from being severely inefficient to moderately efficient. DMU 13 is an interesting case where it's inefficient in all cases except for output oriented VRS model, where its performance is almost efficient.

Case 2. In this case, a stochastic input is added to the efficiency evaluation procedure. The mean of that input for each DMU is given in Table 1, the covariance matrix is randomly generated, and the efficiency scores are in Table 2. The performances of DMUs 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 18, 19, and 20 are not affected by the addition of this variable. While DMUs 9, 13, 16, and 17 performances are extremely affected by this variable, they become efficient in all cases. DMU 5 has a special case where it's inefficient in all cases except for the stochastic VRS input-oriented model.

Table 2. Efficiency scores for the 20 DMUs

DMU	CRS		VRS		Stochastic CRS		Stochastic VRS	
	Input	Output	Input	Output	Input	Output	Input	Output
DMU 1	0.474	0.474	0.680	0.624	0.485	0.485	0.706	0.624
DMU 2	0.647	0.647	0.896	0.653	0.647	0.647	0.896	0.653
DMU 3	0.916	0.916	1.000	1.000	0.998	0.998	1.000	1.000
DMU 4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DMU 5	0.350	0.350	0.744	0.407	0.483	0.483	0.932	0.483
DMU 6	0.460	0.460	0.735	0.556	0.485	0.485	0.772	0.563
DMU 7	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DMU 8	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DMU 9	0.314	0.314	0.609	0.463	1.000	1.000	1.000	1.000
DMU 10	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DMU 11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DMU 12	0.217	0.217	0.743	0.263	0.217	0.217	0.744	0.263
DMU 13	0.566	0.566	0.576	0.902	1.000	1.000	1.000	1.000
DMU 14	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

DMU	CRS		VRS		Stochastic CRS		Stochastic VRS	
	Input	Output	Input	Output	Input	Output	Input	Output
DMU 15	0.619	0.619	0.721	0.766	0.774	0.774	0.794	0.852
DMU 16	0.683	0.683	0.731	0.928	1.000	1.000	1.000	1.000
DMU 17	0.613	0.613	0.817	0.699	1.000	1.000	1.000	1.000
DMU 18	0.978	0.978	1.000	1.000	0.978	0.978	1.000	1.000
DMU 19	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DMU 20	0.977	0.977	1.000	1.000	0.977	0.977	1.000	1.000

Table 3. Hypothetical data for the 20 DMUs over period 2

DMU	$r^{(2)}$	$r^{(2)}$	$r^{(2)}$	$v^{(2)}$	$v^{(2)}$
DMU 1	161	9	19.607	2248	0.638
DMU 2	193	6	8.826	4451	0.339
DMU 3	126	6	9.788	3502	0.476
DMU 4	135	8	12.364	3422	0.549
DMU 5	199	6	15.882	3056	0.181
DMU 6	156	3	4.685	2999	0.168
DMU 7	111	6	5.901	4666	0.941
DMU 8	167	4	1.324	3868	0.264
DMU 9	168	1	4.868	1562	0.536
DMU 10	129	10	15.308	1428	0.956
DMU 11	176	8	4.979	2513	0.702
DMU 12	160	8	15.187	1194	0.691
DMU 13	108	1	8.620	1413	0.351
DMU 14	108	7	8.824	2988	0.015
DMU 15	142	6	1.041	4554	0.106
DMU 16	186	9	2.456	1294	0.226
DMU 17	169	1	2.670	2725	0.241
DMU 18	179	4	11.937	2894	0.500
DMU 19	183	2	16.738	1085	0.737
DMU 20	172	8	6.727	4832	0.557

Case 3. In this case, one more period is added to the above one and the data associated with the inputs and outputs for the new period are set in Table 3. The catch-up, frontier-shift and total performance change based on the Malmquist index technology are gathered in Table 4. For the CRS MI-DEA model, 55% of the DMUs improved their performance, and these are DMU 1, 2, 5, 6, 7, 8, 9, 12, 13, 15, and 17. It is worth mentioning that a remarkable improvement by DMU 9 and DMU 12 can be recognized. However, the rest DMUs regress in their performance. For the VRS MI-DEA model, also 55% appeared to progress in efficiency evaluation, but not the same DMUs. The DMUs 2, 6, 7, 8, 9, 10, 13, 15, 16, 17, and 20 improved while the others didn't. The occurrence of DMU 12 is very interesting as according to the CRS model, its relative performance improved by 276% and was among the highest, whereas according to the VRS model, its performance declined. This highlights the importance of choosing a suitable orientation for the case under study.

Table 4. Malmquist index DEA productivity change factor

DMU	CRS			VRS		
	Catch-up	frontier-shift	tfpch	Catch-up	frontier-shift	tfpch
DMU 1	0.986	1.130	1.114	0.999	0.979	0.979
DMU 2	1.202	0.971	1.166	1.013	1.025	1.039
DMU 3	0.787	1.040	0.818	0.869	1.012	0.879
DMU 4	0.603	1.039	0.627	0.811	0.960	0.778
DMU 5	1.505	0.909	1.369	0.764	0.993	0.758
DMU 6	1.795	0.908	1.629	1.224	0.990	1.211
DMU 7	1.000	1.012	1.012	1.000	1.014	1.014
DMU 8	1.000	1.219	1.219	1.000	1.201	1.201
DMU 9	3.187	1.040	3.313	1.641	0.986	1.617
DMU 10	0.875	1.084	0.948	1.000	1.003	1.003
DMU 11	0.852	1.034	0.881	0.862	0.984	0.848
DMU 12	2.471	1.117	2.760	0.923	1.033	0.954
DMU 13	1.766	0.956	1.689	1.737	1.019	1.770
DMU 14	0.658	1.032	0.679	1.000	0.986	0.986
DMU 15	1.616	0.807	1.305	1.387	0.894	1.240
DMU 16	0.748	0.939	0.702	1.008	1.074	1.082
DMU 17	1.630	1.041	1.698	1.223	1.005	1.229
DMU 18	0.692	0.950	0.657	0.694	0.963	0.668
DMU 19	1.000	0.882	0.882	1.000	0.887	0.887
DMU 20	0.774	1.097	0.849	1.000	1.007	1.007

It is worth mentioning that the MATLAB function used obtained the same results as the well-known program DEAP [19] with the absence of a stochastic variable, as this last is not implemented in DEAP.

5. Conclusion and future work

This paper provides a constructive algorithm to solve various types of DEA models. A unified second-order conic optimization problem is obtained representing all the former models. In addition, an associated MATLAB code is given for further usage and implementation. It is to be noted that this code can be easily transformed into different programming languages. Among the DEA models, input/output-oriented, variable/constant return to scale, and normal/Malmquist models are considered together with the fact that stochastic variables (inputs or outputs) can be integrated.

For future work, it will be interesting to allow the stochastic variables under consideration to follow different probability distributions and not be restricted to the normal one [31]. Furthermore, reflecting fuzzy DEA models [32] is an enhancement of the provided algorithm. Moreover, including a function that systematically runs regression analysis of the obtained efficiency scores perhaps is an added value to DEA.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

References

- [1] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision-making units," *European journal of operational research*, vol. 3, no. 4, pp. 339–338, 1979.
- [2] R. C. Sickles and V. Zelenyuk, *Measurement of productivity and efficiency*. Cambridge University Press, 2019.
- [3] A. Dellnitz, "Big data efficiency analysis: Improved algorithms for data envelopment analysis involving large datasets," *Computers & Operations Research*, vol. 137, p. 105553, 2022.
- [4] A. Emrouznejad and G. Yang, "A survey and analysis of the first 40 years of scholarly literature in DEA: 1978–2016," *Socio-Economic Planning Sciences*, vol. 61, pp. 4–8, Mar. 2018, doi: 10.1016/j.seps.2017.01.008.
- [5] P. Dutta, B. Jaikumar, and M. S. Arora, "Applications of data envelopment analysis in supplier selection between 2000 and 2020: A literature review," *Annals of Operations Research*, pp. 1–56, 2021.
- [6] F. S. M. Chachuli, N. A. Ludin, M. A. M. Jedi, and N. H. Hamid, "Transition of renewable energy policies in Malaysia: Benchmarking with data envelopment analysis," *Renewable and Sustainable Energy Reviews*, vol. 150, p. 111456, 2021.
- [7] J. Yang and B. Chen, "Energy efficiency evaluation of wastewater treatment plants (WWTPs) based on data envelopment analysis," *Applied Energy*, vol. 289, p. 116680, 2021.
- [8] X. Zeng, Z. Zhou, and C. Liu, "Chinese urban energy and carbon congestion effects: A data envelopment analysis and materials balance approach," *Journal of Cleaner Production*, vol. 341, p. 130817, 2022.
- [9] A. Monzeli, B. Daneshian, G. Tohidi, M. Sanei, and S. Razavyan, "Evaluating the Efficiency of Hospital Emergencies during COVID-19 Pandemic Crisis in the Presence of Undesirable Inputs in DEA," *Fuzzy Optimization and Modeling Journal*, vol. 2, no. 3, pp. 47–55, 2021.
- [10] N. Mourad, A. M. Habib, and A. Tharwat, "Appraising healthcare systems' efficiency in facing COVID-19 through data envelopment analysis," *10.5267/j.dsl*, vol. 10, no. 3, pp. 301–310, 2021, doi: 10.5267/j.dsl.2021.2.007.
- [11] A. Taherinezhad and A. Alinezhad, "Nations performance evaluation during SARS-CoV-2 outbreak handling via data envelopment analysis and machine learning methods," *International Journal of Systems Science: Operations & Logistics*, pp. 1–18, 2022.
- [12] N. Neykov, S. Krišt'áková, I. Hajdúchová, M. Sedliačiková, P. Antov, and B. Giertliová, "Economic efficiency of forest enterprises—Empirical study based on data envelopment analysis," *Forests*, vol. 12, no. 4, p. 462, 2021.
- [13] Y. Sun and N. Wang, "Eco-efficiency in China's Loess Plateau Region and its influencing factors: a data envelopment analysis from both static and dynamic perspectives," *Environmental Science and Pollution Research*, vol. 29, no. 1, pp. 483–497, 2022.
- [14] M. Mardani, M. Sabouni, H. Azadi, and M. Taki, "Rice production energy efficiency evaluation in north of Iran; application of Robust Data Envelopment Analysis," *Cleaner Engineering and Technology*, vol. 6, p. 100356, 2022.
- [15] M. N. Nodin, Z. Mustafa, and S. I. Hussain, "Assessing rice production efficiency for food security policy planning in Malaysia: A non-parametric bootstrap data envelopment analysis approach," *Food Policy*, vol. 107, p. 102208, 2022.
- [16] W.-T. Pan, M.-E. Zhuang, Y.-Y. Zhou, and J.-J. Yang, "Research on sustainable development and efficiency of China's E-Agriculture based on a data envelopment analysis-Malmquist model," *Technological Forecasting and Social Change*, vol. 162, p. 120298, 2021.
- [17] M. Loganathan and M. H. Subrahmanya, "Efficiency of Entrepreneurial Universities in India: A Data Envelopment Analysis," *Journal of the Knowledge Economy*, pp. 1–25, 2022.

-
- [18] N. Mourad and A. Tharwat, "The Efficiency of a University's Colleges: A Case Study using Data Envelopment Analysis," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 8, pp. 515–523, 2020.
- [19] "Software - School of Economics - University of Queensland." <https://economics.uq.edu.au/cepa/software> (accessed Apr. 17, 2022).
- [20] "MATLAB - MathWorks - MATLAB & Simulink." <https://www.mathworks.com/products/matlab.html> (accessed Apr. 17, 2022).
- [21] J.-M. Huguenin, "Data envelopment analysis (DEA)," A pedagogical guide for decision makers in the public sector, Swiss Graduate School of Public Administration, Lausanne, 2012.
- [22] W. W. Cooper, L. M. Seiford, and K. Tone, *Data envelopment analysis: a comprehensive text with models, applications, references and DEA-solver software*, vol. 2. Springer, 2007.
- [23] M. M. Martič, M. S. Novakovič, and A. Baggia, "Data envelopment analysis-basic models and their utilization," *Organizacija*, vol. 42, no. 2, 2009.
- [24] W. W. Cooper, H. Deng, Z. Huang, and S. X. Li, "Chance constrained programming approaches to congestion in stochastic data envelopment analysis," *European Journal of Operational Research*, vol. 155, no. 2, pp. 487–501, 2004.
- [25] W. W. Cooper, H. Deng, Z. Huang, and S. X. Li, "Chance constrained programming approaches to technical efficiencies and inefficiencies in stochastic data envelopment analysis," *Journal of the Operational Research Society*, vol. 53, no. 12, pp. 1347–1356, 2002.
- [26] B. Lin and R. Fei, "Regional differences of CO2 emissions performance in China's agricultural sector: A Malmquist index approach," *European Journal of Agronomy*, vol. 70, pp. 33–40, Oct. 2015, doi: 10.1016/j.eja.2015.06.009.
- [27] T. S. Desta, "Are the best African banks really the best? A Malmquist data envelopment analysis," *MEDAR*, vol. 24, no. 4, pp. 588–610, Oct. 2016, doi: 10.1108/MEDAR-02-2016-0016.
- [28] J. J. V. Sánchez, "Malmquist index with time series to data envelopment analysis," *Multi-Criteria Methods and Techniques Applied to Supply Chain Management*, vol. 111, 2018.
- [29] "Release DEA · nahiamourad/DEA," GitHub. <https://github.com/nahiamourad/DEA/releases/tag/v1.0.0> (accessed Apr. 17, 2022).
- [30] P. A. M. Dirac, *The principles of quantum mechanics*. Oxford university press, 1981.
- [31] N. Mourad and A. Tharwat, "Mixed Stochastic Input Oriented Data Envelopment Analysis Model," 2019.
- [32] A. Hatami-Marbini, A. Emrouznejad, and M. Tavana, "A taxonomy and review of the fuzzy data envelopment analysis literature: two decades in the making," *European journal of operational research*, vol. 214, no. 3, pp. 457–472, 2011.