

# IFRO Working Paper



Benchmarking with uncertain data:  
a simulation study comparing  
alternative methods

*Jens Leth Hougaard*  
*Pieter Jan Kerstens*  
*Kurt Nielsen*

**2019 / 05**

**IFRO Working Paper 2019 / 05**

Benchmarking with uncertain data: a simulation study comparing alternative methods

Authors: Jens Leth Hougaard, Pieter Jan Kerstens, Kurt Nielsen

JEL-classification: C61, D24

This research was partly funded by the Bio Based Industries Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 720757 Tech4Effect.

Published: September 2019

See the full series IFRO Working Paper here:

[www.ifro.ku.dk/english/publications/ifro\\_series/working\\_papers/](http://www.ifro.ku.dk/english/publications/ifro_series/working_papers/)

Department of Food and Resource Economics (IFRO)

University of Copenhagen

Rolighedsvej 25

DK 1958 Frederiksberg DENMARK

[www.ifro.ku.dk/english/](http://www.ifro.ku.dk/english/)

# Benchmarking with uncertain data: a simulation study comparing alternative methods\*

Jens Leth Hougaard<sup>†‡</sup>     Pieter Jan Kerstens<sup>§</sup>     Kurt Nielsen<sup>¶</sup>

September 2, 2019

## Abstract

We consider efficiency measurement methods in the presence of uncertain input and output data, and without the (empirically problematic) assumption of convexity of the production technology. In particular, we perform a simulation study in order to contrast two well-established methods, IDEA and Fuzzy DEA, with a recently suggested extension of Fuzzy DEA in the literature (dubbed the HB method). We demonstrate that the HB method has important advantages over the conventional methods, resulting in more accurate efficiency estimates and narrower bounds for the efficiency scores of individual Decision Making Units (DMUs): thereby providing more informative results that may lead to more effective decisions. The price is computational complexity. Although we show how to significantly speed up computational time compared to the original suggestion, the HB method remains the most computationally heavy method among those considered. This may limit the use of the method in cases where efficiency estimates have to be computed on the fly, as in interactive decision support systems based on large data sets.

**Keywords:** data envelopment analysis, data uncertainty, fuzzy, imprecise data envelopment analysis, simulation

---

\*This research was partly funded by the Bio Based Industries Joint Undertaking under the European Union's Horizon 2020 research and innovation program under grant agreement No 720757 Tech4Effect.

<sup>†</sup>Department of Food and Resource Economics (IFRO), University of Copenhagen, Rolighedsvej 25, DK-1958 Frederiksberg C, Denmark. Email: jlh@ifro.ku.dk

<sup>‡</sup>Economics, NYU Shanghai, China.

<sup>§</sup>Department of Food and Resource Economics (IFRO), University of Copenhagen, Rolighedsvej 25, DK-1958 Frederiksberg C, Denmark. Email: pjk@ifro.ku.dk.

<sup>¶</sup>Department of Food and Resource Economics (IFRO), University of Copenhagen, Rolighedsvej 25, DK-1958 Frederiksberg C, Denmark. Email: kun@ifro.ku.dk

# 1 Introduction

*Overview:* When studying benchmarking in practice there are often considerable data uncertainties involved: for instance using data that by nature is imprecise, like preference information or information extracted from text and images. Consequently, a wide range of benchmarking methods have been developed to account for data uncertainties: see e.g., [Emrouznejad and Yang \(2018\)](#) and [Hatami-Marbini et al. \(2011\)](#). In the present paper we compare the performance of two of the most popular among these methods to a recently suggested method in [Hougaard and Baležentis \(2014\)](#). By a simulation study, we show that this new method has a significant advantage in terms of outcome: results are more accurate, and reported efficiency bounds are narrower, yielding more meaningful (and useful) interpretations of the results. The price, however, is computational complexity. Even though we show how to significantly reduce the computational burden and the implementation complexity compared to the original formulation of the method, it is still far heavier to compute than the immediate alternatives. This method is therefore certainly an attractive alternative to use in many applications, but given its computational cost might not be ready yet for interactive benchmarking purposes where results must be computed and analyzed on the fly.

*Background:* Benchmarking by way of efficiency measurement techniques has a long line of tradition, see e.g., [Bogetoft and Otto \(2010\)](#). In particular, variations of Data Envelopment Analysis (DEA) has proved tremendously popular. Conventional DEA, and its variations, requires crisp data on inputs and outputs, and produce a set of crisp efficiency scores, one for each Decision Making Unit (DMU) that is being analyzed. Yet, crisp data and crisp efficiency scores often fail to reflect the underlying uncertainties that are naturally involved in practical applications of such methods.

Consequently, there has been several attempts to extend the methods to encompass data uncertainties and result in efficiency ranges (or bounds) rather than crisp scores: all in order to improve the decision process related to benchmarking, see e.g., [Hatami-Marbini et al. \(2011\)](#) for a recent review of these methods. Two main approaches are (i) Imprecise DEA (IDEA) as originally suggested in [Cooper et al. \(1999\)](#) and further developed by several authors (e.g., [Park \(2004, 2007\)](#), [Despotis and Smirlis \(2002\)](#)); and (ii) Fuzzy DEA where data are represented by fuzzy sets and fuzzy programming techniques are used as originally suggested in [Sengupta \(1992\)](#) and also further developed by several authors (e.g., [Kao and Liu \(2000\)](#), [Triantis and Girod \(1998\)](#), [Hougaard \(1999, 2005\)](#)).

However, in practice many of these methods produce efficiency results with little

informative value for the decision maker: ranges of efficiency scores are typically too wide to be informative. In other words, there is a delicate balance between acknowledging the uncertainties involved and producing informative results that may lead to effective and justified decisions.

We test the performance of a new method designed to produce more informative results in the practically appealing case of a non-convex production technology. The method was suggested in [Hougaard and Baležentis \(2014\)](#) and relates directly to crisp FDH as defined in [Tulkens \(1993\)](#) by mimicking the computation of efficiency, but allowing for fuzzy data and fuzzy efficiency scores. In particular, the notion of dominance – by which the DMUs are ordered – is constructed using a probabilistic approach.

*Content:* In section 2 we present the model and some preliminaries. Section 3 present the two well established methods, IDEA and Fuzzy DEA, as well as the newly suggested HB method. In section 4 we present the simulation study and its results, and section 5 concludes.

## 2 Model and preliminaries

We consider a finite set of Decision Making Units (DMUs), where each DMU  $k$  uses  $N$  inputs  $\mathbf{X}_{k,t} \in \mathbb{R}_+^N$  to produce  $M$  outputs  $\mathbf{Y}_{k,t} \in \mathbb{R}_+^M$  at time period  $t = 1, \dots, T$ . The production technology is  $\mathcal{T}_t = \left\{ (\mathbf{X}, \mathbf{Y}) \in \mathbb{R}_+^{(N+M)} \mid \mathbf{X} \text{ can produce } \mathbf{Y} \right\}$  at time period  $t$ , with the only assumption of free disposability of inputs and outputs: i.e.,  $(\mathbf{X}, \mathbf{Y}) \in \mathcal{T}_t \Rightarrow (\mathbf{X}', \mathbf{Y}') \in \mathcal{T}_t$ , with  $\mathbf{X}' \geq \mathbf{X}$ , and  $\mathbf{Y}' \leq \mathbf{Y}$ .

Unlike most other models, we do *not* assume convexity of  $\mathcal{T}_t$  for two reasons: (i) convexity is debatable on both theoretical and practical grounds; and (ii) non-convex DEA models can be solved by simple enumeration which makes them considerable faster than the ordinary linear programs resulting from the convexity assumption.

Furthermore, let  $(\tilde{\mathbf{X}}_{k,t}, \tilde{\mathbf{Y}}_{k,t}) = (\mathbf{X}_{k,t} + \epsilon_{k,t}^x, \mathbf{Y}_{k,t} + \epsilon_{k,t}^y)$  be the actual observations with measurement errors  $(\epsilon_{k,t}^x, \epsilon_{k,t}^y)$  which are i.i.d and normally distributed (i.e.,  $\epsilon_{k,t}^x \sim \mathcal{N}(0, \sigma_x^2)$  and  $\epsilon_{k,t}^y \sim \mathcal{N}(0, \sigma_y^2)$ ).

In order to simplify the computations of the considered methods, we assume that the observed data is interval-bounded and we construct these interval bounds using

confidence intervals for the normal distribution:

$$\mathbf{X}_{k,t} \in [\mathbf{X}_{k,t}^L, \mathbf{X}_{k,t}^U] = [\tilde{\mathbf{X}}_{k,t} + z_{\alpha/2}\sigma_x, \tilde{\mathbf{X}}_{k,t} + z_{1-\alpha/2}\sigma_x] \quad (1a)$$

$$\mathbf{Y}_{k,t} \in [\mathbf{Y}_{k,t}^L, \mathbf{Y}_{k,t}^U] = [\tilde{\mathbf{Y}}_{k,t} + z_{\alpha/2}\sigma_y, \tilde{\mathbf{Y}}_{k,t} + z_{1-\alpha/2}\sigma_y] \quad (1b)$$

By varying the confidence level  $\alpha$ , we can shrink or widen the intervals. In the simulations below we set  $\alpha = 0.05$ .

Data uncertainty can also be expressed making use of fuzzy numbers. In particular, we consider triangular fuzzy numbers which have an intuitive interpretation of capturing the uncertainty surrounding an observation. More formally, a triangular fuzzy number  $A$  is specified by numbers  $a, b, c \in \mathbb{R}_+$  where  $a \leq b \leq c$  and its  $\alpha$ -cut is defined as  $(A)_\alpha = [(a)_\alpha, (c)_\alpha] = [(b-a)\alpha + a, -(c-b)\alpha + c] \forall \alpha \in [0, 1]$  where  $\alpha$  captures the uncertainty level of the observation. No uncertainty corresponds to  $\alpha = 1$  and yields  $(A)_1 = b$  (“kernel”) while complete uncertainty corresponds to  $\alpha = 0$  and yields  $(A)_0 = [a, c]$  (“support”). Figure 1 shows an example of a triangular fuzzy number and one  $\alpha$ -cut.

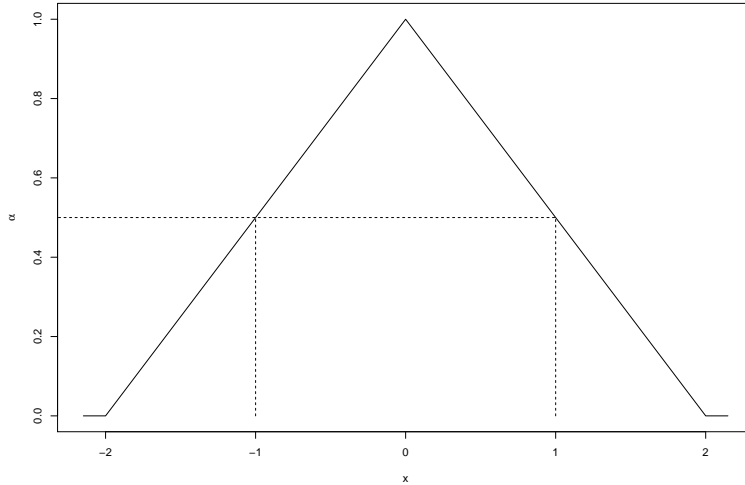


Figure 1: Triangular fuzzy number with  $a = -2$ ,  $b = 0$  and  $c = 2$ . The  $\alpha$ -cut for  $\alpha = 0.5$  yields the interval  $[-1, 1]$ .

Interval-bounded data as in (1) is then straightforwardly cast into triangular fuzzy numbers representation by specifying  $a = \mathbf{X}_{k,t}^L$ ,  $b = \tilde{\mathbf{X}}_{k,t}$  and  $c = \mathbf{X}_{k,t}^U$  for the input data

and analogously for the output data.

### 3 Three different methods to handle data uncertainty

In the following we present two popular approaches to handle data uncertainties in benchmarking. One is a natural extension of conventional DEA called Imprecise DEA (IDEA), the other relates to data in the form of fuzzy numbers. We contrast these approaches with a new method where fuzzy numbers are handled without having to rely on (fuzzy) programming techniques.

#### 3.1 Imprecise Data Envelopment Analysis (IDEA)

The first method we consider is “Imprecise Data Envelopment Analysis” (IDEA) which was initially proposed by [Cooper et al. \(1999\)](#). The model we consider here follows [Park \(2004, 2007\)](#) closely with some minor adjustments. We refer the interested reader to [Park \(2007\)](#) for the full derivation and details. Measuring efficiency using an output Farrell distance function approach, we start from the following IDEA model:

$$\max_{\theta, \lambda} \theta \tag{2a}$$

$$\sum_{k=1}^K \lambda_k \tilde{\mathbf{X}}_k \leq \tilde{\mathbf{X}}_0 \tag{2b}$$

$$\sum_{k=1}^K \lambda_k \tilde{\mathbf{Y}}_k \geq \theta \tilde{\mathbf{Y}}_0 \tag{2c}$$

$$\sum_{k=1}^K \lambda_k = 1 \tag{2d}$$

$$\lambda_k \in \{0, 1\} \quad \forall k = 1, \dots, K \tag{2e}$$

$$(\tilde{X}_1^n, \dots, \tilde{X}_K^n) = \tilde{\mathbf{X}}^n \in D_n^- \quad \forall n = 1, \dots, N \tag{2f}$$

$$(\tilde{Y}_1^m, \dots, \tilde{Y}_K^m) = \tilde{\mathbf{Y}}^m \in D_m^+ \quad \forall m = 1, \dots, M \tag{2g}$$

with the constraint sets

$$D_n^- = \{\mathbf{X}^n \in \mathbb{R}_+^K \mid \mathbf{H}_n^- \mathbf{X}^n \leq \mathbf{h}_n^-\} \quad \forall n = 1, \dots, N \tag{2h}$$

$$D_m^+ = \{\mathbf{Y}^m \in \mathbb{R}_+^K \mid \mathbf{H}_m^+ \mathbf{Y}^m \leq \mathbf{h}_m^+\} \quad \forall m = 1, \dots, M \tag{2i}$$

The constraint sets are quite general and allow for multiple different types of imprecise data (see [Park \(2004\)](#) for examples). In our case of interval-bounded data (1), these constraint sets become:

$$D_n^- = \left\{ \mathbf{X}^n \in \mathbb{R}_+^K \mid \begin{pmatrix} -\mathbf{I}_K \\ \mathbf{I}_K \end{pmatrix} \mathbf{X}^n \leq \begin{pmatrix} -\mathbf{X}_L^n \\ \mathbf{X}_U^n \end{pmatrix} \right\} \quad \forall n = 1, \dots, N \quad (3)$$

$$D_m^+ = \left\{ \mathbf{Y}^m \in \mathbb{R}_+^K \mid \begin{pmatrix} -\mathbf{I}_K \\ \mathbf{I}_K \end{pmatrix} \mathbf{Y}^m \leq \begin{pmatrix} -\mathbf{Y}_L^m \\ \mathbf{Y}_U^m \end{pmatrix} \right\} \quad \forall m = 1, \dots, M \quad (4)$$

Due to the assumed linear structure of the constraint sets, [Park \(2007\)](#) recognized that efficiency bounds can easily be computed using linear programming. It is quite straightforward to show that we can compute inefficiency bounds  $[\theta_0^L, \theta_0^U]$  by computing:

$$\theta_0^L = \max_{\theta, \lambda} \theta \quad (5a)$$

$$\sum_{k=1}^K \lambda_k \mathbf{X}_k^U \leq \mathbf{X}_0^L \quad (5b)$$

$$\sum_{k=1}^K \lambda_k \mathbf{Y}_k^L \geq \theta \mathbf{Y}_0^U \quad (5c)$$

$$\sum_{k=1}^K \lambda_k = 1 \quad (5d)$$

$$\lambda_k \in \{0, 1\} \quad \forall k = 1, \dots, K. \quad (5e)$$

and

$$\theta_0^U = \max_{\theta, \lambda} \theta \quad (6a)$$

$$\sum_{k=1}^K \lambda_k \mathbf{X}_k^L \leq \mathbf{X}_0^U \quad (6b)$$

$$\sum_{k=1}^K \lambda_k \mathbf{Y}_k^U \geq \theta \mathbf{Y}_0^L \quad (6c)$$

$$\sum_{k=1}^K \lambda_k = 1 \quad (6d)$$

$$\lambda_k \in \{0, 1\} \quad \forall k = 1, \dots, K. \quad (6e)$$



The intuition for these programs is as follows: the best-case scenario  $\theta_0^L$  for the evaluated DMU 0 is that its inputs are at the lower bound  $\mathbf{X}_0^L$  and its produced output at the upper bound  $\mathbf{Y}_0^U$  while all other DMUs' inputs equal the upper bound  $\mathbf{X}_k^U$  and produce the lower bound of outputs  $\mathbf{Y}_k^L$ . Similarly, the worst-case scenario  $\theta_0^U$  for the evaluated DMU 0 is that  $\tilde{\mathbf{X}}_0 = \mathbf{X}_0^U$  and  $\tilde{\mathbf{Y}}_0 = \mathbf{Y}_0^L$  while all other DMUs' input usage  $\tilde{\mathbf{X}}_k$  is at the lower bound  $\mathbf{X}_k^L$  and they all produce their upper bound  $\mathbf{Y}_k^U$ . This upper bound  $\theta_0^U$  essentially corresponds to the solution of a robust optimization problem with a box constraint (see [Gorissen et al. \(2015\)](#) for an introduction).

### 3.2 Fuzzy FDH

[Kao and Liu \(2000\)](#), here dubbed the KL method, derive an intuitively simple approach to compute FDH efficiency bounds from fuzzy numbers data. Their idea is actually identical to the underlying idea of [Park \(2004, 2007\)](#) for imprecise data with the only distinction that [Kao and Liu \(2000\)](#) compute efficiency bounds for different  $\alpha$ -cuts and that their approach is applicable to fuzzy numbers in general. For each  $\alpha$ -cut we find  $[(\theta_0^L)_\alpha, (\theta_0^U)_\alpha]$  by solving:

$$(\theta_0^L)_\alpha = \max_{\theta, \lambda} \theta \tag{7a}$$

$$\sum_{k=1}^K \lambda_k (\mathbf{X}_k^U)_\alpha \leq (\mathbf{X}_0^L)_\alpha \tag{7b}$$

$$\sum_{k=1}^K \lambda_k (\mathbf{Y}_k^L)_\alpha \geq \theta (\mathbf{Y}_0^U)_\alpha \tag{7c}$$

$$\sum_{k=1}^K \lambda_k = 1 \tag{7d}$$

$$\lambda_k \in \{0, 1\} \quad \forall k = 1, \dots, K. \tag{7e}$$

and

$$(\theta_0^U)_\alpha = \max_{\theta, \lambda} \theta \quad (8a)$$

$$\sum_{k=1}^K \lambda_k (\mathbf{X}_k^L)_\alpha \leq (\mathbf{X}_0^U)_\alpha \quad (8b)$$

$$\sum_{k=1}^K \lambda_k (\mathbf{Y}_k^U)_\alpha \geq \theta (\mathbf{Y}_0^L)_\alpha \quad (8c)$$

$$\sum_{k=1}^K \lambda_k = 1 \quad (8d)$$

$$\lambda_k \in \{0, 1\} \quad \forall k = 1, \dots, K. \quad (8e)$$

Thus, IDEA in (5)-(6) coincides with (7)-(8) for  $\alpha = 0$  in our particular case of triangular fuzzy numbers. To summarize the obtained efficiency intervals for different  $\alpha$ , we simply compute the arithmetic average of the lower and the arithmetic average of the upper bounds for the different  $\alpha$ .

The above programs (7)-(8) are mixed integer programming problems, but can efficiently be solved by enumeration (as with ordinary FDH methods (Tulkens, 1993)). The equivalent enumeration formula are:

$$(\theta_0^L)_\alpha = \max_{\forall k \in D_0^L} \left( \min_{\forall m=1, \dots, M} \left\{ \frac{(Y_k^{m,L})_\alpha}{(Y_0^{m,U})_\alpha} \right\} \right) \quad (9a)$$

where the dominance set is

$$D_0^L = \{k | (\mathbf{X}_k^U)_\alpha \leq (\mathbf{X}_0^L)_\alpha \text{ and } (\mathbf{Y}_k^L)_\alpha \geq (\mathbf{Y}_0^U)_\alpha\} \quad (9b)$$

and

$$(\theta_0^U)_\alpha = \max_{\forall k \in D_0^U} \left( \min_{\forall m=1, \dots, M} \left\{ \frac{(Y_k^{m,U})_\alpha}{(Y_0^{m,L})_\alpha} \right\} \right) \quad (10a)$$

where the dominance set is

$$D_0^U = \{k | (\mathbf{X}_k^L)_\alpha \leq (\mathbf{X}_0^U)_\alpha \text{ and } (\mathbf{Y}_k^U)_\alpha \geq (\mathbf{Y}_0^L)_\alpha\} \quad (10b)$$

Thus, solving both programs requires one simple loop over all observations. This implies that the computational effort scales linearly with the number of observations  $K$ .

### 3.3 Fuzzy FDH without programming

In contrast to the previous KL method, the fuzzy FDH approach of [Hougaard and Baležentis \(2014\)](#), here dubbed the HB method, does not rely on fuzzy programming techniques but utilizes ranking probabilities of intervals when defining a dominance relation between pairs of DMUs as well as Max and Min operators over sets of intervals. For each  $\alpha$ -level, data takes the form of intervals and whether one interval is smaller (larger) than another interval is represented by the probability that a value drawn from the first interval is smaller (larger) than a value drawn independently from the other interval given that values are independently and uniformly distributed over the intervals. Now, such probabilities can be established between every pair of DMUs for every (interval-valued) input and output, and based on these probabilities we can define a pairwise dominance relation. In the HB-method, the pairwise dominance relation  $\succ_d$  is defined as follows: DMU  $j$  dominates DMU  $j^*$  (written  $j \succ_d j^*$ ) if,

$$\begin{aligned}
 j \succ_d j^* \Leftrightarrow & \left[ \frac{\sum_{i=1}^N P^{\min} \left( [X_j^{i,L}, X_j^{i,U}], \{ [X_j^{i,L}, X_j^{i,U}], [X_{j^*}^{i,L}, X_{j^*}^{i,U}] \} \right)}{N} > 1/2 \text{ and} \right. \\
 & \exists i : P^{\min} \left( [X_j^{i,L}, X_j^{i,U}], \{ [X_j^{i,L}, X_j^{i,U}], [X_{j^*}^{i,L}, X_{j^*}^{i,U}] \} \right) < \psi \text{ and} \\
 & \frac{\sum_{r=1}^M P^{\max} \left( [Y_j^{r,L}, Y_j^{r,U}], \{ [Y_j^{r,L}, Y_j^{r,U}], [Y_{j^*}^{r,L}, Y_{j^*}^{r,U}] \} \right)}{M} > 1/2 \text{ and} \\
 & \exists r : P^{\max} \left( [Y_j^{r,L}, Y_j^{r,U}], \{ [Y_j^{r,L}, Y_j^{r,U}], [Y_{j^*}^{r,L}, Y_{j^*}^{r,U}] \} \right) < \psi \Big], \quad (11)
 \end{aligned}$$

where  $\psi \in (0, 1]$  is a parameter that determines the lowest acceptable level for partial dominance (the higher  $\psi$ , the harder it is for a DMU  $j$  to dominate), and  $P^{\min}(z_j, Z)$  ( $P^{\max}(z_j, Z)$ ) represent the probability that a value  $x$  drawn from interval  $z_j \in Z$  is smallest (largest) among values drawn independently from each interval in  $Z$ . The calculation of these probabilities is quite straightforward and enumerates all possible combinations of how  $z_j$  can be smallest (largest) among values drawn independently from every other interval in  $Z$  (see [Hougaard and Baležentis \(2014\)](#) for details).

Hence, DMU  $j$  dominates DMU  $j^*$  if and only if, on average over all inputs, the probability that DMU  $j$ 's input is the smallest among the two is larger than 1/2, with none being smaller than the parameter  $\psi$ , and if, on average over all outputs, the probability that DMU  $j$ 's output is largest among the two is larger than 1/2 (again with none

being smaller than  $\psi$ ). The set of DMUs dominating a given DMU  $j^*$  is denoted,

$$\mathcal{U}(j^*, \{1, \dots, K\}) = \{j \in \{1, \dots, K\} \setminus \{j^*\} \mid j \succ_d j^*\}. \quad (12)$$

Based on this set we can now mimic the standard definition of the Farrell efficiency score applied to the FDH model (see [Hougaard and Baležentis \(2014\)](#) for details) which further requires the definition of Max and Min operators over intervals.

Again we utilize the probabilities  $P^{\min}(z_j, Z)$  and  $P^{\max}(z_j, Z)$ , but since the set  $Z$  now may contain several intervals and the computation regards all possible combinations of how  $z_j$  can be smallest (largest) among values drawn independently from every other interval in  $Z$ , the computation of these probabilities quickly becomes cumbersome due to the potentially huge number of possible combinations. This turns out to be a major drawback of the method for practical applications even though the computations can quite easily be parallelized.

In order to avoid these cumbersome computations, we propose a solution method that relies on numerical integration. Let  $X_i \sim \mathcal{U}(z_i^L, z_i^R)$  and its cumulative uniform distribution  $F_{X_i}(x)$  with support  $[z_i^L, z_i^R]$  is:

$$F_{X_i}(x) = \begin{cases} 0 & \text{if } x < z_i^L \\ \frac{x - z_i^L}{\|z_i\|} & \text{if } z_i^L \leq x \leq z_i^R \\ 1 & \text{if } x > z_i^R \end{cases} \quad (13)$$

This piecewise nature of the cumulative uniform distribution can be exploited for the calculation of  $P^{\min}(z_j, Z)$  and  $P^{\max}(z_j, Z)$ . The probability that a value  $x$  drawn from an interval  $z_j \in Z$  is smallest among values drawn independently from each interval in  $Z$  is:

$$\begin{aligned}
P^{\min}(z_j, Z) &= \int_{z_j^L}^{z_j^R} \left[ \prod_{i \neq j} P(X_i \geq x) \right] P(X_j = x) dx \\
&= \int_{z_j^L}^{z_j^R} \left[ \prod_{i \neq j} (1 - F_{X_i}(x)) \right] \frac{1}{\|z_j\|} dx \\
&= \int_{z_j^L}^{\max\{z_j^L, \min_{i \neq j}\{z_i^L\}\}} \frac{1}{\|z_j\|} dx + \int_{\max\{z_j^L, \min_{i \neq j}\{z_i^L\}\}}^{\min_i\{z_i^R\}} \left[ \prod_{i \neq j} \left(1 - \frac{x - z_i^L}{\|z_i\|}\right) \right] \frac{1}{\|z_j\|} dx \\
&\quad + \int_{\min_i\{z_i^R\}}^{z_j^R} \frac{0}{\|z_j\|} dx \\
&= \left[ \frac{x}{\|z_j\|} \right]_{z_j^L}^{\max\{z_j^L, \min_{i \neq j}\{z_i^L\}\}} + \int_{\max\{z_j^L, \min_{i \neq j}\{z_i^L\}\}}^{\min_i\{z_i^R\}} \left[ \prod_{i \neq j} \left(1 - \frac{x - z_i^L}{\|z_i\|}\right) \right] \frac{1}{\|z_j\|} dx \\
&= \frac{\max\{z_j^L, \min_{i \neq j}\{z_i^L\}\} - z_j^L}{\|z_j\|} + \int_{\max\{z_j^L, \min_{i \neq j}\{z_i^L\}\}}^{\min_i\{z_i^R\}} \left[ \prod_{i \neq j} \left(1 - \frac{x - z_i^L}{\|z_i\|}\right) \right] \frac{1}{\|z_j\|} dx.
\end{aligned} \tag{14}$$

Analogously, one can rewrite the probability that a value  $x$  drawn from an interval  $z_j \in Z$  is largest among values drawn independently from each interval in  $Z$ :

$$\begin{aligned}
P^{\max}(z_j, Z) &= \int_{z_j^L}^{z_j^R} \left[ \prod_{i \neq j} P(X_i \leq x) \right] P(X_j = x) dx \\
&= \int_{z_j^L}^{z_j^R} \left[ \prod_{i \neq j} F_{X_i}(x) \right] \frac{1}{\|z_j\|} dx \\
&= \int_{z_j^L}^{\max_i \{z_i^L\}} \frac{0}{\|z_j\|} dx + \int_{\max_i \{z_i^L\}}^{\min\{z_j^R, \max_{i \neq j} \{z_i^R\}\}} \left[ \prod_{i \neq j} \left( \frac{x - z_i^L}{\|z_i\|} \right) \right] \frac{1}{\|z_j\|} dx \\
&+ \int_{\min\{z_j^R, \max_{i \neq j} \{z_i^R\}\}}^{z_j^R} \frac{1}{\|z_j\|} dx \\
&= \int_{\max_i \{z_i^L\}}^{\min\{z_j^R, \max_{i \neq j} \{z_i^R\}\}} \left[ \prod_{i \neq j} \left( \frac{x - z_i^L}{\|z_i\|} \right) \right] \frac{1}{\|z_j\|} dx + \left[ \frac{x}{\|z_j\|} \right]_{\min\{z_j^R, \max_{i \neq j} \{z_i^R\}\}}^{z_j^R} \\
&= \int_{\max_i \{z_i^L\}}^{\min\{z_j^R, \max_{i \neq j} \{z_i^R\}\}} \left[ \prod_{i \neq j} \left( \frac{x - z_i^L}{\|z_i\|} \right) \right] \frac{1}{\|z_j\|} dx + \frac{z_j^R - \min\{z_j^R, \max_{i \neq j} \{z_i^R\}\}}{\|z_j\|}.
\end{aligned} \tag{15}$$

The integrals in equations (14) and (15) can be solved directly using numerical integration procedures.<sup>1</sup> Furthermore, splitting up the integral in this way has the advantage that one minimizes the incurred numerical error due to the numerical integration method because the numerical integration interval is minimal. This entirely avoids the combinatorial procedure inherent in the definition of probabilities  $P^{\min}(z_j, Z)$  and  $P^{\max}(z_j, Z)$  (Hougaard and Baležentis (2014)) and makes the method tractable for larger datasets.

In our simulation study below we consider output-oriented efficiency only (the input-oriented case can be found in Hougaard and Baležentis (2014)). The output-oriented efficiency bounds are computed by:

$$E_{\alpha}^{\text{out}}(j^*, \mathcal{U}(j^*, \{1, \dots, K\})) = \arg \max_{\{z_j^{\text{out}}(\alpha) | j \in \mathcal{U}(j^*, \{1, \dots, K\})\}} \{P^{\min}(z_j^{\text{out}}(\alpha), Z^{\text{out}}(\alpha))\} \tag{16a}$$

where

$$z_j^{\text{out}}(\alpha) = \left[ \frac{Y_{j^*}^{r',L}}{Y_j^{r',U}}, \frac{Y_{j^*}^{r',U}}{Y_j^{r',L}} \right], \tag{16b}$$

---

<sup>1</sup>R comes with the function `integrate` to accomplish this.

and the output dimension  $r'$  with highest probability of dominating is determined by

$$r' = \arg \max_{r=1, \dots, M} P^{\max} \left( \left[ \frac{Y_j^{r,L}}{Y_j^{r,U}}, \frac{Y_j^{r,U}}{Y_j^{r,L}} \right], \left\{ \left[ \frac{Y_j^{r,L}}{Y_j^{r,U}}, \frac{Y_j^{r,U}}{Y_j^{r,L}} \right] \right\}_{r=1, \dots, M} \right). \quad (16c)$$

As before one calculates these efficiency bounds for different choices of  $\alpha$  and then computes an arithmetic average of the different efficiency bounds. Note that  $E_\alpha^{out}(\cdot) \in [0, 1]$  are output efficiency scores and the other methods provide Farrell output measures  $\theta_0 \geq 1$ . Both measures are related by:  $[1/(\theta_0^U)_\alpha, 1/(\theta_0^L)_\alpha] = E_\alpha^{out}(\cdot)$ .

## 4 Simulation study

We now compare the three methods – the IDEA method, the KL method and the HB method – by a simulation study.

### 4.1 Data generation process

The simulation data is generated as follows. The 2-dimensional input data  $\mathbf{X}_k \in \mathbb{R}_+^2$  of observation  $k$  is drawn uniformly over the closed interval  $[1, 2]$  (i.e.,  $X_k^1, X_k^2 \sim \mathcal{U}(1, 2)$ ). The 1-dimensional output  $Y_k \in \mathbb{R}_+$  is computed as  $Y_k = \min \{f(X_k^1), f(X_k^2)\}$  where  $f(\cdot)$  is a piecewise function consisting of two logistic functions:

$$f(X) = \begin{cases} \frac{4}{1+e^{-20(X-1.2)}} + 3 & \text{if } 1 \leq X < 1.5; \\ \frac{4}{1+e^{-10(X-1.5)}} + \frac{4}{1+e^{-6}} + 1 & \text{if } 1.5 \leq X \leq 2.0. \end{cases} \quad (17)$$

Figure 2 shows  $f(X)$  over its entire domain and clearly shows that it is non-concave. Thus, this particular choice of function yields a non-convex production set and therefore matches the underlying assumptions of the discussed models. Finally, we ensure the generated data is weakly efficient in both the input and output direction by running a radial input-output oriented DEA model and projecting the generated observations on the frontier of the production set.

For the purpose of this simulation we only introduce inefficiency and noise in the output. We draw the inefficiency term from a half-normal distribution and the noise

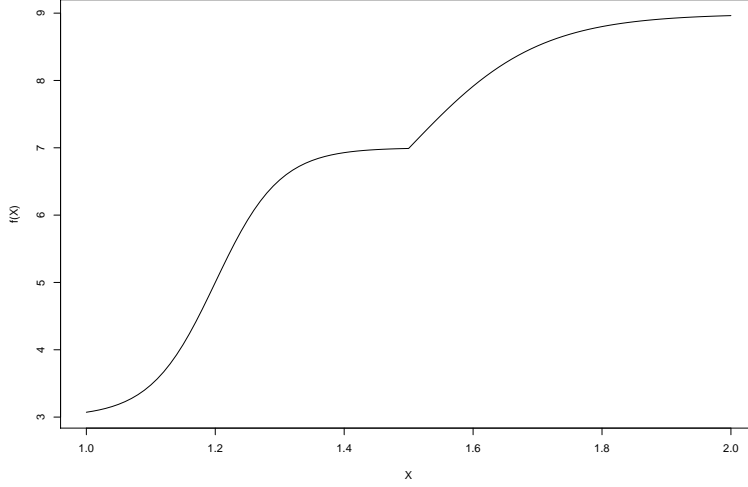


Figure 2:  $f(X)$

term from a normal distribution. The final observations  $k = 1, \dots, K$  are:

$$\tilde{\mathbf{X}}_k = \mathbf{X}_k \quad (18a)$$

$$\tilde{\mathbf{Y}}_k = \frac{1}{\theta_k} \mathbf{Y}_k + \epsilon_k^y. \quad (18b)$$

where  $\frac{1}{\theta_k} = \frac{1}{1+\tilde{u}_k}$  with  $\tilde{u}_k \sim \mathcal{N}^+(0, \sigma_\theta^2)$  and  $\epsilon_k^y \sim \mathcal{N}(0, \sigma_y^2)$ . All DEA methods envelop the data as close as possible and therefore at least one of the simulated observations needs to be efficient. Otherwise it would be impossible to accurately estimate  $\tilde{\theta}_k$  with our DEA methods: any DEA method would find  $\theta_k = 1$  for at least one observation while one can expect that all observations in the simulated data have  $\tilde{\theta}_k > 1$ .<sup>2</sup> For this reason we randomly select  $K/20$  observations and make them efficient. Figure 3 shows an example of simulated data for  $K = 500$  with  $\sigma_\theta = 0.1$  and  $\sigma_y = 0.1$ .

Because all noise parameters are known and the inputs are noise-free, (1) simplifies to:

$$\mathbf{X}_k \in [\mathbf{X}_k^L, \mathbf{X}_k^U] = [\mathbf{X}_k, \mathbf{X}_k] \quad (19a)$$

$$\mathbf{Y}_k \in [\mathbf{Y}_k^L, \mathbf{Y}_k^U] = \left[ \tilde{\mathbf{Y}}_k + z_{\alpha/2} \sigma_y, \tilde{\mathbf{Y}}_k + z_{1-\alpha/2} \sigma_y \right] \quad (19b)$$

We compare the discussed methods for 4 different scenarios with varying  $\sigma_\theta$  and  $\sigma_y$ .

---

<sup>2</sup>The probability of exactly drawing  $\tilde{\theta}_k = 1$  for any DMU  $k$  is 0.



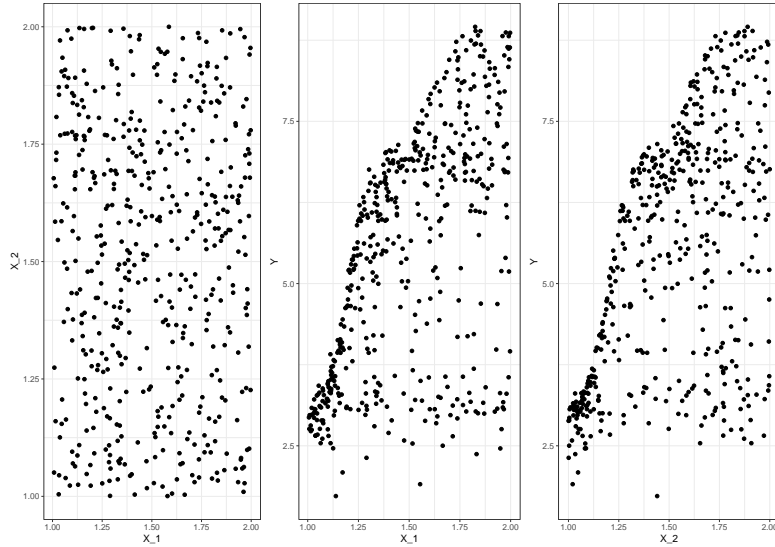


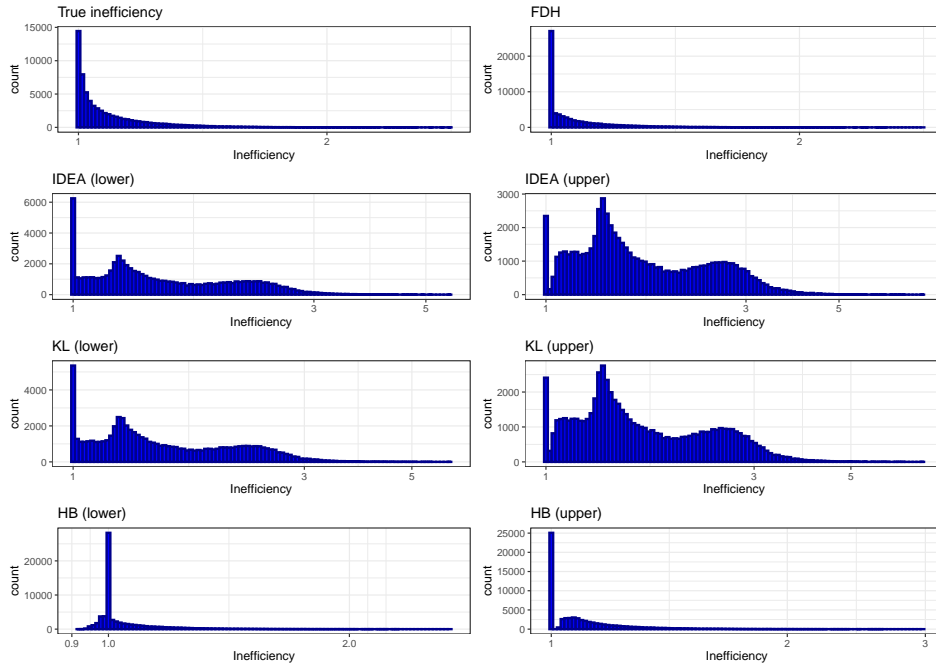
Figure 3: Example of generated data for  $K = 500$  with  $\sigma_\theta = 0.1$  and  $\sigma_y = 0.1$ .

In each of these scenarios we consider sample sizes  $K = \{50, 100, 500\}$  and replicate every simulation 100 times. The data generation process only adds inefficiency in the output, so we compute output-oriented efficiency scores. Finally, for both  $\alpha$ -level approaches we calculate efficiency bounds for  $\alpha = \{0.0, 0.5, 1.0\}$  and average the obtained bounds. We use  $\psi = 0.4$  for the HB method unless stated otherwise.

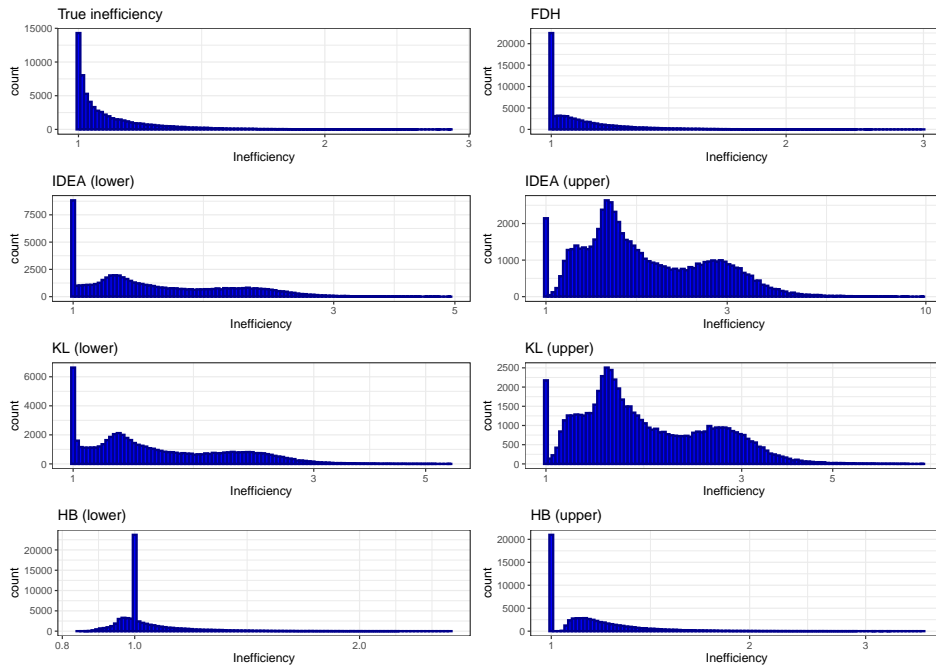
## 4.2 Results

We first consider the distribution of the true inefficiency  $\tilde{\theta}_k$  and the computed inefficiency bounds. Figures 4-5 show histograms on the true and the computed inefficiency bounds. The top left figure shows a histogram of the true inefficiency  $\tilde{\theta}_k$  over all 100 repetitions. The top right shows a histogram of  $\theta_k$  computed using conventional FDH. The other methods calculate inefficiency bounds, so we plot 2 histograms for every method: one for the lower bound (left) and another for the upper bound (right). First of all, by comparing the top left and top right figures it is clear that the obtained distribution with conventional FDH is very similar to the distribution of the true inefficiency in shape, but that it has a much longer tail with much larger inefficiency scores. Thus, uncertainty in the data can have a very large impact on the computed inefficiency scores.

Figures 6-7 show boxplots of the absolute estimation error for every method (except for conventional FDH) and Table 1 contains summary statistics. For the methods that compute efficiency bounds (all except the conventional FDH), we compute this estima-

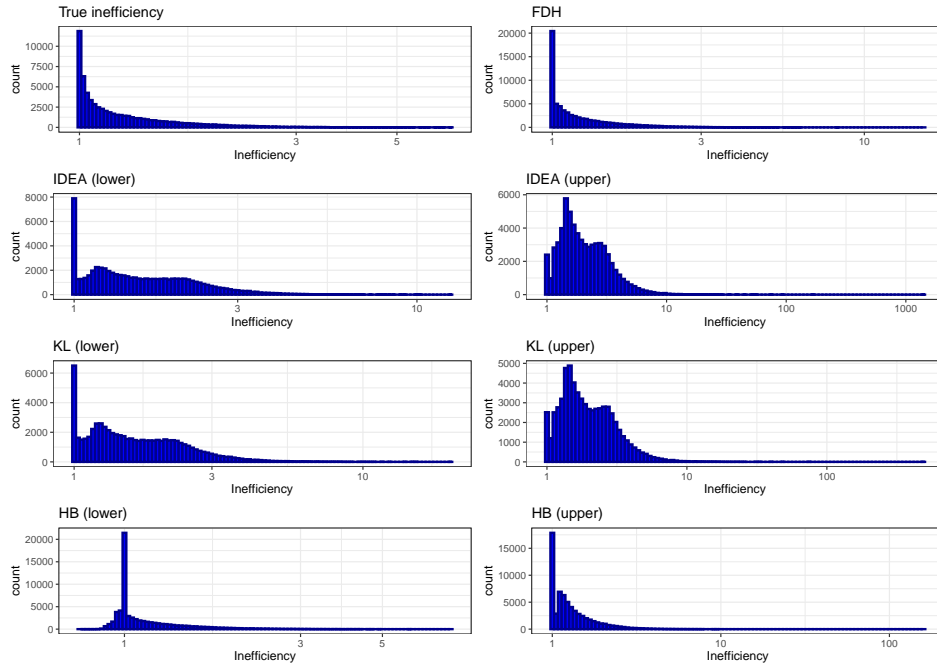


(a) Simulation with  $\sigma_\theta = 0.1$  and  $\sigma_y = 0.1$ .

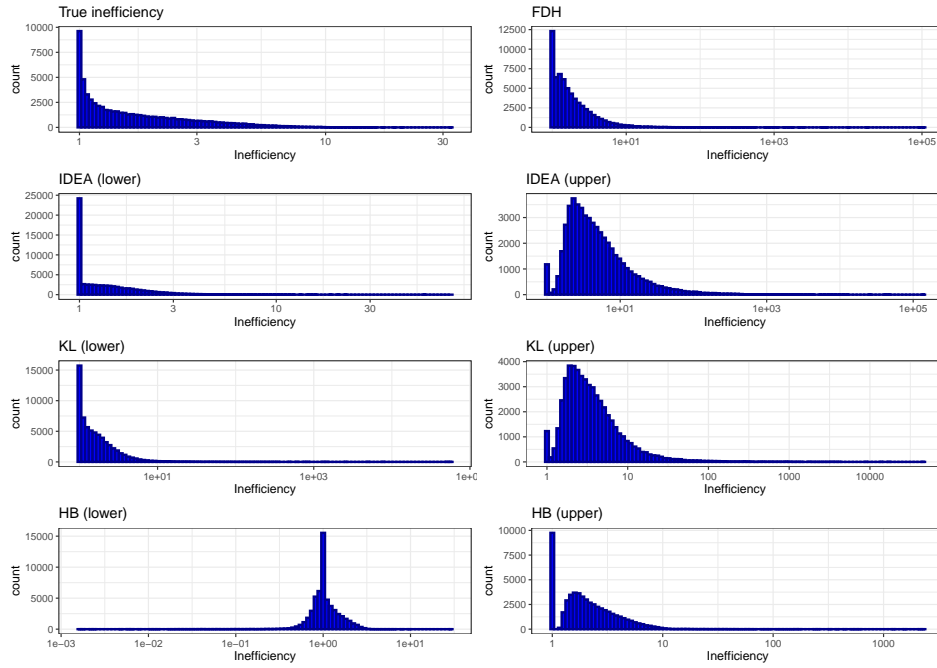


(b) Simulation with  $\sigma_\theta = 0.1$  and  $\sigma_y = 0.2$ .

Figure 4: Histogram of estimated inefficiency scores over all 100 repetitions.



(a) Simulation with  $\sigma_\theta = 0.3$  and  $\sigma_y = 0.2$ .



(b) Simulation with  $\sigma_\theta = 1.0$  and  $\sigma_y = 0.9$ .

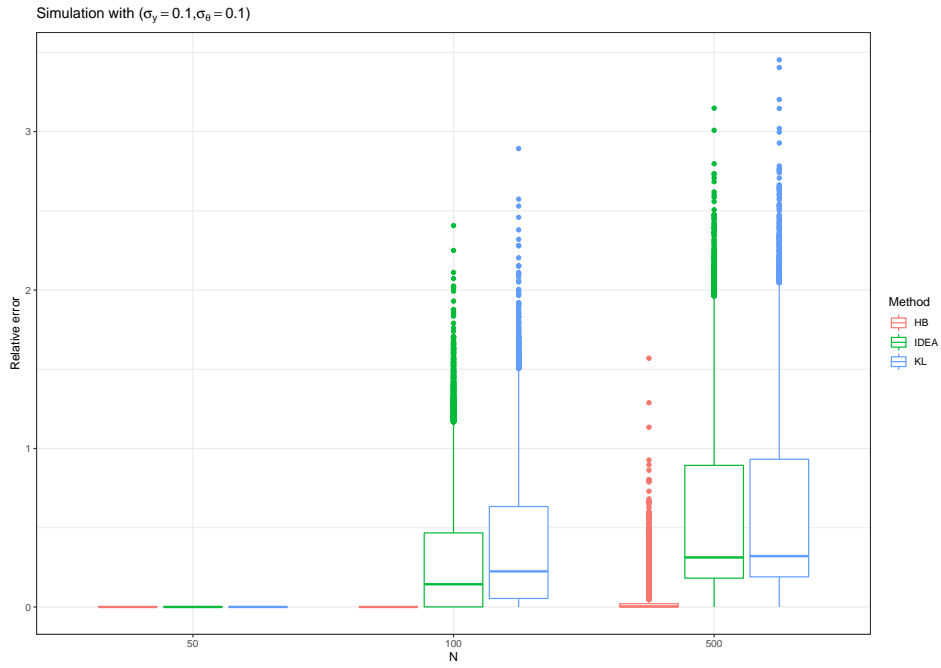
Figure 5: Histogram of estimated inefficiency scores over all 100 repetitions.

tion error as follows: if the computed efficiency bounds contain  $\tilde{\theta}_k$  then the estimation error is 0; Otherwise, we set the estimation error to  $\min \left\{ |\theta_k^L - \tilde{\theta}_k|, |\theta_k^U - \tilde{\theta}_k| \right\}$ . We normalize this absolute estimation error by dividing by the true efficiency  $\tilde{\theta}_k$ . From the figures and the table we conclude the following. First of all, conventional FDH performs quite good in general. However, as the noise level increases its error also increases. This is particularly so for the maximum error. Second, it is clear that the HB method performs best in all but one simulation ( $\sigma_\theta = 1.0, \sigma_y = 0.9$  is the exception): i.e., the average error, standard deviation of the error and maximum error are all lowest compared to the IDEA and the KL method. Third, both the IDEA and the KL method perform quite similar with IDEA performing better in some cases and worse in others.

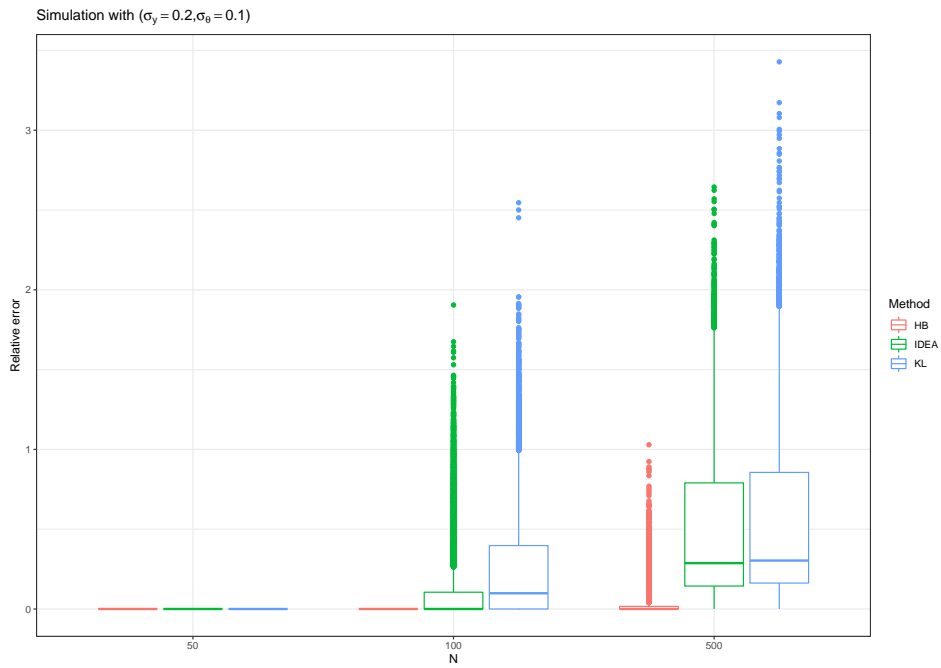
The HB method has a parameter  $\psi \in (0, 1]$  that determines the lowest acceptable level of partial dominance in any dimension. We investigate the impact of this parameter on the estimation error in Figure 8a. This figure shows the results of a simulation with  $\sigma_\theta = 1.0$  and  $\sigma_y = 0.9$  for various choices of  $\psi$  over 100 repetitions. As the figure shows, increasing  $\psi$  increases the likelihood of larger estimation errors (i.e. the area near 0 shrinks and the tails fatten slightly). This is expected, because a higher  $\psi$  reduces the number of dominating peers.

All the considered methods (except conventional FDH) compute efficiency bounds. Besides the estimation error, it is also important to consider the width of these efficiency bounds: a method can have a low estimation error simply because the computed efficiency bounds are very wide. Table 2 contains the summary statistics on the width of the obtained efficiency bounds relative to the true efficiency (i.e.,  $(\theta_k^U - \theta_k^L)/\tilde{\theta}_k$ ). It is clear from this table that the IDEA method performs well in terms of accuracy because its computed bounds are wide. In fact, they are widest of the three considered methods. Here, the HB method performs best with bounds that are much more narrow than the other methods. The performance of the KL method is similar to the IDEA method, but yields more narrow bounds in general and significantly more narrow in the worst case. The impact of  $\psi$  in the HB method on the relative width of the bounds is shown in Figure 8b. Increasing  $\psi$  seems to shrink the bounds.

Finally, we consider the computational time in Table 3. There are no surprises here: the HB method is much slower compared to the other methods which scale linearly with the number of observations. Furthermore, the high standard deviation (SD) also indicates that the computation time of the HB method not only depends on the number of observations but also depends a lot on the particular data set. Figure 9 visualizes the computation time per observation as a function of  $K$  over all simulations. It also shows the result of regressing  $K$  on the computation time per observation for each method.

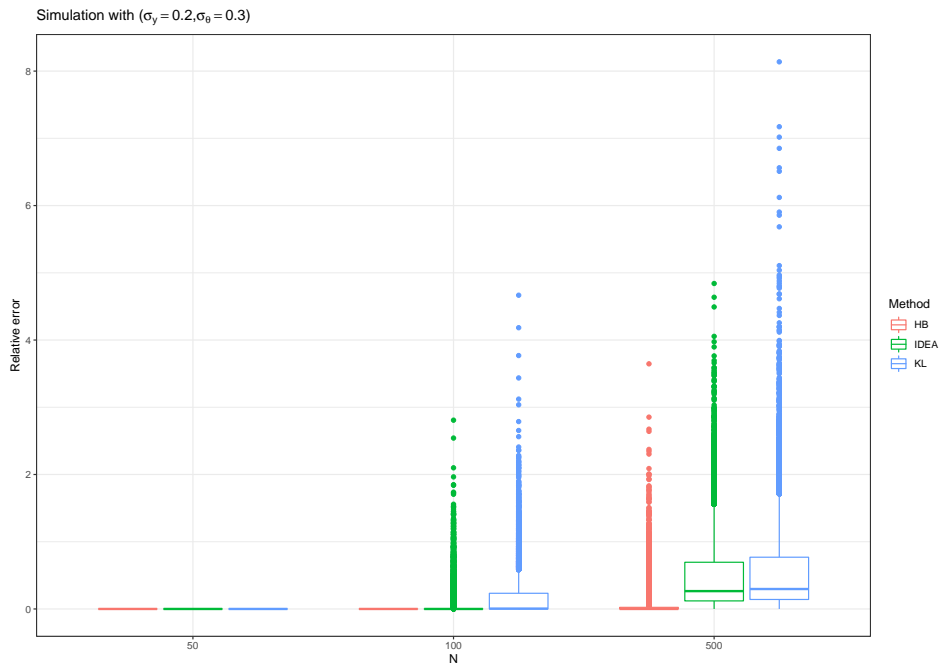


(a) Simulation with  $\sigma_\theta = 0.1$  and  $\sigma_y = 0.1$

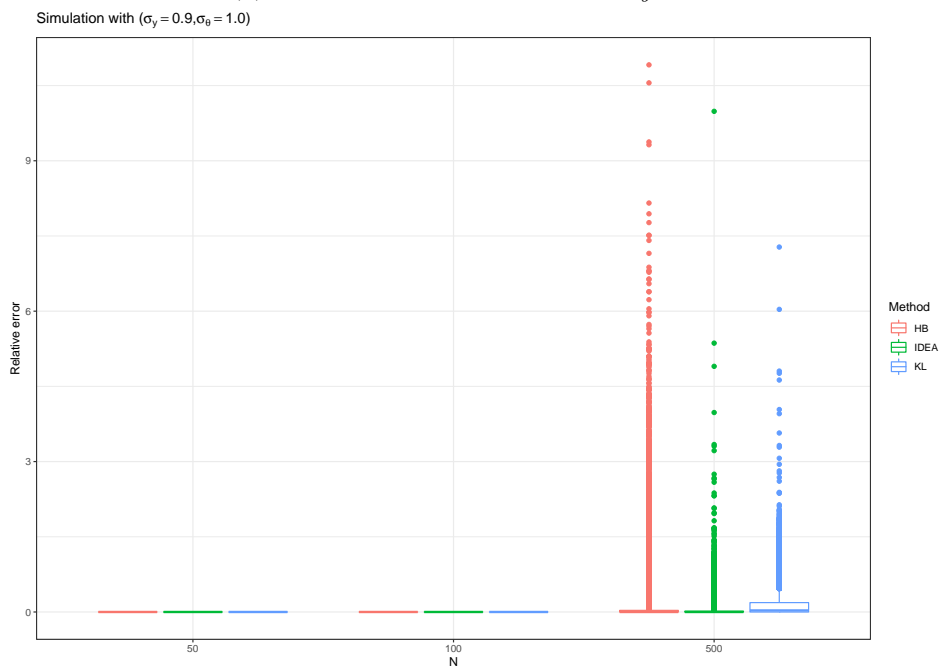


(b) Simulation with  $\sigma_\theta = 0.1$  and  $\sigma_y = 0.2$

Figure 6: Boxplots of the absolute estimation error relative to  $\tilde{\theta}_k$  over all 100 repetitions.



(a) Simulation with  $\sigma_\theta = 0.3$  and  $\sigma_y = 0.2$

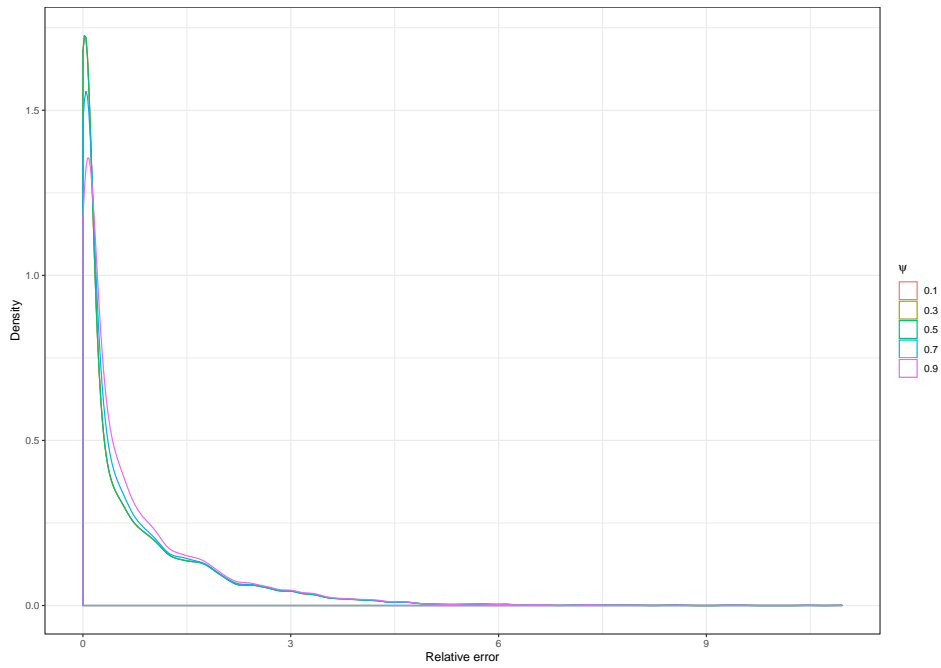


(b) Simulation with  $\sigma_\theta = 1.0$  and  $\sigma_y = 0.9$

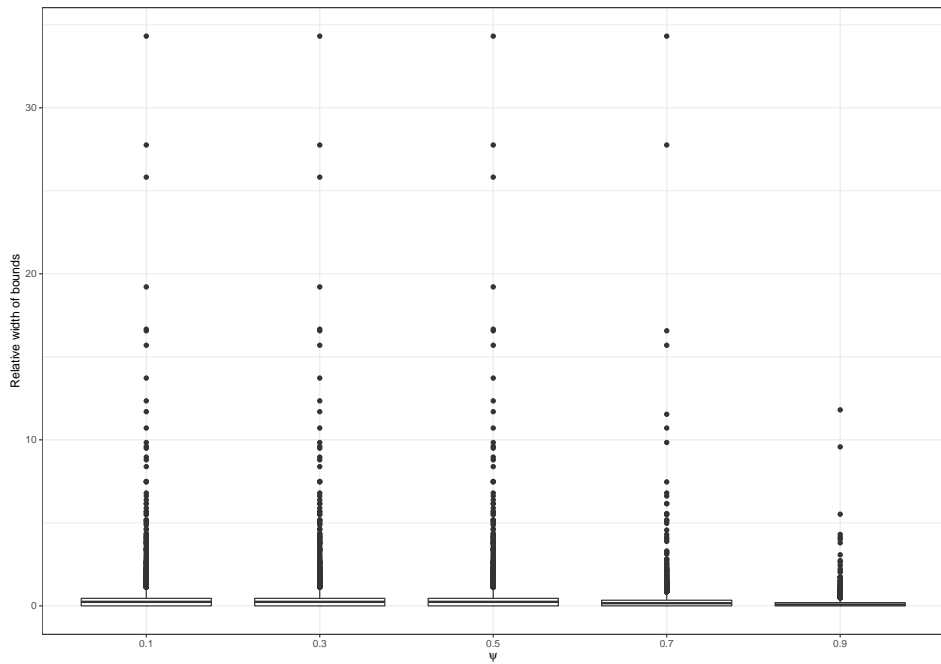
Figure 7: Boxplots of the absolute estimation error relative to  $\tilde{\theta}_k$  over all 100 repetitions.

| Simulation with $\sigma_\theta = 0.1$ and $\sigma_y = 0.1$ |        |       |         |          |        |          |          |       |
|--|--------|-------|---------|----------|--------|----------|----------|-------|
| N  | Method | nobs  | Mean    | SD       | Min    | Pctl(25) | Pctl(75) | Max   |
| 50   | FDH    | 5000  | -0.0470 | 0.0617   | -0.611 | -0.0671  | -0.005   | 0.095 |
|  | IDEA   | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | KL     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | HB     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 100  | FDH    | 10000 | -0.0337 | 0.0514   | -0.514 | -0.0527  | -0.0018  | 0.176 |
|  | IDEA   | 10000 | 0.3083  | 0.3929   | 0      | 3e-04    | 0.467    | 2.41  |
|  | KL     | 10000 | 0.3991  | 0.4319   | 0      | 0.0536   | 0.634    | 2.89  |
|  | HB     | 10000 | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 500  | FDH    | 50000 | -0.0089 | 0.0353   | -0.452 | -0.0224  | 0.0065   | 0.217 |
|  | IDEA   | 50000 | 0.5348  | 0.4728   | 0      | 0.182    | 0.894    | 3.15  |
|  | KL     | 50000 | 0.5564  | 0.5001   | 0      | 0.19     | 0.932    | 3.45  |
|  | HB     | 50000 | 0.0196  | 0.0456   | 0      | 0        | 0.0205   | 1.57  |
| Simulation with $\sigma_\theta = 0.1$ and $\sigma_y = 0.2$ |        |       |         |          |        |          |          |       |
| N  | Method | nobs  | Mean    | SD       | Min    | Pctl(25) | Pctl(75) | Max   |
| 50   | FDH    | 5000  | -0.0430 | 0.0683   | -0.457 | -0.0687  | -0.0025  | 0.271 |
|  | IDEA   | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | KL     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | HB     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 100  | FDH    | 10000 | -0.0226 | 0.0637   | -0.501 | -0.0468  | 0        | 0.443 |
|  | IDEA   | 10000 | 0.1235  | 0.2554   | 0      | 0        | 0.105    | 1.9   |
|  | KL     | 10000 | 0.2709  | 0.3784   | 0      | 0        | 0.397    | 2.55  |
|  | HB     | 10000 | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 500  | FDH    | 50000 | 0.0107  | 0.0569   | -0.52  | -0.0165  | 0.0365   | 0.452 |
|  | IDEA   | 50000 | 0.4753  | 0.4229   | 0      | 0.144    | 0.79     | 2.64  |
|  | KL     | 50000 | 0.5187  | 0.4731   | 0      | 0.162    | 0.856    | 3.43  |
|  | HB     | 50000 | 0.0174  | 0.0442   | 0      | 0        | 0.0164   | 1.03  |
| Simulation with $\sigma_\theta = 0.3$ and $\sigma_y = 0.2$ |        |       |         |          |        |          |          |       |
| N  | Method | nobs  | Mean    | SD       | Min    | Pctl(25) | Pctl(75) | Max   |
| 50   | FDH    | 5000  | -0.0808 | 0.1093   | -0.743 | -0.125   | -0.0078  | 0.504 |
|  | IDEA   | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | KL     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | HB     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 100  | FDH    | 10000 | -0.0520 | 0.0962   | -0.749 | -0.0869  | -0.0011  | 0.741 |
|  | IDEA   | 10000 | 0.0566  | 0.1837   | 0      | 0        | 0        | 2.81  |
|  | KL     | 10000 | 0.1928  | 0.3515   | 0      | 0        | 0.233    | 4.67  |
|  | HB     | 10000 | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 500  | FDH    | 50000 | -0.0053 | 0.0772   | -0.825 | -0.0363  | 0.028    | 1.9   |
|  | IDEA   | 49995 | 0.4452  | 0.4362   | 0      | 0.12     | 0.695    | 4.84  |
|  | KL     | 49995 | 0.5058  | 0.5171   | 0      | 0.141    | 0.77     | 8.14  |
|  | HB     | 50000 | 0.0325  | 0.1016   | 0      | 0        | 0.024    | 3.65  |
| Simulation with $\sigma_\theta = 1.0$ and $\sigma_y = 0.9$ |        |       |         |          |        |          |          |       |
| N  | Method | nobs  | Mean    | SD       | Min    | Pctl(25) | Pctl(75) | Max   |
| 50   | FDH    | 4892  | 0.2068  | 5.9712   | -0.929 | -0.231   | 0.0474   | 283   |
|  | IDEA   | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | KL     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | HB     | 5000  | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 100  | FDH    | 9817  | 0.4759  | 19.3308  | -0.905 | -0.155   | 0.161    | 1860  |
|  | IDEA   | 10000 | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | KL     | 10000 | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
|  | HB     | 10000 | 0.0000  | 0.0000   | 0      | 0        | 0        | 0     |
| 500  | FDH    | 49138 | 2.8491  | 357.4243 | -0.915 | -0.0358  | 0.343    | 63800 |
|  | IDEA   | 38420 | 0.0478  | 0.1467   | 0      | 0        | 0.0162   | 9.99  |
|  | KL     | 38420 | 0.1449  | 0.2544   | 0      | 0        | 0.187    | 7.28  |
|  | HB     | 41345 | 0.1116  | 0.4355   | 0      | 0        | 0.0324   | 10.9  |

Table 1: Summary statistics of the estimation error relative to  $\tilde{\theta}_k$  over all 100 repetitions.



(a) Absolute estimation error relative to  $\tilde{\theta}_k$  as a function of  $\psi$ .



(b) Relative width of bounds as a function of  $\psi$ .

Figure 8: Impact of  $\psi$  on estimation error and relative width of bounds.



| Simulation with $\sigma_\theta = 0.1$ and $\sigma_y = 0.1$ |        |       |        |          |       |          |          |       |
|--|--------|-------|--------|----------|-------|----------|----------|-------|
| N  | Method | nobs  | Mean   | SD       | Min   | Pctl(25) | Pctl(75) | Max   |
| 50   | IDEA   | 5000  | 0.1549 | 0.1194   | 0     | 0.0875   | 0.227    | 0.6   |
|  | KL     | 5000  | 0.1063 | 0.0775   | 0     | 0.0672   | 0.156    | 0.364 |
|  | HB     | 5000  | 0.0230 | 0.0385   | 0     | 0        | 0.0491   | 0.158 |
| 100  | IDEA   | 10000 | 0.1960 | 0.1327   | 0     | 0.109    | 0.305    | 0.757 |
|  | KL     | 10000 | 0.1321 | 0.0824   | 0     | 0.0789   | 0.203    | 0.434 |
|  | HB     | 10000 | 0.0345 | 0.0457   | 0     | 0        | 0.0602   | 0.195 |
| 500  | IDEA   | 50000 | 0.2130 | 0.1308   | 0     | 0.124    | 0.306    | 0.964 |
|  | KL     | 50000 | 0.1428 | 0.0792   | 0     | 0.0892   | 0.204    | 0.548 |
|  | HB     | 50000 | 0.0538 | 0.0447   | 0     | 0        | 0.0771   | 0.233 |
| Simulation with $\sigma_\theta = 0.1$ and $\sigma_y = 0.2$ |        |       |        |          |       |          |          |       |
| N  | Method | nobs  | Mean   | SD       | Min   | Pctl(25) | Pctl(75) | Max   |
| 50   | IDEA   | 5000  | 0.3141 | 0.2456   | 0     | 0.151    | 0.455    | 1.45  |
|  | KL     | 5000  | 0.2146 | 0.1577   | 0     | 0.119    | 0.311    | 0.833 |
|  | HB     | 5000  | 0.0546 | 0.0837   | 0     | 0        | 0.107    | 0.358 |
| 100  | IDEA   | 10000 | 0.3982 | 0.2813   | 0     | 0.219    | 0.61     | 2.14  |
|  | KL     | 10000 | 0.2669 | 0.1726   | 0     | 0.158    | 0.404    | 1.2   |
|  | HB     | 10000 | 0.0806 | 0.0971   | 0     | 0        | 0.129    | 0.687 |
| 500  | IDEA   | 50000 | 0.4328 | 0.2770   | 0     | 0.247    | 0.615    | 2.35  |
|  | KL     | 50000 | 0.2884 | 0.1656   | 0     | 0.177    | 0.406    | 1.29  |
|  | HB     | 50000 | 0.1208 | 0.0904   | 0     | 0        | 0.175    | 0.557 |
| Simulation with $\sigma_\theta = 0.3$ and $\sigma_y = 0.2$ |        |       |        |          |       |          |          |       |
| N  | Method | nobs  | Mean   | SD       | Min   | Pctl(25) | Pctl(75) | Max   |
| 50   | IDEA   | 5000  | 0.3393 | 0.3171   | 0     | 0.15     | 0.489    | 6.92  |
|  | KL     | 5000  | 0.2260 | 0.1863   | 0     | 0.119    | 0.329    | 3.07  |
|  | HB     | 5000  | 0.0780 | 0.1040   | 0     | 0        | 0.135    | 1.19  |
| 100  | IDEA   | 10000 | 0.4340 | 0.3652   | 0     | 0.23     | 0.621    | 5.69  |
|  | KL     | 10000 | 0.2837 | 0.2084   | 0     | 0.165    | 0.408    | 2.75  |
|  | HB     | 10000 | 0.1043 | 0.1125   | 0     | 0        | 0.165    | 0.989 |
| 500  | IDEA   | 49995 | 0.4883 | 1.5507   | 0     | 0.257    | 0.655    | 335   |
|  | KL     | 49995 | 0.3142 | 0.5537   | 0     | 0.182    | 0.426    | 114   |
|  | HB     | 50000 | 0.1387 | 0.1765   | -15.7 | 0.0943   | 0.201    | 27.1  |
| Simulation with $\sigma_\theta = 1.0$ and $\sigma_y = 0.9$ |        |       |        |          |       |          |          |       |
| N  | Method | nobs  | Mean   | SD       | Min   | Pctl(25) | Pctl(75) | Max   |
| 50   | IDEA   | 4080  | 6.6226 | 83.0191  | 0     | 0.762    | 2.89     | 4940  |
|  | KL     | 4080  | 2.7132 | 27.7448  | 0     | 0.512    | 1.66     | 1650  |
|  | HB     | 4652  | 0.6776 | 18.4334  | -669  | 0        | 0.76     | 990   |
| 100  | IDEA   | 8096  | 6.2900 | 43.3440  | 0     | 0.976    | 3.49     | 1970  |
|  | KL     | 8096  | 2.6989 | 14.5759  | 0     | 0.676    | 1.94     | 658   |
|  | HB     | 8926  | 0.8174 | 12.4030  | -172  | 0        | 0.907    | 782   |
| 500  | IDEA   | 41244 | 9.6105 | 225.3891 | 0     | 1.25     | 3.98     | 31100 |
|  | KL     | 41244 | 3.8847 | 75.1663  | 0     | 0.837    | 2.18     | 10400 |
|  | HB     | 42767 | 0.7947 | 12.6804  | -2100 | 0.487    | 1.03     | 1340  |

Table 2: Relative width of efficiency bounds (i.e.,  $(\theta_k^U - \theta_k^L)/\tilde{\theta}_k$ ) over all 100 repetitions.

Notice in particular the large variation in computation time within a fixed  $K$  of the HB method which again underscores our earlier remark that the computation time depends on the particular data set. Nevertheless, the computation time for this method never exceeds 5.3 seconds per observation.

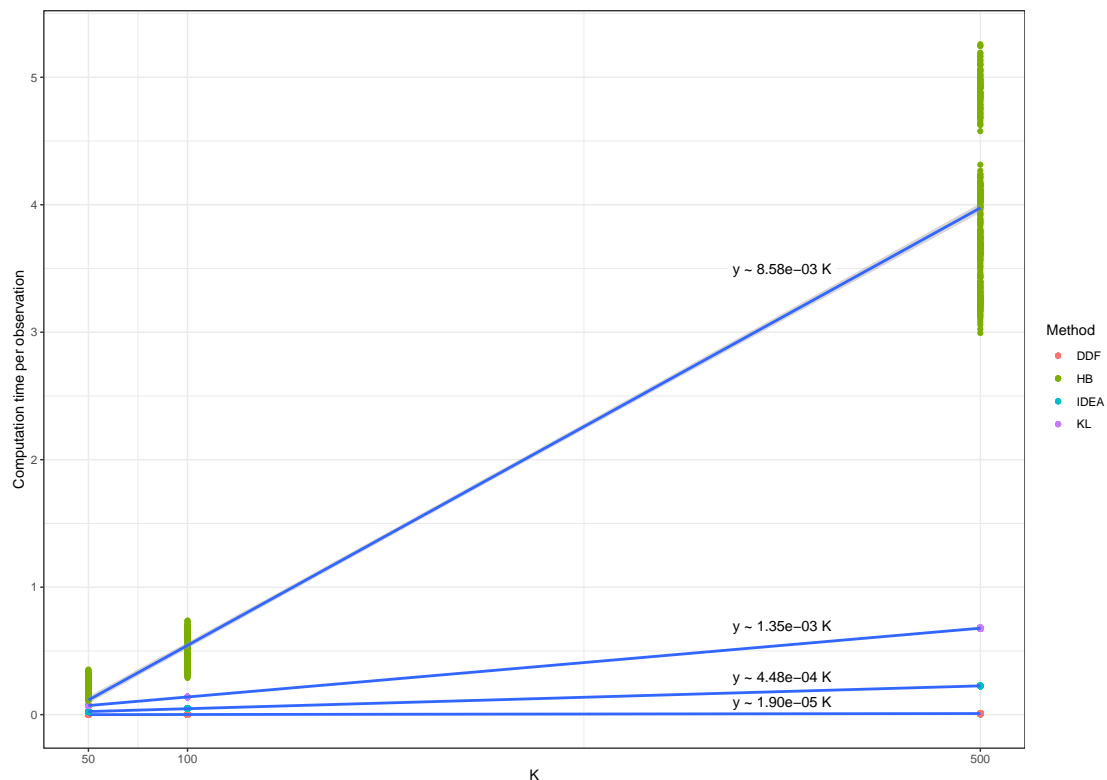


Figure 9: Computation time per observation over all simulations.

## 5 Conclusions

Common for the methods we consider in this paper is that they construct efficiency bounds instead of one crisp efficiency score. We compare these methods on three different aspects (i.e., accuracy, width of the bounds and computation time) and find that the HB method of [Hougaard and Baležentis \(2014\)](#) clearly outperforms the other two methods on accuracy and width of the bounds aspects. We also propose a new approach to computing the key component and computational intensive interval-probabilities in the HB method, which relies on numerical integration. This significantly speeds-up the necessary computations. Nevertheless, the method remains the slowest among those

| Simulation with $\sigma_\theta = 0.1$ and $\sigma_y = 0.1$ |        |      |           |         |      |          |          |      |
|--|--------|------|-----------|---------|------|----------|----------|------|
| N  | Method | nobs | Mean      | SD      | Min  | Pctl(25) | Pctl(75) | Max  |
| 50   | FDH    | 100  | 0.0190    | 0.0104  | 0.01 | 0.01     | 0.02     | 0.08 |
|  | IDEA   | 100  | 1.1942    | 0.0313  | 1.15 | 1.17     | 1.2      | 1.36 |
|  | KL     | 100  | 3.5693    | 0.0554  | 3.5  | 3.54     | 3.58     | 3.85 |
|  | HB     | 100  | 7.0359    | 0.7058  | 5.35 | 6.52     | 7.53     | 8.63 |
| 100  | FDH    | 100  | 0.0891    | 0.0085  | 0.07 | 0.08     | 0.1      | 0.1  |
|  | IDEA   | 100  | 4.6778    | 0.0565  | 4.54 | 4.64     | 4.71     | 4.82 |
|  | KL     | 100  | 13.7760   | 0.0668  | 13.7 | 13.7     | 13.8     | 14   |
|  | HB     | 100  | 34.7999   | 2.3679  | 28.8 | 33.2     | 36.3     | 41.2 |
| 500  | FDH    | 100  | 5.5388    | 0.1384  | 5.23 | 5.43     | 5.64     | 5.88 |
|  | IDEA   | 100  | 112.8520  | 0.4694  | 112  | 113      | 113      | 114  |
|  | KL     | 100  | 339.0633  | 1.1480  | 336  | 338      | 340      | 341  |
|  | HB     | 100  | 1633.7511 | 47.9344 | 1500 | 1610     | 1660     | 1760 |
| Simulation with $\sigma_\theta = 0.1$ and $\sigma_y = 0.2$ |        |      |           |         |      |          |          |      |
| N  | Method | nobs | Mean      | SD      | Min  | Pctl(25) | Pctl(75) | Max  |
| 50   | FDH    | 100  | 0.0175    | 0.0081  | 0.01 | 0.01     | 0.02     | 0.04 |
|  | IDEA   | 100  | 1.1908    | 0.0268  | 1.15 | 1.17     | 1.2      | 1.34 |
|  | KL     | 100  | 3.5620    | 0.0468  | 3.49 | 3.54     | 3.58     | 3.92 |
|  | HB     | 100  | 7.7758    | 0.6794  | 6.4  | 7.22     | 8.31     | 9.32 |
| 100  | FDH    | 100  | 0.0838    | 0.0086  | 0.07 | 0.08     | 0.09     | 0.1  |
|  | IDEA   | 100  | 4.6795    | 0.0737  | 4.54 | 4.63     | 4.73     | 4.84 |
|  | KL     | 100  | 13.7757   | 0.0670  | 13.6 | 13.7     | 13.8     | 13.9 |
|  | HB     | 100  | 38.2494   | 2.3104  | 33.5 | 36.6     | 40.2     | 43.1 |
| 500  | FDH    | 100  | 5.2294    | 0.1498  | 4.9  | 5.15     | 5.32     | 5.77 |
|  | IDEA   | 100  | 112.8547  | 0.6106  | 112  | 112      | 113      | 116  |
|  | KL     | 100  | 339.0549  | 1.2678  | 336  | 338      | 340      | 341  |
|  | HB     | 100  | 1840.9044 | 47.0419 | 1720 | 1810     | 1870     | 1960 |
| Simulation with $\sigma_\theta = 0.3$ and $\sigma_y = 0.2$ |        |      |           |         |      |          |          |      |
| N  | Method | nobs | Mean      | SD      | Min  | Pctl(25) | Pctl(75) | Max  |
| 50   | FDH    | 100  | 0.0196    | 0.0090  | 0.01 | 0.01     | 0.02     | 0.06 |
|  | IDEA   | 100  | 1.1883    | 0.0270  | 1.14 | 1.17     | 1.2      | 1.34 |
|  | KL     | 100  | 3.5671    | 0.0486  | 3.5  | 3.54     | 3.58     | 3.93 |
|  | HB     | 100  | 9.8604    | 0.9849  | 7.62 | 9.24     | 10.3     | 12.9 |
| 100  | FDH    | 100  | 0.0777    | 0.0080  | 0.06 | 0.08     | 0.08     | 0.1  |
|  | IDEA   | 100  | 4.6947    | 0.0762  | 4.54 | 4.64     | 4.75     | 4.87 |
|  | KL     | 100  | 13.7827   | 0.0639  | 13.6 | 13.7     | 13.8     | 13.9 |
|  | HB     | 100  | 46.1235   | 2.8282  | 39.1 | 43.9     | 47.8     | 54.2 |
| 500  | FDH    | 100  | 4.4429    | 0.1674  | 4.04 | 4.35     | 4.54     | 4.99 |
|  | IDEA   | 100  | 112.7903  | 0.5403  | 112  | 112      | 113      | 114  |
|  | KL     | 100  | 339.0430  | 1.3611  | 336  | 338      | 340      | 342  |
|  | HB     | 100  | 2040.8704 | 42.6421 | 1930 | 2010     | 2070     | 2160 |
| Simulation with $\sigma_\theta = 1.0$ and $\sigma_y = 0.9$ |        |      |           |         |      |          |          |      |
| N  | Method | nobs | Mean      | SD      | Min  | Pctl(25) | Pctl(75) | Max  |
| 50   | FDH    | 100  | 0.0173    | 0.0068  | 0.01 | 0.01     | 0.02     | 0.04 |
|  | IDEA   | 100  | 1.2169    | 0.0215  | 1.18 | 1.2      | 1.23     | 1.31 |
|  | KL     | 100  | 3.6221    | 0.0450  | 3.52 | 3.59     | 3.64     | 3.85 |
|  | HB     | 100  | 13.9948   | 1.4879  | 10.6 | 13       | 15.1     | 17.7 |
| 100  | FDH    | 100  | 0.0598    | 0.0112  | 0.03 | 0.05     | 0.07     | 0.08 |
|  | IDEA   | 100  | 4.7721    | 0.0747  | 4.61 | 4.72     | 4.82     | 4.97 |
|  | KL     | 100  | 13.9592   | 0.0644  | 13.8 | 13.9     | 14       | 14.1 |
|  | HB     | 100  | 62.7362   | 5.0362  | 51.5 | 60       | 66.2     | 73.8 |
| 500  | FDH    | 100  | 2.1834    | 0.4372  | 1.03 | 1.9      | 2.49     | 3.34 |
|  | IDEA   | 100  | 113.0578  | 0.5079  | 112  | 113      | 113      | 115  |
|  | KL     | 100  | 339.4892  | 1.2626  | 336  | 339      | 340      | 342  |
|  | HB     | 100  | 2454.8042 | 77.0949 | 2290 | 2400     | 2500     | 2630 |

Table 3: Summary statistics on the computation time (in seconds) over all 100 repetitions.

tested. Depending on the specific application, this method is therefore certainly an attractive alternative if computation time is less of a concern.

## References

- BOGETOFT, P. AND L. OTTO (2010): *Benchmarking with Dea, Sfa, and R*, vol. 157, Springer Science & Business Media.
- COOPER, W. W., K. S. PARK, AND G. YU (1999): “IDEA and AR-IDEA: Models for dealing with imprecise data in DEA,” *Management science*, 45, 597–607.
- DESPOTIS, D. K. AND Y. G. SMIRLIS (2002): “Data envelopment analysis with imprecise data,” *European Journal of Operational Research*, 140, 24–36.
- EMROUZNEJAD, A. AND G.-L. YANG (2018): “A survey and analysis of the first 40 years of scholarly literature in DEA: 1978–2016,” *Socio-Economic Planning Sciences*, 61, 4–8.
- GORISSEN, B. L., İ. YANIKOĞLU, AND D. DEN HERTOĞ (2015): “A practical guide to robust optimization,” *Omega*, 53, 124–137.
- HATAMI-MARBINI, A., A. EMROUZNEJAD, AND M. TAVANA (2011): “A taxonomy and review of the fuzzy data envelopment analysis literature: two decades in the making,” *European journal of operational research*, 214, 457–472.
- HOUGAARD, J. L. (1999): “Fuzzy scores of technical efficiency,” *European Journal of Operational Research*, 115, 529–541.
- (2005): “A simple approximation of productivity scores of fuzzy production plans,” *Fuzzy Sets and Systems*, 152, 455–465.
- HOUGAARD, J. L. AND T. BALEŽENTIS (2014): “Fuzzy efficiency without convexity,” *Fuzzy Sets and Systems*, 255, 17–29.
- KAO, C. AND S.-T. LIU (2000): “Fuzzy efficiency measures in data envelopment analysis,” *Fuzzy Sets and Systems*, 113, 427–437.
- PARK, K. (2004): “Simplification of the transformations and redundancy of assurance regions in IDEA (imprecise DEA),” *Journal of the Operational Research Society*, 55, 1363–1366.

- (2007): “Efficiency bounds and efficiency classifications in DEA with imprecise data,” *Journal of the Operational Research Society*, 58, 533–540.
- SENGUPTA, J. K. (1992): “A fuzzy systems approach in data envelopment analysis,” *Computers & Mathematics with Applications*, 24, 259–266.
- TRIANTIS, K. AND O. GIROD (1998): “A mathematical programming approach for measuring technical efficiency in a fuzzy environment,” *Journal of Productivity Analysis*, 10, 85–102.
- TULKENS, H. (1993): “On FDH efficiency analysis: some methodological issues and applications to retail banking courts, and urban transit,” *Journal of Productivity Analysis*, 4, 183.