Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2018) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

World Conference on Transport Research - WCTR 2019 Mumbai 26-31 May 2019

Hybrid Data Envelopment Analysis for Large-Scale Smartphone Data Modeling

Dimitrios I. Tselentis*, Eleni I. Vlahogianni, George Yannis, Loukas Kavouras

National Technical University of Athens, 5, Iroon Polytechniou str. GR-15773, Athens, Greece

Abstract

This paper deals with the problem of improving the existing optimization techniques for Data Envelopment Analysis (DEA). The algorithm proposed herein is a combination of the "quickhull algorithm" and a DEA algorithm written in Python programming language. To the best of the authors' knowledge no prior effort has been made to date to propose a methodology for reducing the running time of a DEA problem that incorporates multiple inputs and outputs. The algorithmic implementation is applied on the existing problem of driving efficiency evaluation by exploiting a driving data sample of 10,088 trips collected from smartphone devices. Results indicate that the proposed algorithm is performing relatively well for Big Data compared to other existing DEA algorithmic methodologies that yield the same optimal solution such as Standard DEA and RBE DEA methodologies. The results obtained are calculated for the test sets of 100, 500, 1000, 5,000 and 10,088 Decision-Making-Units (DMUs) and compared in terms of running time of each of the algorithms applied. The results of per trip analysis can be exploited in order to classify trips into different efficiency categories (such as efficient, less efficient, non-efficient) and present their main characteristics.

Peer-review under responsibility of WORLD CONFERENCE ON TRANSPORT RESEARCH SOCIETY.

Keywords: Data Envelopment Analysis; Convex Hull; Driving Efficiency; Smartphone Data

1. Introduction

Data envelopment Analysis (DEA) is a technique that has been exhaustively applied in literature (Cook and Seiford (2009), Emrouznejad et al., 2008, Hollingsworth et al., 1999) to evaluate the efficiency of Decision-Making Unit (DMU) mainly in scientific fields such as economics, management and health. It has also been used in assessing public transportation system performance (Karlaftis et al., 2013), as well as traffic safety (Shen et al., 2011, Egilmez and McAvoy 2013, Alper et al., 2015), but never before in evaluating driver's behavior.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: author@institute.xxx

Peer-review under responsibility of WORLD CONFERENCE ON TRANSPORT RESEARCH SOCIETY

Rapid technological progress, especially in telematics and Big Data analytics, along with the increase in the information technologies' penetration and use by drivers (e.g. smartphones), provide unprecedented opportunities to accurately monitor and collect large-scale data on driving behavior. As a result, it becomes even more necessary for scientists to come across a practical solution to tackle the problem of analyzing large databases using optimization techniques to support policy-makers and stakeholders (Vlahogianni 2015).

One of the most important issue arising is that linear programming techniques, such as DEA requires a significant amount of time to perform on a database of a large-scale. Many suggestions including Reduced Basis Entry (RBE) and Early Identification of Efficient DMUs (EIE) have been made to date to reduce the running time of DEA with some of them performing notably better (Dulá, 2008, Barr, 1997, Ali, 1993, Dulá and López, 2009).

This paper addresses the multiple inputs and multiple outputs DEA problem for large-scale data by proposing an algorithmic modification of DEA based on computational geometry. The approach is based on the "quickhull" algorithm, which is the computational geometry approach of frame recognition, i.e. the convex hull, to allow for extreme points identification before applying the standard DEA approach to the whole sample. Methodologically, this paper also extends past research by implementing the proposed modified DEA in multiple inputs and outputs settings. will be implemented to test its effectiveness. The proposed approach is evaluated in terms of computation time, compared to other proposed approaches i.e. the RBE technique and the standard DEA procedure. The proposed methodology is applied to a real-life case study of 10,088 recorded trips collected from Eighty-eight (88) drivers.

2. Improving DEA computational efficiency: State-of-the-art

2.1. Linear Programming Problem Reduction Procedures

There is a considerable number of techniques that can be used to take advantage of the special structure that the DEA LPs have, apart from those known to improve LP performance (e.g. hot starts, product forms, multiple pricing etc.) in general. Among them, the two most popular are Reduced Basis Entry (RBE) and Early Identification of Efficient DMUs (EIE) (Ali, 1993). Both ideas are based on the same concept about DEA LPs i.e. that a DEA LP final solution is only influenced by efficient DMUs. The efficiency frontier consists only of those extreme data points (DMUs) whose efficiency equals to one. In other words, presence or absence of less efficient DMUs from the LP coefficient matrix has no effect on defining the optimal solution and, therefore, it would be preferable to be absent to reduce the required running time for each LP. Ideally, for each LP that is necessary to be solved to identify the efficiency index and peers of every DMU m, it is recommended to omit all DMUs apart from the efficient ones and DMU m.

The basic concept of RBE is to iterate over the entire set of DMUs and omit every DMU that appears to be inefficient from subsequent LP formulations. LPs are iteratively formulated and solved and, as a result, it is easy to implement RBE. The application of this technique reduces the size of the LPs by one every time that a non-efficient DMU is identified so that running time is reduced. EIE on the other hand provides prior knowledge that a DMU is efficient or not when a DMU's variable appears in a basis of an optimal solution of an envelopment LP, or its constraint is an equality at optimality in a multiplier form. According to (Dulá, 2008), EIE techniques appear to have significantly less influence on reducing LPs running time than RBE. This is because of the fact that in large-scale databases the ratio of the efficient DMUs to the total number of DMUs is usually small. Besides that, it is shown that a subset of the efficient DMUs can have a dominant presence in optimal bases. These two techniques have been tested together (Barr, 1997) and reports demonstrated a significant impact on computation time reduction. (Dulá, 2008) also concludes that most of improvement influence is due to RBE technique implementation.

2.2. Pre-processing methodologies

Another option for reducing computation time for DEA is pre-processing ideas several of which have been implemented in literature (Dulá and López, 2009, Dulá and Hickman, 1997). In general, when the status of one or more DMUs is determined without having to solve the entire LP problem then less time is needed and a pre-processor is deemed to be effective. Classifying DMUs is exactly what pre-processors are used for. In most cases, it is not required to solve an LP for that entity since advanced classification of an efficient DMU takes place with an

inexpensive pre-processor. Especially when estimating the efficiency score of a DMU is not the goal of a research, the entire LP solution is obviated for inefficient DMUs since they are classified by the pre-processor. Therefore, a pre-processor can be proved extremely valuable in terms of cost and effectiveness. If efficiency score is required, a pre-processor could also be exploited to reduce the cost of inefficient DMUs identification by achieving fast classification. As described above, if a DMU is inefficient, it can be omitted from the solution of the LPs that should be solved for scoring and benchmarking the entire inefficient DMUs set. As an example, in variable-returns-to-scale (VRS) model, a pre-processor could be simple sorting. Maximum and minimum attribute values of outputs and inputs of DMUs respectively of the entire data set are likely to represent efficient DMUs. More details on sorting in DEA can be found in (Dulá and Hickman, 1997).

2.3. Data partitioning approaches

The basic concept of this approach is that RBE methodologies are combined with data partitioning schemes. The fundamental idea lies on the principle that if a DMU m is found to be inefficient within a set E of DMUs, DMU m will be inefficient within any superset that includes set E. To apply this, the main dataset is partitioned into equally sized subsets and DEA LPs are solved using RBE techniques for every DMU in each of those subsets. When this procedure is over, all inefficient DMUs are identified and compose the new sets of DMUs which still are subsets of the initial database. By applying the same approach repetitively in every superset created by efficient DMUs of each subset, a final superset comprising of the efficient DMUs of the initial dataset is composed. As a second phase of the procedure, a DEA LP is solved for each of the inefficient DMU m using only all DMUs that were evaluated as efficient plus DMU m in each iteration. As described above, the advantage that this approach offers is that computation time is significantly reduced since the LP comprises of a considerable lower number of variables; especially when the number of efficient DMUs (density) is low.

The LPs arising are estimated to be much smaller than would otherwise be used in the standard approach. Nevertheless, depending on how data is partitioned, the number of efficient DMUs and a few other parameters, there is a slight chance that total required time will be more than that required if RBE without data partitioning was applied. In schemes like these, called “hierarchical decomposition” schemes, performance is affected by the size and density of the initial and intermediary sets of DMUs. According to (Barr, 1997), optimizing this procedure requires experimental tuning and results of the same research indicate a considerable improvement in computation time.

2.4. Computational geometry procedures

Computational geometry procedures are another idea that has been extensively used as an improvement to the DEA procedure to solve the problem of the exterior points frame. The most common version of frame determination is the convex hull problem that tackles the problem of extreme points identification of a finitely generated polyhedral set. Convex hull algorithm builds the frame consisting of the extreme points (the efficient DMUs) as well as the vertices and line segments that joins them two by two (without intersecting the interior of the polyhedron). In general, the convex hull of a finite point set S is the set of all convex combinations of its points. Convex hull’s equivalent problem in DEA is the one of identifying solely the efficient DMUs (Dulá and López, 2006) without having to solve the entire LP for each of the DMUs in the set. As described above, the benefit of computational geometry methods such as the convex hull approach is that the size of the new LPs to be solved are considerably smaller and the number of the LP’s variable is represented by the total number of efficient DMUs in the dataset plus one, which is the DMU examined in each iteration.

3. Methodological approach

3.1. Problem setup

The general idea of DEA is to minimize inputs (input-oriented model) or maximize the outputs of a problem (output-oriented model). More specifically, on the case study examined a trip should be longer in kilometers maintaining the same number of harsh braking or accelerating events or have a lower number of harsh braking/accelerating events for

the same mileage. From a road safety perspective, increasing mileage increases crash risk (Tselentis et al., 2017) and, therefore, an input-oriented DEA program is being developed aiming to minimize inputs (recorded metrics) maintaining the same number of outputs (recorded distance). Although a trip cannot literally behave as a decision making unit, it can be evaluated as a DMU and, therefore, it will be considered as such for the purpose of this research considering trip attributes as inputs and outputs of the DEA program. This is deemed to be a correct assumption on a trip basis since a) all variables used are continuous quantitative variables as those used in previous DEA studies (Cook and Seiford (2009), Hollingsworth et al. (1999), Karlaftis et al., (2013), Egilmez and McAvoy, 2013) and b) a driver should reduce his mileage (Tselentis et al., 2017) and the frequency of some of his driving characteristics such as harsh acceleration and braking, mobile phone usage and speeding (Aarts and Van Schagen, 2006, Young et al, 2007, Hong et al., 2014, Tselentis et al., 2017). It is also implicitly assumed that the driving efficiency problem is a Constant Returns to Scale (CRS) problem and that the average and sum of all metrics (inputs) recorded, such as the number of harsh acceleration and braking events, changes proportionally to the sum of driving distance (output).

3.2. Theoretical background of DEA

DEA is a non-parametric approach that does not require any assumptions about the functional form of a production function and a priori information on importance of inputs and outputs. The relative efficiency of a DMU is measured by estimating the ratio of weighted outputs to weighted inputs and comparing it with other DMUs. DEA allows each DMU to choose the weights of inputs and outputs which maximize its efficiency. The DMUs that achieve efficiency equal to unit are considered efficient, while DMUs with efficiency scores between zero and unit are considered as inefficient. The first DEA model is the CCR model that assumes that production exhibits constant returns to scale i.e. outputs are increased proportionally to inputs (Charnes et al., 1978). DEA models can also be distinguished based on the objective of a model; that can be either outputs maximization (output-oriented model) or inputs minimization (input-oriented model).

In the present application of the modified DEA, the objective is to minimize the number of harsh acceleration and braking events etc. that take place for a specific driving distance, rendering the application as an input-oriented (IO) DEA model. This is much more realistic than to maximize driving distance for given metrics, since the latter would increase the exposure of a driver and, therefore, driving risk. It is also implicitly assumed that the driving efficiency problem is a Constant Returns to Scale (CRS) problem and that the average and sum of all metrics (inputs) recorded, such as the number of harsh accelerations and brakings in each $trip_i$, changes proportionally to the sum of driving distance (output).

Let X and Y to represent the set of inputs and outputs, respectively. Let the subscripts i and j to represent particular inputs and outputs respectively. Thus x_i represents the i^{th} input, and y_j represent the j^{th} output of a DMU. The input-oriented CCR model evaluates the efficiency of DMU_0 by solving *(1)*, which is the (envelopment form) linear program and its mathematical formulation as formulated in (Ramanathan, 2003), (Cooper et al., 2006):

$$\min \theta_B$$

Subject to the following constraints: (1)

$$\theta_B * x_0 - X * \lambda \geq 0$$
$$Y * \lambda \geq y_0$$
$$\lambda_i \geq 0 \forall \lambda_i \in \lambda$$

where λ_i is the weight coefficient for each DMU_i that is an element of set λ, X is the set of Inputs, Y is the set of outputs and θ_B is a scalar representing the efficiency of reference DMU_0.

The objective function of this linear programming problem (DEA) is min θ_i i.e. minimize the efficiency of DMU_i (in this case $trip_i$). In order to benchmark the efficiency of all trips (of each DMU) of the database, this linear programming problem should be solved for each DMU_i (i.e. each $trip_i$). This is radically increasing the processing

time of each problem as the number of DMUs and the dimensions of the problem (number of inputs and outputs) are increasing. For that reason, a great effort has been made in literature to reduce computation time of DEA in large-scale data.

3.3. Reduced Basis Entry (RBE) Algorithm

The Reduced Basis Entry (RBE) algorithm for DEA is iteratively solving the DEA LP for all DMUs in the database. The main difference from the standard DEA approach is that if the reference DMU_0 examined is found to be inefficient in an iteration, it is excluded from all the next solutions. Therefore, each time a non-efficient DMU is recognized, variables are reduced by one and as a result, the running time of the next LP will be lower. Thus, total computations are less expensive in terms of time. The pseudocode of RBE algorithm is given below:

```
For every DMU_x in DMUset:
    θ_x, λ_x = DEA(DMU_χ, input, output)
    if θ_x < 1 then:
        DMUset.remove(DMU_x)
        delete input_x
        delete output_x
```

where DEA is the function written for solving the DEA LPs given reference DMU name, input matrix and output matrix, θ_x is the estimated efficiency for DMU_x, λ_x is the weight coefficient of DMU_x and input and output are the matrices containing inputs and outputs respectively. This algorithm results in constructing two sets comprising of the thetas and lambdas of all DMUs in the dataset.

3.4. Convex Hull theoretical background

The convex hull of a set of points is the smallest convex set that contains all points in the set. Reducing the required computation time for finding the optimal solution of convex hull is a fundamental problem for mathematics and computational geometry. In quickhull algorithm (Barber et al., 1996), that is used herein, it is assumed that points are in a general position, so that their convex hull is a simplicial complex (Preparata et al., 1985). Its vertices and facets represent a d-dimensional convex hull. A point is deemed to be extreme, and, therefore, lies on the hull, if it is a vertex of the convex hull. Each facet comprises of a set of vertices, a set of neighboring facets, and a hyperplane equation. The ridges of the convex hull are the (d - 2) - dimensional faces. The point where the vertices of two neighboring facets intersect, constructs a ridge. Quickhull makes use of two geometric operations (Barber et al., 1996), oriented hyperplane through d points and signed distance to hyperplane. It represents a hyperplane by its outward-pointing unit normal and its offset from the origin. The inner product of the point and normal plus the offset represents the signed distance of a point to a hyperplane. A halfspace of points that have negative distances from the hyperplane is defined by the hyperplane. A point is above the hyperplane if this distance is positive.

Assuming a set E containing N DMUs convex hull algorithm will initially estimate the set E_e consisting of the number N_e of the most efficient DMUs. Consequently, each DMU m of the set $\{E\text{-}E_e\}$ of the non-efficient DMUs will be run with the set E_e creating $(N\text{-}N_e)$ DEA linear problems with N_e+1 (DMU that is evaluated) variables. This will allow for the calculation of efficiency θ and slacks λ of the peers for each DMU m of the set $\{E\text{-}E_e\}$ of the non-efficient DMUs.

It should be mentioned that the DEA - convex hull algorithm consists of 3 different steps namely convex hull solution, determination of the efficient DMUs, DEA solution for non-efficient DMUs. At the first step, convex hull points are identified creating thus a superset of N_c points that includes all efficient DMUs. Nonetheless, because some of the convex hull points are not efficient DMUs since they do not lie on the efficiency frontier, N_c DEA LPs are solved to find the N_e efficient DMUs. During the third step, $(N\text{-}N_e)$ DEA LPs one for each of the inefficient DMUs to estimate parameters θ_i and λ_i.

4. Data collection and processing

4.1. Tables

A naturalistic driving experiment is implemented in this research by recording personalized driving behaviour analytics in real time, exploiting data collected from smartphone device sensors using a smartphone application developed by OSeven Telematics. Eighty-eight (88) drivers participated in the designed experiment that took place between 28/09/2016 and 05/12/2016 and a large database of 10,088 trips is created. The data were anonymized so that the driving behaviour of each was not connected with any personal information.

OSeven has developed an integrated system for the recording, collection, storage, evaluation and visualization of driving behaviour data using smartphone applications and advanced Machine Learning algorithms. This innovative large-scale data collection and analysis methodology applied, presents new challenges by gathering large quantities of data for analysis during this research. The system developed integrates a data collection, transmission and processing procedure using Smartphones, the main features of which are outlined in the next paragraphs.

4.2. Data recording and transmission from smartphone

A naturalistic driving experiment is implemented in this research by recording driving behaviour analytics in real time, exploiting data collected from smartphone device sensors using a smartphone application developed by OSeven Telematics. Eighty-eight (88) drivers participated in the designed experiment that took place between 28/09/2016 and 05/12/2016 and a large database of 10,088 trips is created. The data were anonymized so that the driving behaviour of each participant was not connected with any personal information. OSeven has developed an integrated system for the recording, collection, storage, evaluation and visualization of driving behaviour data using smartphone applications and advanced Machine Learning algorithms. The basic operating frame of the data flow is shown in Figure 1.

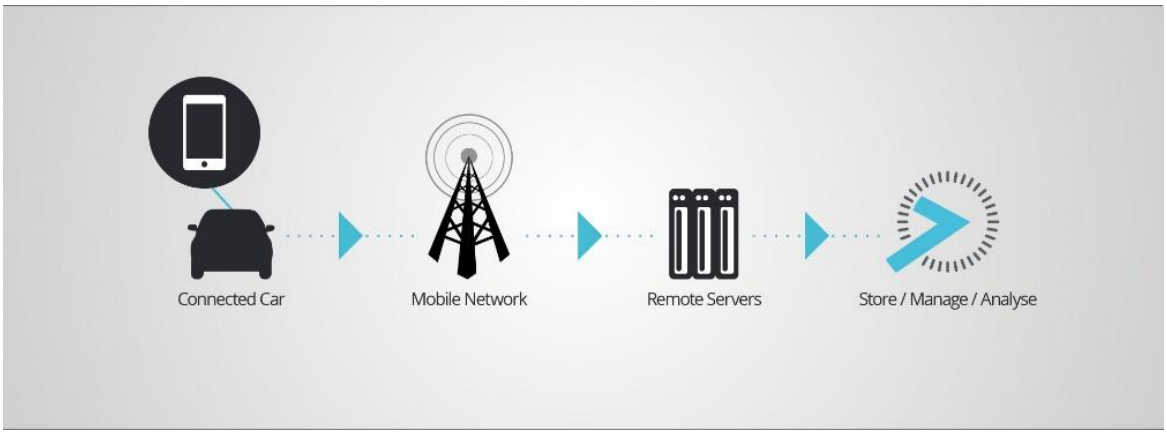

Fig. 1. OSeven data flow system.

Recorded data come from various smartphone sensors and data fusion algorithms provided by Android (Google) and iOS (Apple). Indicatively, technology sensors that are integrated in mobile phone are the Accelerometer*, the Gyroscope*, Magnetometer and the GPS (speed, course, longitude, latitude), while Fusion Data provided by iOS and Android are Yaw, pitch, roll, Linear acceleration*, and Gravity* (symbol * refers to sensors recording attributes in x, y, z axes). Data recording frequency varies depending on the type of the sensor with a minimum value of 1Hz. After the end of the trip, the application transmits all data to a central database via an appropriate communication channel such as a Wi-Fi or cellular network such as a 3G/4G network (online options) based on the user settings. After data are stored in the cloud server for central processing and data reduction, they are processed using big data mining techniques and machine learning (ML) algorithms.

Machine learning methods (filtering, clustering and classification methods) are used to clean the data from existing noise and errors, and identify repeating patterns within data. Artificial intelligence methods allow for the detection of aggressive behavior of the driver in the form of harsh events, mobile phone distraction, travel mode identification, speed limit exceedance as well as where the determination of the time and spatial characteristics of all the above. The procedure of the machine learning algorithms and big data mining techniques include fata filtering and outlier detection, data smoothening, speeding regions, harsh acceleration, braking and cornering events, mobile usage, risky hours driving. Aggregated Data are analyzed and filtered to retain only those indicators that will be used as inputs and outputs for the DEA problem. Data filtering and DEA improvement algorithms are performed in Python programming language and several scripts are written for this reason. Python packages used include pandas and numpy for numeric calculations and transformations, scipy that features quickhull algorithm and pulp for linear programming problem construction. More details on the algorithm implementation are given below. Coding is applied using Pycharm IDE Community edition, for Python & Scientific development. The computer used is an Intel® Core™ i7 CPU K 875 @ 2.93GHz × 8 featuring a 2.0 GiB Ram memory running on ubuntu 16.04 LTS.

5. Implementation and results

Models representing driving behaviour in all road types are constructed, with multiple inputs and outputs. Input and output selection is a critical procedure for DEA and should be linked to the conceptual specifications of each problem. (Dyson et al., 2001) discussed several issues that should be taken into consideration before applying DEA to a dataset. One of the pitfalls is that the efficiency score might be miscalculated when input and output variables are in the form of percentiles and/or ratios simultaneously with raw data (Cooper et al., 2006).

The specific data used in this study are metrics recorded in the form of raw data i.e. the number of harsh braking (HB)/ accelerations (HA)/ cornering (HC) events, seconds driving over the speed limit (SP) and seconds used the mobile phone (MU). All metrics are recorded per road type (urban, rural, highway) e.g. haurban (number of harsh accelerations that occurred in urban road), harural, hahighway, hburban (number of harsh brakings that occurred in urban road), hbrural, hbhighway etc.

In this specific experiment, distance per different road type travelled are considered as DEA outputs and again, convex hull algorithm is applied before applying standard DEA. Convex hull's dimension is determined by the sum of inputs and outputs of the DEA problem.

5.1. Modified Multiple Input-Output DEA: a comparative study

Overall, the three approaches (Standard, RBE and CH DEA) are tested for seven different scenarios, i.e. for 100, 500, 1000, 2500, 5000, 7500 and 10088 DMUs, the results of which are presented in the following section.

The amount of computational memory required to perform the Convex Hull – DEA (CH DEA) approach is notably high. Quickhull algorithm applied herein does not support medium-sized inputs in 9-D and higher, which is the limitation of the present study. This is the reason why the authors choose to test their models only for six inputs and three outputs in order to create a convex hull problem of 9-D which can be calculated as described in the previous section. Three outputs and six inputs are examined, instead of two or four for instance, for the results to be easily explained from a transportation engineering perspective. The combinations of number of harsh acceleration, braking and cornering events, seconds driving over the speed limit and seconds used the mobile phone per road type with distance per road type were used to create 5 different DEA problems but herein only harsh acceleration per road type with distance per road type is chosen to be presented to avoid chattering. All models provided similar results and therefore conclusions drawn can be generalized regardless of the variables chosen in the model. The specifications of the models implemented are shown in Table 1.

In every scenario tested, results showed that CH DEA method yield the exactly the same results as the other two approaches tested in terms of identifying the most efficient DMUs, calculating the lamdas and theta values, determining the peers and calculating the efficient level of inputs and outputs for each DMU. This is a weighty outcome because for the first time tests proved the efficacy of the proposed methodology for performing a multiple input and output CH DEA.

Table 1. Inputs and outputs of DEA models used.

DEA Models	Set of Inputs used	Set of Outputs used
Model 1	1) HA in urban road	1) Distance in urban road
	2) HA in rural road	2) Distance in rural road
	3) HA in highway	3) Distance in highway
Model 2	1) HB in urban road	1) Distance in urban road
	2) HB in rural road	2) Distance in rural road
	3) HB in highway	3) Distance in highway
Model 3	1) HC in urban road	1) Distance in urban road
	2) HC in rural road	2) Distance in rural road
	3) HC in highway	3) Distance in highway
Model 4	1) SP in urban road	1) Distance in urban road
	2) SP in rural road	2) Distance in rural road
	3) SP in highway	3) Distance in highway
Model 5	1) MU in urban road	1) Distance in urban road
	2) MU in rural road	2) Distance in rural road
	3) MU in highway	3) Distance in highway

Table 2 shows lambdas and theta for the first ten DMUs, where L_X stands for the lamda coefficient of the efficient DMU X that acts as a peer for the DMU examined. For instance, for the first row of the table where DEA is solved for DMU_1 (i.e. $trip_1$), L_{745}, the value of the lamda coefficient of DMU_{745} ($trip_{745}$), is equal to 0.015. The efficient level of inputs for $trip_1$ can be calculated as the product sum of the lamdas and the input values of each of the identified peers whereas to find the efficient level of outputs for the same DMU, each output value should be divided by theta value. Again, taking trip 1 as example, the efficient level of ha_{urban} can be estimated using (2):

$$\text{Efficient level of } ha_{urban_1} = L_{745} * ha_{urban_{745}} + L_{4403} * ha_{urban_{4403}} + L_{5293} * ha_{urban_{5293}} \Rightarrow$$
$$\Rightarrow \text{Efficient level of } ha_{urban_1} = 0.015 * ha_{urban_{745}} + 0.008 * ha_{urban_{4403}} + 2.223 * ha_{urban_{5293}} \quad (2)$$

On the other hand, the efficient level of e.g. $distance_{urban}$ is calculated in (3):

$$\text{Efficient level of } distance_{urban_1} = distance_{urban_1} / theta_1 \quad (3)$$

Table 2. Lamdas and thetas of the first ten DMUs.

	Optimal Lamdas				Theta
	L745	L4403	L5293	L9493	
Trip1	0.015	0.008	2.223	-	0.008
Trip2	0.085	0.097	1.335	-	0.022
Trip3	-	0.009	0.536	0.0002	0.003
Trip4	0.009	0.032	2.121	-	0.008
Trip5	-	0.011	1.841	0.025	0.021
Trip6	-	0.032	2.722	0.016	0.016
Trip7	0.054	0.039	1.716	-	0.013
Trip8	0.025	0.028	1.355	-	0.007
Trip9	0.041	0.024	2.955	-	0.012
Trip10	0.003	0.025	1.761	-	0.007
*Inputs = ['ha_{urban}', 'ha_{rural}', '$ha_{highway}$'], Outputs = ['$distance_{urban}$', '$distance_{rural}$', '$distance_{highway}$']					

It should be noted that it is not feasible to illustrate the table of the whole database as it is a matrix of 50,000 rows, one for each DMU, and as a result, only the first ten are shown indicatively.

5.2. Comparative study based on computational time

Results illustrated in table 3, indicate a superiority of the proposed method over the standard and RBE DEA approaches in terms of computation time. As anticipated, CH DEA approach significantly outperformed the other two especially for samples of a smaller scale. Results are not presented only as absolute values but as percentages of improvement as well in order for the results to be representative regardless of a computer's performance. As anticipated, computation time appears to be linearly increased in CH DEA method as the time required for each LP to be solved depends only on the number of the efficient DMUs found in the first step of the process. The number of used DMU in the LP in each iteration is kept constant (plus the reference DMU in each iteration) and as a result, the total time is proportionally increased to the total number of DMUs. In the specific DEA problem presented in Table 3, the density of the efficiency DMUs is found to be very low which reduces the computation time considerably since each of the 10073 LPs (10088 in total minus 15 efficient) that needs to be solved has only 16 (15 efficient plus 1 reference DMU in each LP) DMUs.

RBE was also confirmed to perform faster than standard approach especially for larger datasets. Nonetheless, the percentage of running time improvement over the standard DEA approach is kept constant aside from the sample size. On the other hand, RBE is found to be significantly slower than CH DEA; ranging between 33.33% and 99.97% from 100 to 10088 DMUs respectively. It is evident that for small scale samples of less than 500 DMUs the computational time gain is not worthwhile and probably standard approach should be preferred. Finally, standard approach and RBE is proved to be a non-feasible option for analyzing large-scale data using DEA which need several days (more than 40 and 12 days respectively) of processing on a conventional computer. This implies that alternative solutions such as the one examined in this paper should be further investigated and appraised especially when it comes for analyzing Big Data with DEA. The efficacy of the CH DEA algorithm investigated here, in terms of running time, provides encouraging insights for future enhancements on DEA addressing the issue of reducing its computation time.

Table 3. Computation time for seven scenarios.

	Computation time (sec)			CH DEA % computation time improvement over	
DMU No	Standard DEA Approach	RBE DEA	CH DEA	Standard DEA Approach	RBE DEA
100	11	6	4	63.64%	33.33%
500	477	169	21	95.60%	87.57%
1000	3250	1121	41	98.74%	96.34%
2500	44435	15570	94	99.79%	99.40%
5000	398485	123986	180	99.95%	99.85%
7500	1400909	444498	231	99.98%	99.95%
10088	3519372	1089731	314	99.99%	99.97%
*Inputs = ['ha_{urban}', 'ha_{rural}', '$ha_{highway}$'], Outputs = '$distance_{urban}$', '$distance_{rural}$', '$distance_{highway}$']					

Running time results are also illustrated in fig. 2; convex hull results are plotted in the secondary axis because computation time showed that convex hull significantly outperforms the other two approaches tested and therefore demonstration would not be distinguishable.

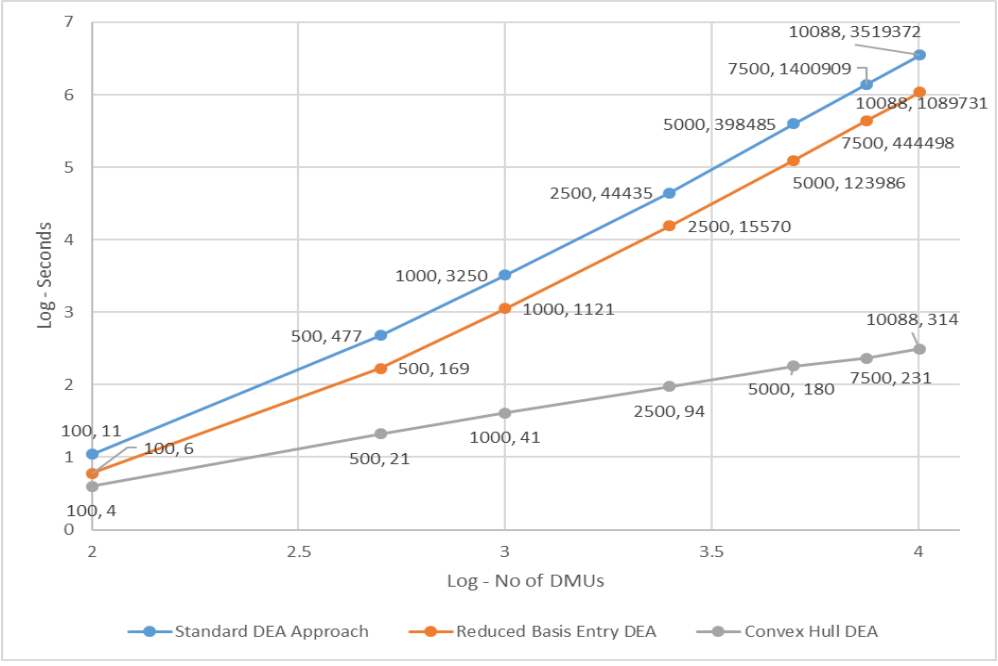

Fig. 2. Computation time of the three methodologies implemented.

6. Conclusions

This study encounters the problem of improving the existing optimization techniques for Data Envelopment Analysis (DEA). A methodology for reducing the computation time of a large-scale DEA problem that incorporates multiple inputs and outputs is proposed herein, which is a combination of the convex hull problem (quickhull

algorithm) and DEA. To the best of the authors' knowledge no effort has been made in the past to combine the computational geometry procedure of convex hull with DEA for reducing the running time of a large-scale DEA problem that features multiple inputs and outputs. To test the efficiency of the proposed approach, a driving data sample of 10,088 trips collected from smartphone devices was exploited for the purpose of this study with the aim to identify which trips are efficient and which are not based on the drivers behavior. The scenarios that were taken into consideration for testing included sets of 100, 500, 1000, 5,000 and 10,088 trips (DMUs) and compared based on running time of each of the algorithms applied.

Results indicate that the proposed CH DEA algorithm is performing significantly better for large-scale data compared to other existing DEA algorithmic methodologies such as Standard DEA and RBE DEA methodologies. The efficacy of the CH DEA algorithm investigated here, in terms of running time, provides encouraging insights for future enhancements on DEA addressing the issue of reducing its computation time. A major improvement is not noticed for data of a smaller scale implying that traditional DEA approach might be preferable. Further research is needed towards improving the algorithms to overcome the dimensionality limitation, so as the algorithm can incorporate input and output matrices of higher dimensions.

Another important finding of this research is that it suggests an approach to assess the driving efficiency of a trip. The methodology to estimate the efficiency index of a trip and identify the "peers" of a trip and, therefore, its efficient level of inputs (harsh accelerations/ harsh brakings etc. per road type) and outputs (distance per road type) was shown. The results of per trip analysis can be further exploited as an innovative approach to measure the per trip efficiency of a database that includes a vast number of trips. This presents a considerable opportunity for road safety, as, in this framework, both trips and drivers could be classified into different efficiency categories (such as efficient, less efficient, non-efficient) and further evaluate their main characteristics in terms of traffic risk, performance, aggressiveness, eco-driving etc.

Acknowledgements

This research is co-financed by the European Union - European Regional Development Fund (ERDF) and Greek national funds through the Operational Program "Competitiveness, Entrepreneurship and Innovation" (EPAnEK) of the National Strategic Reference Framework (NSRF) - Research Funding Program: BeSmart - Multi-modal driver behavior and safety support system on the basis of smartphone applications.

The authors would like to thank OSeven Telematics, London, UK for providing all necessary data exploited to accomplish this study.

References

Cook, W. D., & Seiford, L. M. Data envelopment analysis (DEA)–Thirty years on. European journal of operational research, Vol. 192, No. 1, 2009, pp. 1-17. https://doi.org/10.1016/j.ejor.2008.01.032

Emrouznejad, A., Parker, B. R., & Tavares, G. Evaluation of research in efficiency and productivity: A survey and analysis of the first 30 years of scholarly literature in DEA. Socio-economic planning sciences, Vol. 42, No 3, 2008, pp. 151-157. https://doi.org/10.1016/j.seps.2007.07.002

Hollingsworth, B., Dawson, P. J., & Maniadakis, N. Efficiency measurement of health care: a review of non-parametric methods and applications. Health care management science, Vol 2, No. 3, 1999, pp. 161-172. https://doi.org/10.1023/A:1019087828488

Karlaftis, M.G., Gleason, J.M., Barnum, D.T. 'Bibliography of Urban Transit Data Envelopment Analysis (DEA) Publications.' Available at SSRN: http://ssrn.com/abstract=1350583, http://dx.doi.org/10.2139/ssrn.1350583, 2013

Shen, Y., Hermans, E., Ruan, D., Wets, G., Brijs, T., & Vanhoof, K. A generalized multiple layer data envelopment analysis model for hierarchical structure assessment: A case study in road safety performance evaluation. Expert systems with applications, Vol 38, No. 12, 2011, pp. 15262-15272. https://doi.org/10.1016/j.eswa.2011.05.073

Egilmez, G., & McAvoy, D. Benchmarking road safety of US states: A DEA-based Malmquist productivity index approach. Accident Analysis & Prevention, Vol. 53, 2013, pp. 55-64. https://doi.org/10.1016/j.aap.2012.12.038

Alper, D., Sinuany-Stern, Z., & Shinar, D. Evaluating the efficiency of local municipalities in providing traffic safety using the Data Envelopment Analysis. Accident Analysis & Prevention, Vol. 78, 2015, pp. 39-50. https://doi.org/10.1016/j.aap.2015.02.014

Vlahogianni, Eleni I. Computational intelligence and optimization for transportation big data: challenges and opportunities. Engineering and Applied Sciences Optimization. Springer International Publishing, 2015, pp. 107-128. https://doi.org/10.1007/978-3-319-18320-6_7

Dulá, J. H. A computational study of DEA with massive data sets. Computers & Operations Research, Vol. 35, No. 4, 2008, pp. 1191-1203. https://doi.org/10.1016/j.cor.2006.07.011

Barr, R.S. and M.L. Durchholz. Parallel and hierarchical decomposition approaches for solving large-scale Data Envelopment Analysis models. Annals of Operations Research, Vol. 73, 1997, pp. 339-372. https://doi.org/10.1023/A:1018941531019

Ali, A. I. Streamlined computation for data envelopment analysis. European journal of operational research, Vol. 64, No. 1, 1993, pp. 61-67. https://doi.org/10.1016/0377-2217(93)90008-B

Dulá, J. H., & López, F. J. Preprocessing DEA. Computers & Operations Research, Vol. 36, No. 4, 2009, pp. 1204-1220. https://doi.org/10.1016/j.cor.2008.01.004

Aarts, L., & Van Schagen, I. Driving speed and the risk of road crashes: A review. Accident Analysis & Prevention, Vol. 38, No. 2, 2006, pp. 215-224. https://doi.org/10.1016/j.aap.2005.07.004

Young, K., Regan, M., & Hammer, M. Driver distraction: A review of the literature. Distracted driving, 2007, pp. 379-405.

Hong, J. H., Margines, B., & Dey, A. K. A smartphone-based sensing platform to model aggressive driving behaviors. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems, 2014 April, pp. 4047-4056. https://doi.org/10.1145/2556288.2557321

Tselentis, D. I., Yannis, G., & Vlahogianni, E. I. Innovative motor insurance schemes: a review of current practices and emerging challenges. Accident Analysis & Prevention, Vol 98, 2017, pp. 139-148. https://doi.org/10.1016/j.aap.2016.10.006

Dulá, J.H. and B. L. Hickman. Effects of excluding the column being scored from the DEA envelopment LP technology matrix. Journal of the Operational Research Society, Vol. 48, 1997, pp. 1001– 1012. https://doi.org/10.1057/palgrave.jors.2600434

Dulá, J. H., & López, F. J. Algorithms for the frame of a finitely generated unbounded polyhedron. INFORMS Journal on Computing, Vol. 18, No. 1, 2006, pp. 97-110. https://doi.org/%7B0%7D

Charnes, A., Cooper, W. W., & Rhodes, E. Measuring the efficiency of decision making units. European journal of operational research, Vol. 2, No. 6, 1978, pp. 429-444. https://doi.org/10.1016/0377-2217(78)90138-8

Ramanathan, Ramu, ed. An introduction to data envelopment analysis: a tool for performance measurement. Sage, 2003.

Cooper, W. W., Seiford, L. M., & Tone, K. Introduction to data envelopment analysis and its uses: with DEA-solver software and references. Springer Science & Business Media. 2006.

Barber, C. B., Dobkin, D. P., & Huhdanpaa, H. The quickhull algorithm for convex hulls. ACM Transactions on Mathematical Software (TOMS), Vol. 22, No. 4, 1996, pp. 469-483. https://doi.org/10.1145/235815.235821

Preparata, F. P., & Shamos, M. I. Introduction. In Computational Geometry, Springer New York, 1985, pp. 1-35, https://doi.org/10.1007/978-1-4612-1098-6_1

Dyson RG, Allen R, Camanho AS, Podinovski VV, Sarrico CS, Shale EA. Pitfalls and protocols in DEA. European Journal of Operational Research, Vol. 132, Issue 2, 2001, pp. 245–259. https://doi.org/10.1016/S0377-2217(00)00149-1

Cooper WW, Seiford LM, Tone K. Data envelopment analysis: a comprehensive Text with models, applications, references and DEA-Solver Software, Springer Science & Business Media, 2006.