Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 3063–3070
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.26054.3208

Quantum algorithms for solving data envelopment analysis models

Najmeh Malekmohammadi[a,*], Ali Ebrahimnejad[b], Samira Sayedsalehi[c]

[a]Department of Mathematics, South Tehran Branch, Islamic Azad University, Tehran, Iran
[b]Department of Mathematics, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran
[c]Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

(Communicated by Saman Babaie-Kafaki)

Abstract

In traditional optimization models, classical computers have been used to solve mathematical linear or non-linear models. In addition, by data envelopment analysis (DEA) models one can evaluate the relative efficiency of the decision-making units originally in the form of mathematical non-linear models. In this paper, a new attitude is presented in which quantum computers and quantum algorithms can be used to solve DEA models by two different methods. This suggested attitude is worthy to mention as this subject is new and will attract researchers to discuss and improve it.

Keywords: Data Envelopment Analysis, Non-linear Models, Quantum Algorithm, Quantum Optimization
2020 MSC: 90XX, 68Q12

1 Introduction

As we know that ordinary computers work on 0 and 1, in other words, the basic unit of information is the bit which can assume one of two states that labeled 0 and 1. For performing a task (whether it is a calculation or reserving a ticket) the instance of the task is translated into a string of 0 and 1 (the input), which is then done by an algorithm. In quantum computing, the basic unit of information is called quantum bit. A Quantum computer can work something like both 0 and 1 simultaneously and can be more powerful as it is able to solve FACTORING (an ancient hard problem) in polynomial time. Quantum parallelism is a situation as quantum computers operate on superpositions of all classical search states and allow them to evaluate properties of all states in about the same time as the classical machine needs for a single evaluation. All states in quantum computer are described by state vectors including complex numbers named amplitudes. As regards, the state will collapse to one of the basis states if by measuring the state to the computational basis, these numbers are tried to read. This tremendous storage and parallel processing capacity can be utilized in generic problems which include some global properties of functions whose values have been maintained in several multi-qubit quantum state. For specific problems, quantum algorithms provide exponential speedups over their classical peers, the most famous example is a Shor's factoring algorithm [5].

*Corresponding author
Email addresses: N_malekmohammadi@azad.ac.ir (Najmeh Malekmohammadi), a.ebrahimnejad@qaemiau.ac.ir (Ali Ebrahimnejad), S_sayedsalehi@azad.ac.ir (Samira Sayedsalehi)

Received: January 2022 *Accepted:* May 2022

In quantum algorithms, there are two quantum phenomena Entanglement and Quantum Interference. Entanglement enables the coding of data in a nontrivial superposition of states which are not able to be itemized into individual qubit states. Quantum Interference is the key component inducing initial states (input) to final states (output) discriminating intermediate multi-qubit superpositions in some assigned way. Decision problems are localized by quantum search algorithms which have an expeditiously computable test to determine a given state is a solution or not. Sans applying any information about the problems over this test, by utilizing amplitude amplification quantum computers improve quadratically in search speed. If quantum algorithm doesn't exist to solve practical problems, quantum computer hardware will be useless like a computer which has no operating system. More comprehensive information can be reached in [7] and [8].

On the other side, as a mathematical and computer's viewpoint, an optimization problem is the problem of searching the best solution from all feasible solutions. Optimization problems (for example, nonlinear programming) can be solved by quantum computing. Side and Erol [9] showed the improvement in computation with a quantum algorithm as they considered a real-world optimization problem. Hogg and Portnov [6] mentioned the quantum optimization algorithm (which is presented in their paper) operates on superpositions of all search states and attempts to investigate a state with a fairly low cost. The cost related to each search state is applied to adopt the phase of the state's amplitude and amplitudes from various states are combined by a mixing operation. Particularly, the overall algorithm includes a number of independent trials, so that each of them returns a single state after the ultimate measurement. The number of trials can be stabled beforehand if some (expectantly low-cost) state is needed within a preset time-bound or can resume until some other indicator is content, e.g., an enough low-cost state is gained or there is no further improvement in a long series of trials. In this regard, this algorithm acts like incomplete classical heuristics which want to offer low-cost states but can find the absolute minimum with no guarantee. In addition, when the minimum is gained, there is no guarantee for the minimum cost and no further trials for giving a lower cost. In summary, they have applied the cost related to states in an optimization problem to adopt a superposition for increasing the amplitudes linked to low-cost states.

As the constraints in linear programming are mostly like linear equations, it is worthy to mention that Harrow et al. [5] proposed a quantum algorithm to approximate attributes of the solution of a set of linear equations. Compared to classical algorithms for the same task, their algorithm can be much exponentially faster. Sun et al. [11] concentrated on solving nonlinear programming (NLP) problems based on quantum-behaved particle swarm optimization (QPSO). After a brief introduction to the original particle swarm optimization (PSO), they explained the origin and the improvement of QPSO and the penalty function method for constrained NLP problems. The efficiency of QPSO was tested on some benchmark functions (unconstrained and constrained) and made comparisons with PSO with both inertia weight and constriction factor. Finally, by the numerical results, the advantages of their method over the previous one were shown.

In addition, data envelopment analysis (DEA) regarded as a fair method for evaluating the relative efficiency of decision making units (DMUs) which is firstly proposed by Charnes et al. [2] in 1978. The conventional DEA models have been introduced to evaluate the relative efficiency of DMUs with exact data. As the data in real world problems are imprecise, since then many models introduced in DEA with fuzzy, stochastic and ordinal data and have been solved by classic optimization algorithms. For instance, Manuel et al. [13] rectified the problem of the relative efficiency when the inputs/outputs are fuzzy data. Hosseinzadeh et al. [14] introduced advanced DEA models with R coded and improved it to Fuzzy DEA. Banker [15] evaluated the relative efficiency by considering the possibility of random factors. The proposed models generally include both a minimum absolute deviation regression model and DEA model.

In this paper, a new approach is proposed in which the quantum algorithms are applied to DEA models by two different methods. In the first method, the efficiency score related to each search state is utilized to adjust the phase of the state amplitude, and a mixing operation combines amplitudes from different states. In the second one, the proposed algorithm is suitable when we have large condition number in the matrix which includes inputs and outputs in DEA models. The main contributions of this paper can be summarized as follows:

- The connection between DEA models and quantum algorithms.
- Presenting two quantum algorithms for solving optimization models.
- Proposing a suitable method when the matrix (includes inputs and outputs in DEA models) is presented with large condition number.
- Introducing quantum algorithms for solving DEA models is worthy to mention as this subject is new and will attract researchers to discuss and improve it.

The rest of this paper is organized as follows. In section 2, data envelopment analysis is introduced. In section 3, the two new approaches are presented to show how quantum algorithms can solve DEA models. Finally, section 4 concludes the paper.

2 Data Envelopment Analysis

In this section, the CCR envelopment model (the first DEA model) proposed by Charnes et al. [2] is introduced. Suppose we have a set of $j = 1, ..., n$ decision making units (DMUs) and each unit uses input $X \in R_+^m$ quantities to produce output quantities $Y \in R_+^s$. Now, consider the index sets of inputs, $I = 1, ..., m$ and outputs, $O = 1, ...s$. Also $o(o \in \{1, ..., n\})$ is the DMU under assessment (usually denoted by DMU_o). To evaluate the relative efficiency of DMU_o, it is needed to solve the following linear mathematical programming problem:

$$
\begin{aligned}
Min \quad & \theta_o \\
s.t. \quad & \sum_{j=1}^{n} \lambda_j x_{ij} \leq \theta_o x_{io}, \qquad i \in I, \\
& \sum_{j=1}^{n} \lambda_j y_{kj} \geq y_{ko}, \qquad k \in O, \\
& \lambda_j \geq 0, \forall j, \theta_o \quad free.
\end{aligned} \tag{2.1}
$$

The major aim in primary DEA models was the evaluation of the efficiency in DMUs. Since, one of the things which is important for the decision makers (DMs) to know is the projection of the DMUs onto the efficient frontier (acceptable and ideal for them or not), lots of papers are begun under the names of target setting. Since then, DEA has been used for future programming of organizations and the responses of different policies.

In Figure 1, the process of target setting via DEA is shown, geometrically. In this figure DMUs A, B, C, D and E are relatively efficient (under variable returns to scale). DMU F is located in the interior of the production possibility set and thus is relatively inefficient. Using ordinary data envelopment analysis, the efficiency and performance of DMU F can be increased by projecting its input/output mix along the segment KCG on the efficient frontier. The increase of output will project F at point G whilst the decrease of input at point K. So, the two strategies constitute two alternative scenarios for improving the performance of DMU F. Further scenarios can also be explored by decreasing input and increasing output simultaneously which project unit F at point C of the efficient frontier KCG.

The development of projection scenarios on the efficient frontier may have a negative effect on performance. Consider the case of the input decrease of DMU F which projects it to point K. The mathematical feasibility of such a projection, however, does not guarantee managerial feasibility, too. Reducing the input levels of unit F could perhaps have a negative effect on its performance by forcing it to a point such as $K^{'}$ in the near future. This implies that the intentional/ unintentional consequences of target setting should be explored as a part of the target setting process (Athanassopoulos et al. [12]).

DMU_o is CCR efficient if $\theta^* = 1$ and the slacks s_k^{+*}, s_i^{-*}in every optimal solution are zero. The target input and output for DMU_o can be obtained as follows:

$$
\begin{aligned}
\bar{x}_{io} &= \theta_o^* x_{io} - s_i^{*-}, \forall i \in I \\
\bar{y}_{ko} &= y_{ko} + s_k^{*+}, \forall k \in O.
\end{aligned}
$$

As mentioned before, all models in DEA are solved by classic optimization algorithms which are mainly presented in [4]. This paper demonstrates that it is possible to solve DEA models by quantum algorithms.

3 Quantum Algorithms in DEA

In this section, it is shown that DEA models can be solved by two quantum algorithms which are inspired from [5] and [6]. The first algorithm operates on superpositions of all search states (Figure 2) and tries to discover a state with relatively low efficiency. The efficiency score related to each search state is utilized to adjust the phase of the state's amplitude, and a mixing operation combines amplitudes from different states. As a matter of fact, the overall

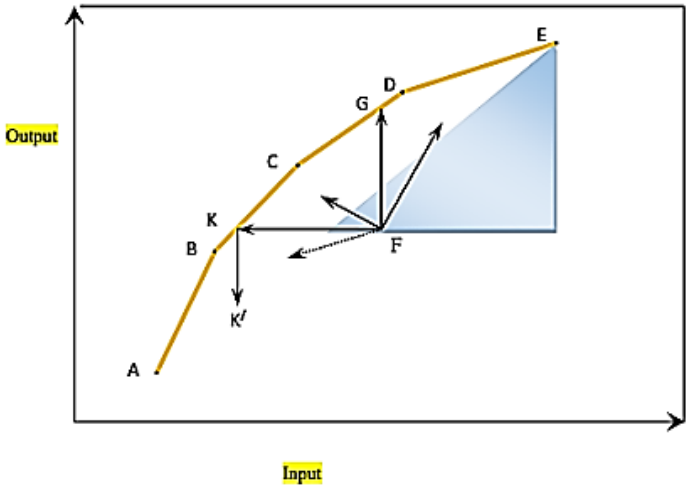

Figure 1: Target Setting Scenarios by Means of Frontier Analysis

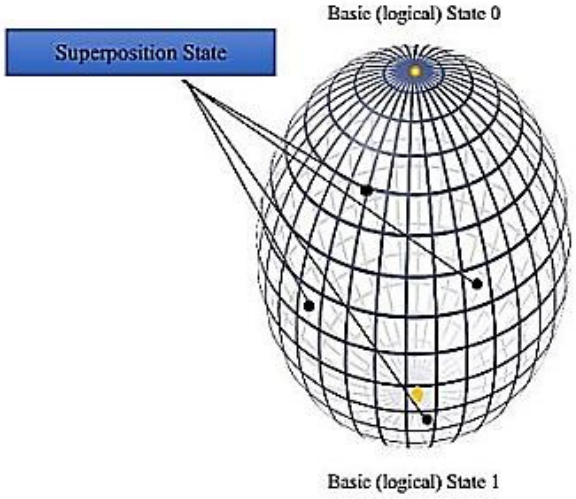

Figure 2: Superpositions of all states

algorithm consists of a number of independent trials, each of which returns a single state after the final measurement and can continue till low-efficiency state is found or a long series of trials gives no further improvement.

To find the minimum efficiency θ, such optimization searches can be treated as a series of decision problems with different assumed values for minimum efficiency. To introduce the algorithm, suppose a quantum computer with n bits, first of all let $\mu_s^{(0)} = 2^{(\frac{-n}{2})}$ for each of 2^n states s and next for steps 1 through j, adopt amplitude phases based on the efficiency related to the states and then mix them. These operations accord with matrix multiplication of the state vector, with the final state vector presented by:

$$\mu^{(j)} = D^{(j)}F^{(j)}, ..., D^{(1)}F^{(1)}\mu^{(0)} \tag{3.1}$$

where, for step h; $D^{(h)}$ is the mixing matrix and $F^{(h)}$ is the phase matrix. Finally in the last step, measure the final superposition, giving state s with probability $p(s) = |\mu_s^{(j)}|^2$.

Therefore, the probability to reach a minimum-efficiency state with a single trial is $F_{min} = \sum_s p(s)$ where the sum is over those s with the minimum efficiency. The mixing matrix is $D^{(h)} = WT^{(h)}W$, where for states r and s, $W_{rs} = 2^{(\frac{-n}{2})}(-1)^{|r \wedge s|}$ is the Walsh transform and $|r \wedge s|$ is the number of 1-bits the states have often. The matrix

$T^{(h)}$ is diagonal with elements relying on $|s|$, the number of 1-bits state s includes $T^{(h)}_{ss} = t^{(h)}_{|s|}$ with

$$t^{(h)}_b = e^{i\pi\tau_h b} \tag{3.2}$$

where τ_h is a constant which depends on the class of problems and the number of steps, but not the particular problem instance being computed. From these definitions, the elements $D^{(h)}_{rs}$ associate only with the Hamming distance between the states, $d(r,s)$, i.e., the number of bits with different values in the two states. Therefore, it can be written $D^{(h)}_{rs} = u^{(h)}_{d(r,s)}$ with $u^{(h)}_d = (-i\ \tan(\frac{\pi\tau_h}{2}))^d$, up to an overall phase and normalization constant. The phase adjustment matrix, $F^{(h)}$, is a unitary diagonal matrix rely on the problem instance which is computed, with values determined by the efficiency score related to define each state: $F^{(h)}_{rr} = p^{(h)}_{\theta(r)}$ and

$$p^{(h)}_\theta = e^{i\pi\rho_h\theta} \tag{3.3}$$

where ρ_h is a constant and $\theta(r)$ is the efficiency related to search state r. This algorithm has the same overall structure as amplitude amplification [6]. The application of the above algorithm can be considered in two approaches; the first one uses a sample from the class of problems to be solved, and numerically adjusts the parameters to give the largest probability of finding a minimum efficiency state when averaged over the sample. In the second approach, the asymptotic average behavior of the algorithm is evaluated, as a function of the phase parameters, and values presenting a pleasant average performance for large problems are chosen as follows:

Suppose there are j steps and the phase parameters as, τ_h and ρ_h for $h = 1, ..., j$. The average amplitude of states with efficiency θ after step h is $\phi^{(h)}_\theta$. The change in average amplitudes from one step to the next is nearly,

$$\phi^{(h)}_{\acute{\theta}} = \sum_{d\theta} u^{(h)}_d p^{(h)}_\theta \phi^{(h-1)}_\theta v(\acute{\theta}, d, \theta) \tag{3.4}$$

where $v(\acute{\theta}, d, \theta)$ is the average number of states with efficiency θ at distance d from a state with efficiency $\acute{\theta}$. This quantity can be represented as $\binom{c}{d} P(\theta|d, \acute{\theta})$, where $P(\theta|d, \acute{\theta})$ is the conditional probability of a state has efficiency θ when at distance d from a state with efficiency $\acute{\theta}$.

Satisfiability is a combinatorial search problem including a propositional formula in n Boolean variables and the requirement to discover a task (true or false) to each variable in which the formula is true. The Satisfiability and the behavior of it have been explained completely in [6].

The second method shows the application of the quantum algorithms in DEA models is based on [5]. Consider the dual form of Model (2.1) as follows:

$$\begin{aligned}
Max \quad & \sum_{k=1}^{s} u_k y_{ko} \\
s.t. \quad & \sum_{i=1}^{m} v_i x_{io} = 1, \\
& \sum_{k=1}^{s} u_k y_{kj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0, \qquad j = 1, ..., m, \\
& u_k \geq 0, k \in O, v_i \geq 0, i \in I.
\end{aligned} \tag{3.5}$$

where $u_k, k \in O$ and $v_i, i \in I$ are dual variables. According to the duality theorems in [1], the optimal values of the primal problem and its dual are identical. Duality theorems imply that solving Model (2.1) and (3.5) are equivalent

to finding a non-negative solution of the following system of equations:

$$\begin{aligned}
\theta_o - \sum_{k=1}^{s} u_k y_{ko} &= 0 \\
\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- &= \theta_o x_{io}, \qquad i \in I, \\
\sum_{j=1}^{n} \lambda_j y_{kj} - s_k^+ &= y_{ko}, \qquad k \in O, \\
\sum_{i=1}^{m} v_i x_{io} &= 1, \\
\sum_{k=1}^{s} u_k y_{kj} - \sum_{i=1}^{m} v_i x_{ij} + f_j &= 0, \qquad j = 1, ..., m.
\end{aligned}$$

This system can be considered as a traditional system of linear equations $Ax = b$ where x is the solution of both Models (2.1) and (3.5). Gaining the non-negative solution of the above system concludes in obtaining $x \geq 0$ which minimizes $\|Ax - b\|$. This is Non-negative Least Squares (NNLS) solution of $Ax = b$.

Now, given matrix A with large condition number (matrix A includes inputs and outputs in DEA models) the proposed algorithm in [5] can be applied. Given a Hermitian $N \times N$ matrix A, suppose λ_l and $u_l, l = 1, ..., N$ are the eigenvalues and the eigenvector basis of A respectively. The ratio between A's largest and smallest eigenvalues is κ. Given A with large condition number, the algorithm can select to invert only the part of $|b>$ which is in the well-conditioned part of A (Figure 3 shows the both well-conditioned and the ill-conditioned matrix).

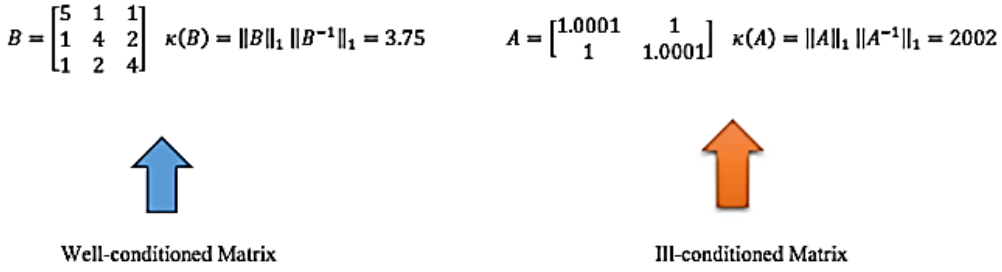

Figure 3: The example of given matrix A with large condition number

Instead of converting $|b>= \sum_l \beta_l |u_l>$ to $|x>= \sum_l \lambda_l^{-1} \beta_l |u_l>$, one can turn it to a state which is close to

$$\sum_{l,\lambda_l<1/\kappa} \lambda_l^{-1} \beta_l |u_l> |well> + \sum_{l,\lambda_l \geq 1/\kappa} \beta_l |u_l> |ill> \tag{3.6}$$

in time proportional to κ^2 for any chosen κ (i.e., not necessarily the true condition number of A). In the last qubit, the size of the ill-conditioned part can be approximated by the users. The pseudoinverse of A can be computed by this procedure, as $1/\kappa$ is taken to be smaller than the smallest nonzero eigenvalue of A and A is not invertible. It is worthy to mention that by applying preconditioner in classical algorithms, a different method is presented to handle ill-conditioned matrices [3, 10]. If a method of procreating a preconditioner matrix B is available such that $\kappa(BA)$ is smaller than $\kappa(A)$, then $A\vec{x} = \vec{b}$ is able to be computed instead of gaining the easier matrix inversion problem $(BA)\vec{x} = B\vec{b}$. In conclusion, if we use suitable preconditioner, the proposed algorithm will be able to run much faster in the case of the preparation of the state proportional to $B|b>$. In addition, the algorithm can be utilized to generate a quantum analogue of Monte Carlo calculations, where given A and $\vec{b}$ are sampled from the vector $\vec{x}$, meaning that the value i occurs with probability $|\vec{x_i}|^2$.

The two proposed methods are introduced in this section to help the DMs to gain the relative efficiency of the DMUs and setting targets for the future programming in the framework of quantum algorithms. This attitude can include the decision maker's preferences, especially in target setting processes better than any existing methods.

Finally, Let us discuss about the main advantages of the quantum algorithms for solving DEA models over the existing traditional DEA approaches:

- The inputs/outputs in DEA models can be presented by quantum data. For example, for evaluating the relative efficiency in the bank branches, the customer satisfaction as output or the employee's behavior as input can be presented by quantum data which have much more capability than fuzzy data as it processes the data in an enormous different modes at the same time.
- Instead of traditional algorithms for solving DEA models, especially the nonlinear ones, the quantum algorithms can be replaced as they have more speed and processing in comparison with the old algorithms.

4 Conclusion

In this paper, a new approach was presented to show how quantum algorithms can solve DEA models by two different methods. In the first method, we can use the efficiency score associated with states in a DEA model to adjust a superposition to increase the amplitudes associated with low-efficiency score states. Given matrix (includes inputs and outputs in DEA models) with large condition number, the second method can select to invert only the part of which is in the well-conditioned part of . The main purpose of this paper is the connection between DEA models and quantum algorithms. This subject can be considered a starting point for researchers to propose new quantum algorithms in DEA models. Due to the increasing growth of quantum theory these days, this paper brings DEA to the world of quantum algorithms and information.

Finally, according to a 2021 report by Deloitte, many applications which can be transformed by quantum computing approaches are listed. It is worthy to mention that, It is difficult for quantum computers to simulate in a classical manner. Programming in quantum computers is too hard as it leads to errors, faults and loss of quantum coherence. Hope these limitations will be rectified in the future quantum computing and algorithms.

References

[1] M.S. Bazaraa, J.J. Jarvis and H.D. Sherali, *Linear programming and network flows*, John Wiley and Sons, 2010.

[2] A. Charnes, W.W. Cooper and E.L. Rhodes, *Measuring the efficiency of decision making units*, Euro. J. Oper. Res. **2** (1978), 429–444.

[3] K. Chen, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, 2005.

[4] W.W. Cooper, LM. Seiford and K. Tone, *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-solver Software*, Kluwer Academic Publishers: Norwell, MA, 2000.

[5] A.W. Harrow and A. Hassidim, S. Lloyd, *Quantum algorithm for linear systems of equation*, Phys. Rev. Lett. **103** (2009), 150–502.

[6] T. Hogg and D. Portnov, *Quantum optimization*, Inf. Sci. **128** (2000), 181–197.

[7] D. McMahon, *Quantum computing explained*, John Wiley and Sons, 2008.

[8] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, New York, 2010.

[9] M. Side and V. Erol, *Applying quantum optimization algorithms for linear programming*, doi:10.20944/preprints201703.0238.v2, April 2017.

[10] D.A. Spielman and S.H. Teng , *Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems*, SIAM J. Matrix Anal. Appl. **35** (2014), no. 3, 835–885.

[11] J. Sun, J. Liu and W. Xu, *Using quantum-behaved particle swarm optimization algorithm to solve non-linear programming problems*, Int J. Comput. Math. **84** (2007), 261–272.

[12] A.D. Athanassopoulos, N. Lambroukos and L.M. Seiford, *Data envelopment scenario analysis for setting targets to electricity generating plants*, Euro. J. Operational Res. **115** (1999), 413–428.

[13] AJ. Manuel, S. Carmen and S. Lozano, *A fuzzy DEA slacks-based approach*, J. Comput. Applied Math. **404** (2022), 113–180.

[14] F. Hosseinzadeh Lotfi, A. Ebrahimnejad, M. Vaez-Ghasemi and Z. Moghaddas, *Data Envelopment Analysis with R*, Springer, 2020.

[15] R. Banker, *Stochastic data envelopment analysis*, Data Envelop Analy. J. **5** (2021), 281–309.