A SpatioTemporal Data Envelopment Analysis (S-T DEA) approach: The need to assess evolving units

Konstantinos Petridis[1,*], **Alexander Chatzigeorgiou**[2], **Emmanouil Stiakakis**[2]

[1]Aston Business School, Aston University, Birmingham B4 7ET, United Kingdom

[2]Department of Applied Informatics, University of Macedonia, 54006 Thessaloniki, Greece

Email: k.petridis@aston.ac.uk, achat@uom.gr, stiakakis@uom.gr

*Corresponding author, e-mail: k.petridis@aston.ac.uk, Tel: +44 (0) 121 204 4900

Abstract

One of the major challenges in measuring efficiency in terms of resources and outcomes is the assessment of the evolution of units over time. Although Data Envelopment Analysis (DEA) has been applied for time series datasets, DEA models, by construction, form the reference set for inefficient units (lambda values) based on their distance from the efficient frontier, that is, in a spatial manner. However, when dealing with temporal datasets, the proximity in time between units should also be taken into account, since it reflects the structural resemblance among time periods of a unit that evolves. In this paper, we propose a two-stage spatiotemporal DEA approach, which captures both the spatial and temporal dimension through a multi-objective programming model. In the first stage, DEA is solved iteratively extracting for each unit only previous DMUs as peers in its reference set. In the second stage, the lambda values derived from the first stage are fed to a Multiobjective Mixed Integer Linear Programming model, which filters peers in the reference set based on weights assigned to the spatial and temporal dimension. The approach is demonstrated on a real-world example drawn from software development.

Keywords: Data Envelopment Analysis, Efficiency, OR in software, Multiobjective Programming, Linear Programming

1. Introduction

The units of most economic, business, and technological environments are by nature evolving systems that change continuously over time. The analysis of such systems calls for an evaluation of their efficiency in terms of outputs and inputs treating each snapshot in time as a separate unit to be assessed. As an example, the efficiency of any company, assessed over several successive months, quarters or years will yield different values that constitute the evolution of its qualities. In the world of software systems, products also evolve over numerous releases with additional features which should be assessed, when the development team is interested in evaluating software quality evolution.

In the context of evolution analysis, when identifying deficiencies in terms of resources and outcomes, it is valuable to determine earlier and efficient versions which can serve as benchmarks. The examination of these benchmarks can provide insight into possibilities for improvements and set goals for inputs and outputs of the inefficient units. The most widely acknowledged approach, that enables benchmarking of different units in terms of efficiency, is Data Envelopment Analysis (DEA) (Charnes et al. 1978). One of the main properties of DEA is the ability to extract a reference set for each inefficient Decision Making Unit (DMU) which contains the efficient units that operate close to the inefficient one (Coelli et al. 2005).

By definition, the efficient frontier provided by DEA considers only the spatial distance among units since DEA constructs an efficient frontier which is a surface enveloping all sampled units, whereas inefficiencies are calculated in relation to that surface. However, when benchmarking is applied on a time series dataset, the temporal dimension should also be considered in conjunction with the spatial dimension when extracting the efficiency score, the reference set, and the projected values on the efficient frontier. Moreover, in various contexts it is desirable to have a single peer as a benchmark to facilitate the process of comparing an inefficient unit to an efficient one in order to identify opportunities for improvement.

Let us consider a company which is evaluated for a number of 10 consecutive time periods. We assume that the DMU corresponding to the 10^{th} period is inefficient and has in its reference set two previous DMUs, namely the 2^{nd} with a lambda value of 0.6 and the 8^{th} with a lambda value of 0.4. In case the analysis aims at identifying a single peer to be used as a benchmark for the 10^{th} DMU, according to the conventional application of DEA one would select the 2^{nd} DMU which has the largest resemblance. However, such a choice would neglect the time dimension, as the 8^{th} DMU, which is still similar in terms of lambda value, has a significantly larger proximity to the DMU of interest. Such a similarity in time is of vital importance in many cases, as the analyst is interested in structural similarities which might vanish for units that are distant in terms of time.

In this paper, we propose a two-stage DEA approach which captures both the spatial and temporal dimension through a multi-objective programming model. In the first stage, DEA is solved iteratively extracting for each unit only previous DMUs as peers in its reference set. In the second stage, the lambda values derived from the first stage are filtered through the selection of weights assigned to the spatial and temporal dimension. The model is formulated so as to select a single out of multiple past peers in the reference set that can serve as a benchmark for comparison.

The rest of the paper is organized as follows: Section 2 provides an overview of previous work on the application of DEA on time series datasets. The need to assess the evolution of efficiency over time and to consider both the temporal and spatial dimension is discussed in Section 3, along with an example in software systems. The problem is demonstrated through an illustrative example in Section 4. The mathematical formulation of the problem and the associated proofs are provided in Section 5. The approach is applied on the illustrative example and the results are discussed in Section 6. A real-world example in a software development context is presented in Section 7. Finally, we conclude in Section 8.

2. Related work

DEA has been employed in several cases where time series data are treated as DMUs. Time series data constitute a significant format in which data variability is reflected (Cook and Seiford 2009). In many applications of the DEA methodology, data for the same entity are available at different points in time. These points are then treated as different DMUs. With regard to the methodological approach that should be adopted when DEA is used with time series data, two noteworthy approaches are the following: (i) “window analysis”, proposed by Charnes et al. (1984), where the basic idea is to regard a DMU in each period of time as if it were a different DMU and compare the efficiency of the DMU with its efficiency in other time periods and with other DMUs in the same time period (Inuiguchi and Mizoshita 2012; Chen and Johnson 2010; Bergendahl 1998) and (ii) DEA-based Malmquist index, proposed by Färe et al. (1994), which is an index with two components, one measuring the change in the technology frontier and the other the change in technical efficiency (Cook and Seiford 2009; Emrouznejad and Thanassoulis 2010;

Emrouznejad and Thanassoulis 2005; Lozano and Villa 2010; Grifell-Tatjé and Lovell 1997).

With regard to the domains that DEA has been applied, when time series data are used, some indicative applications are mentioned below. Hashimoto and Kodama (1997) evaluated livability in Japan for the period 1956-1990 regarding each year as a separate DMU. It is one of the studies which are considered as non-standard DEA applications since positive and negative social indicators were used as inputs and outputs. In another study, DEA was the tool for performance measurement and target setting of manufacturing systems (Jain et al. 2011). The DEA methodology was applied to two different contexts, i.e. (i) a traditional assembly line and (ii) an advanced wafer manufacturing unit. The performance of both manufacturing systems was evaluated on a weekly basis, so that each week was regarded as a distinct DMU. Lynde and Richmond (1999) analyzed quarterly data of the manufacturing sector from 1966 to 1990 in the context of explaining productivity growth in UK. Through the analysis of a set of time series data, it was made possible to define the role of technical progress, technical efficiency, and input slack concerning the growth of total factor productivity. In another application with time series data, DEA was used to evaluate the performance of local exchange carriers from 1997 to 2007 (Moreno et al. 2013). The main feature of this approach is that a global assessment of the performance of a company along the whole time period was attained. The productivity and performance of OECD countries were evaluated based on a dynamic DEA model (Emrouznejad 2003). Movahedi et al. (2007) assessed the efficiency of Iranian Railway for a time series dataset from 1971 to 2004. The efficiency of each year was calculated and compared to the other years. After identifying the efficient years, the super efficiency DEA (Andersen and Petersen 1993) model was employed for an analytical ranking (Charnes et al. 1984). The DEA methodology was also applied to the context of a specific hospital (Rutledge et al. 1995). More specifically, it was employed to determine the efficiency of a non-profit hospital in the southeast United States for a time period of 22 consecutive months. DEA results contributed to managers' decisions as to which months required further attention. Finally, the efficiency scores for the 7 largest Canadian Schedule I banks, over a 10-year period from 1998 to 2007, were obtained using DEA window analysis. Following that, the Malmquist productivity index was used to calculate the productivity changes (Cao and Yang 2009; Chen and Yu 2014). Despite the fact that all the previous works employ time series data sets, there has not been any effort to assess evolving units in terms of temporal and spatial distance of DMUs, which is the purpose of this paper.

3. The need for assessing the evolution of quality over time

For most economic, physical, technological, and other systems, efficiency, in whatever manner it is measured, changes over time as a result of modifications to the underlying system. As an example, the efficiency of any company, assessed over several successive months, quarters or years will yield different values that constitute the evolution of its qualities. In the same way that DEA can be applied for benchmarking different DMUs, it can be applied for benchmarking successive snapshots in time of the same unit, providing an overview for the evolution of its efficiency. This is consistent with the need for continuous learning in contemporary organizations so that managers can sense and respond rapidly and flexibly to change, as claimed in (Avkiran 2009a).

When DMUs refer to multiple instances of the same context at the same time period (e.g. several enterprises/companies assessed in a given year), DMUs have full similarity to be potential peers. In other words, the external conditions are the same for all DMUs. However, when a time series of the same entity is analyzed, then, by definition, the DMU evolves over time. Consequently, the external conditions may differ as it is, for example, the case when assessing the same enterprise/company over successive time snapshots, which may span for several years. It is this particular context in which the proposed S-T DEA approach is applicable.

As an example, let us consider a software system that evolves over time and is distributed as a number of successive releases. If one release is inefficient, previous releases which would be suitable for comparison are efficient releases in near time instances. These software versions have functionality which is comparable, have been developed by more or less the same personnel and rely on similar technologies, assumptions which do not hold for releases which are too distant in time.

In an economic context, one could draw an example from a national economy which is gradually transforming from a merely agricultural one to a more industrial and then to a more technological one. If this kind of parameter is not taken as an input of the DMUs under study (i.e. economies over successive years), then it constitutes an external condition. It is exactly those changes in the external conditions which render the consideration of the time dimension in a DEA approach significant.

When treating each version in time of a company, project or process as a distinct DMU, the application of DEA will yield for each time instance an efficiency score, as well as the corresponding reference set. Let us assume that DEA is applied on DMU(t), DMU(t+1), …, DMU(t+i), where the parentheses indicate the time instance of each examined version. Once the reference set for each DMU is obtained, the goal of the analysts is to locate the most appropriate peer in the reference set, in order to compare an inefficient DMU to an efficient one and identify opportunities for improvement.

Although, one could consider all DMUs in the reference set, this might often be prohibitive in terms of required effort, as the involved systems can be extremely large and complex. To identify the single, most appropriate peer from the ones listed in the reference set of the DMU under study, the following should be taken into account:

- It would not make much sense to identify as reference unit a future DMU, as the comparison to that version would be infeasible. In other words, for DMU(t), we could not select DMU(t+k) as a reference project, since at the time of DMU(t) development or assessment, DMU(t+k) does not exist. In other words, future DMUs should be excluded as candidate reference projects.
- It appears to be wise to consider past DMUs that are closer in time to the DMU of interest. As the underlying system evolves over time, a distant DMU in terms of time might have significantly different properties rendering the comparison less valuable.
- Finally, as in the conventional application of DEA, DMUs that have a higher degree of resemblance to the examined one (i.e. have a larger lambda value in the reference set) should be preferred, as they correspond to systems, that are more similar in terms of inputs and outputs.

Consequently, the selection of a single project from the reference set of a DMU, in case DMUs represent time snapshots of the same system, involves the resolution of a trade-off between similarity captured by lambdas and proximity in time. The straightforward approach to consider two dimensions (time and space) would be to employ a geometrical average of the two distances. In other words, one could consider the distance in time and space as two vectors, which are orthogonal if equal weights are assigned to both dimensions. Then, it would be possible to obtain the resultant of the two vectors for each peer of a DMU under study, and the peer with the minimum length would be the one that should be selected. However, such a geometrical approach would not be able to extract the efficiency of the unit under study, as well as the associated slacks (with regard to those two dimensions). The reasons, for which time series analysis is important to be applied in a software context, are discussed next.

Contemporary software products are extremely complex systems, often consisting of thousands of components which host numerous functions and pieces of data. The partitioning of a software system into components, the allocation of functionality and state to these components, and the specification of interconnections among them constitute the software's architecture or design. The quality of the architecture can be assessed employing certain metrics, such as coupling and cohesion, which quantify corresponding qualitative attributes. The underlying architecture of a software system reflects upon software's external qualities, such as maintainability, reusability, and comprehensibility.

During the initial construction of a software system, its design quality is in general of a high level. However, software systems suffer from the so called "ageing" symptom (Parnas 1994), meaning that as software evolves (i.e. when new versions are released to deliver additional functionality or to fix bugs), its architecture gradually deteriorates. Thus, it becomes crucial for software developers and maintainers to assess the evolution of quality over successive software versions. Each version constitutes a DMU and the role of DEA would be to assess the efficiency of each unit and to propose, for the inefficient software versions, a single past version in the reference set that can serve as a benchmark for comparison.

4. Problem definition

The need to consider both the spatial and temporal dimensions in the selection of efficient peers, when assessing DMUs that represent different snapshots in time for a given entity, can be better explained through an illustrative example. Let us consider an entity having two outputs and a single unitized input that is being evaluated for nine successive periods. The corresponding (fictional) data are shown in Table 1.

Table 1. Data for the illustrative example

DMU	Input	Output1	Output2
DMU(1)	1	6	4
DMU(2)	1	13	7
DMU(3)	1	3	9
DMU(4)	1	4	11
DMU(5)	1	11	12
DMU(6)	1	15	4

DMU(7)	1	6	12
DMU(8)	1	16	9
DMU(9)	1	10	8

The application of a DEA output-oriented model yields a reference set which for each inefficient unit contains the efficient snapshots that operate closer in terms of outputs to the examined DMU (results are shown in Table 2). In other words, only the spatial dimension (i.e. the similarity as calculated by the lambda values) is taken into consideration. The temporal dimension is related to the proximity in time between two units. As already mentioned in Section 3, it would be valuable to propose only past units as efficient peers in the reference set, select a single one to be employed as a benchmark, and moreover allow the user to assign weights to the spatial and temporal dimensions.

As it can be observed from Table 2, the DMU that corresponds to the 9th time period, has two non-zero lambdas representing the efficient peers in its reference set. The trade-off in the selection of a single peer is vividly revealed since DMU(9) is closer in terms of space to DMU(5) ($\lambda_5 > \lambda_8$), while it is closer in terms of distance in time to DMU(8) ($t_9 - t_8 < t_9 - t_5$).

Table 2. Reference set of illustrative example

DMU	φ	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9
1	2.447	0	0	0	0	0.263	0	0	0.737	0
2	1.231	0	0	0	0	0	0	0	1	0
3	1.333	0	0	0	0	1	0	0	0	0
4	1.091	0	0	0	0	1	0	0	0	0
5	1	0	0	0	0	1	0	0	0	0
6	1.067	0	0	0	0	0	0	0	1	0
7	1	0	0	0	0	1	0	0	0	0
8	1	0	0	0	0	0	0	0	1	0
9	1.329	0	0	0	0	**0.543**	0	0	**0.457**	0

Consequently, our goal is to enhance the classical DEA output-oriented approach applied to time series data, in order to extract for an inefficient DMU a single past efficient peer from its reference set considering both the objectives of minimizing the distance in time and maximizing the similarity in space. Moreover, once a single peer is selected, the efficiency score, and the projected values and slacks for each output should be recalculated.

5. Mathematical formulation of S-T DEA

5.1 Notation

Indices/Sets

$i \in I$: DMUs

$\tau \in D \subseteq I$: DMUs ($\tau \geq \mu = \max\{n \cdot m,\ 3 \cdot (n+m)\}$)

$r \in R$:Output

$l \in I$: Reference set

j : Iterations

Parameters

w_{sp}^{j} : Weight of spatial criterion at iteration *j*

w_{t}^{j} : Weight of temporal criterion at iteration *j*

y_{ri} : Output *r* of DMU *i*

λ_i^* : Optimal solutions of lambdas for DMU *i*

$ORD(\bullet)$: Function that attributes the order of set $\bullet$

$\mathbf{A}$: Matrix for storing the reference set of each DMU

$\mathbf{\Delta}$: Matrix for storing the difference between the current DMU and its temporally closest

λ_i^{MAX} : Maximum lambda from the reference set of *i*

δ_i^{MIN} : Minimum temporal distance between DMU *i* and all of its peers

Continuous variables

λ_i : Lambda of each DMU

φ : Efficiency

$\hat{\varphi}$: SpatioTemporal Efficiency **(decision variable)**

$\hat{s}_r^+$: Slack variable for each output (S-T DEA approach) **(decision variable)**

$\hat{y}_{ri}$: Projected output *r* of DMU *i* (S-T DEA output oriented model) **(decision variable)**

Binary variable

ζ_l : 1 if lambda *l* is selected, 0 otherwise **(decision variable)**

5.2 Calculating efficiency scores and reference sets

In this section, the mathematical formulation of the proposed S-T DEA is presented. The main scope of this new approach is the selection of the reference set, under the criteria of space and time, maximizing the efficiency that r outputs produce given a single unitized input. The proposed model can be generalized for the case when multiple inputs are present; however, we illustrate the case of no inputs since in various domains (such as software systems discussed in Section 6) evaluation can be performed solely using outputs.

The proposed model can place emphasis towards either a spatial or a temporal objective through a multi-objective programming formulation using a Weighted Sum Model (WSM) approach to handle the two objectives. The approach has two stages: a) in the first stage, the initial Linear Programming (LP) model for the output-oriented DEA is solved iteratively in order to extract the initial lambda values that capture the similarity of each DMU to its peers, and b) these values are then fed to the second stage that seeks to satisfy both aforementioned objectives. It should be noted that the DEA model applied in the first stage can assume either variable returns to scale (VRS) or constant returns to scale (CRS). Multi-stage models employing DEA have also been proposed, such as the three-stage DEA/SFA (Stochastic Frontier Analysis) approach by Avkiran and Rowlands (2008) in order to account for measurement errors or environment noise, and the four-stage approach by Avkiran (2009b), where DEA is employed in the first and fourth stage to eliminate the impact of exogenous factors on managerial efficiency.

To prohibit the selection of future units as peers in the initial reference set for each DMU, we apply the DEA output-oriented model iteratively, considering in each iteration only units that precede the DMU under examination (e.g. when assessing DMU(5), only DMU(1)…DMU(5) are being fed to the model). However, an empirical rule dictates that the number of DMUs should be equal or greater than $\mu = \max\left\{n \cdot m,\ 3 \cdot \left(n+m\right)\right\}$, where n and m is the number of inputs and outputs, respectively (Cooper et al. 2007). Therefore, the iterations start from this lower bound and continue up to the number of DMUs. This can be formulated as follows:

For $\tau = \mu, \ldots, |I|$

$$\max \ \varphi \quad (1)$$

s.t.

$$\sum_{i \le \tau} \lambda_i = 1 \quad (2)$$

$$\sum_{i \le \tau} y_{ri} \cdot \lambda_i \ge y_{r\tau} \cdot \varphi, \ \forall r \quad (3)$$

$$\lambda_i \ge 0, \ i \le \tau \quad (4)$$

End For

Assuming that there are ten DMUs under examination and each DMU produces two outputs, then an analytical expansion of the aforementioned formulation for two consecutive iterations is shown in Table 3. The values of the virtual problem are: $|I| = 10$ (ten DMUs), $\tau = 9, \ 10$ implying that solutions can be obtained for DMU(9) and DMU(10) and $r = 2$ (two outputs).

Table 3. Two consecutive iterations of DEA so as to include only past units for each DMU

1^{st} Iteration	2^{nd} Iteration
$\max \ \varphi$	$\max \ \varphi$
s.t.	*s.t.*
$\lambda_1^1 + \lambda_2^1 + \lambda_3^1 + \ldots + \lambda_9^1 = 1$	$\lambda_1^2 + \lambda_2^2 + \lambda_3^2 + \ldots + \lambda_{10}^2 = 1$
$y_{1,1} \cdot \lambda_1^1 + y_{1,2} \cdot \lambda_2^1 + \ldots + y_{1,9} \cdot \lambda_9^1 \ge y_{1,9} \cdot \varphi$	$y_{1,1} \cdot \lambda_1^2 + y_{1,2} \cdot \lambda_2^2 + \ldots + y_{1,10} \cdot \lambda_{10}^2 \ge y_{1,10} \cdot \varphi$
$y_{2,1} \cdot \lambda_1^1 + y_{2,2} \cdot \lambda_2^1 + \ldots + y_{2,9} \cdot \lambda_9^1 \ge y_{2,9} \cdot \varphi$	$y_{2,1} \cdot \lambda_1^2 + y_{2,2} \cdot \lambda_2^2 + \ldots + y_{2,10} \cdot \lambda_{10}^2 \ge y_{2,10} \cdot \varphi$
$\lambda_1^1, \lambda_2^1, \ldots, \lambda_9^1 \ge 0$	$\lambda_1^2, \lambda_2^2, \ldots, \lambda_{10}^2 \ge 0$

As it can be seen from the output of the two LP models presented in Table 3, only the reference set and the efficiencies for DMU(9) and DMU(10) are derived. Therefore, the reference set of each DMU will include only previous DMUs. In Table 3, the superscript of the λ variable corresponds to the iteration of the problem, whereas the subscript indicates the corresponding peer of the DMU under investigation.

From the analysis of the aforementioned iterative DEA model, a DMU(τ) corresponding to a particular time point can have at maximum τ peers as reference set and thus τ lambda values, denoted as λ_i^*, $i \le \tau$. Let $\mathbf{A}$ be a matrix with dimensions $|I| \times |I|$, where I is the number of all DMUs. In each row τ of the matrix the lambda values λ_l^* corresponding to each DMU(τ) are stored. For instance, in the 6th row of matrix $\mathbf{A}$ in Figure 1, the two non-zero lambdas imply that DMU(6) has in its reference set DMU(2) and DMU(3). The elements of matrix $\mathbf{A}$ are denoted as a_l^τ.

In order to enable normalization, the maximum value of each row will be required. Let λ_τ^{MAX} be an $|I| \times 1$ vector containing the maximum values of λ_τ^* according to (5):

$$\lambda_\tau^{MAX} = \max_l \left\{ a_l^\tau \right\}, \ \forall \tau \tag{5}$$

Matrix $\mathbf{A}$ is used in order to model the spatial dimension in the proposed S-T DEA approach.

In order to capture the temporal dimension, the distance in time between the DMU under examination and each efficient peer in its reference set should be computed. Let $\mathbf{\Delta}$ be a matrix with the same dimensions as $\mathbf{A}$ containing the time distance (difference in their order) between DMU(τ) from its peers. There is no point to compute these differences between a DMU and a peer in its reference set when the corresponding λ^* equals to zero. For instance, in the 6th row of matrix $\mathbf{\Delta}$ in Figure 1, DMU(6) has a time distance of 4 with DMU(2), and a distance of 3 with DMU(3). The other four distances are set to *M*, where *M* is a very large positive number ($M >> 0$) in order to exclude the corresponding elements from the formulation of the objective function. The elements δ_l^τ of matrix $\mathbf{\Delta}$ are obtained as follows:

$$\delta_l^\tau = \begin{cases} ORD(\tau) - ORD(l), \ a_l^\tau \neq 0 \\ M, a_l^\tau = 0 \end{cases} \tag{6}$$

In order to enable normalization, the maximum value of each row will be required. Let δ_τ^{MIN} be an $|I| \times 1$ vector containing the maximum values of distances in time, which are stored in δ_l^τ according to (7).

$$\delta_\tau^{MIN} = \max_l \left\{ \delta_l^\tau \middle| \delta_l^\tau \neq M \right\}, \ \forall \tau \tag{7}$$

Matrix $\mathbf{\Delta}$ is used in order to model the temporal dimension in the proposed S-T DEA approach.

The aforementioned process is shown in Figure 1.

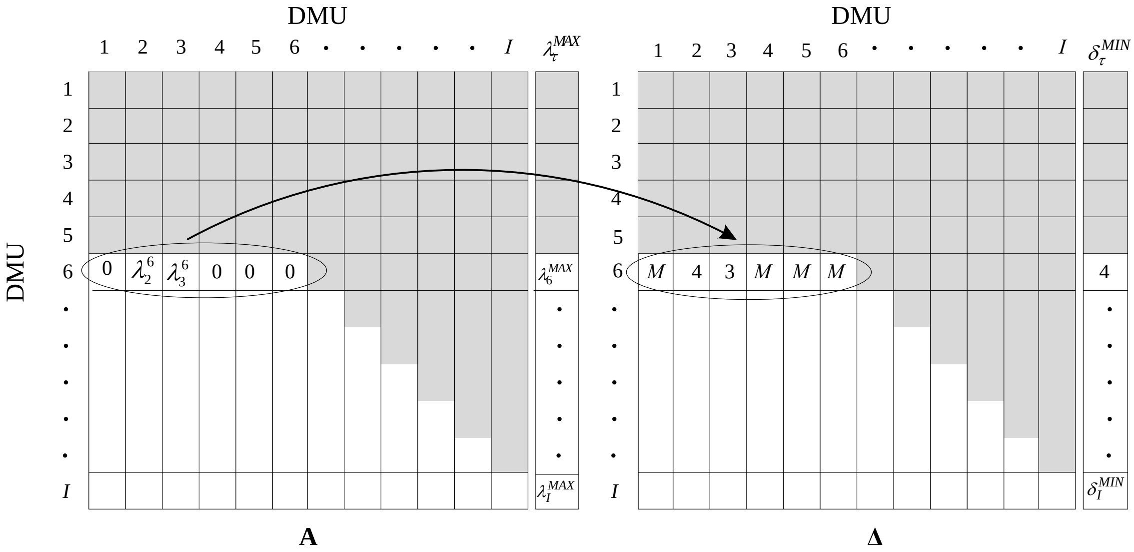

Figure 1. Construction of Tables **A** and **Δ**

In the objective function of the proposed S-T DEA model, the spatial and the temporal criterion are integrated in a single function, using a Weighted Sum Model (Freed and Glover 1981). This approach allows the handling of multiple objectives, as the final objective function consists of the weighted sum of the sub-objectives. The aim of this approach is to maximize the spatial component while minimizing the temporal component. The S-T DEA model is presented below:

For $\tau = \mu, \ldots, |I|$

For $j = 1, \ldots, J$

$$\max \mathrm{w}_{sp}^{j} \cdot \frac{1}{\lambda_\tau^{MAX}} \cdot \sum_l a_l^\tau \cdot \zeta_l - \frac{1}{\delta_\tau^{MIN}} \cdot \mathrm{w}_t^j \sum_l \delta_l^\tau \cdot \zeta_l \tag{8}$$

s.t.

$$\sum_{l \le \tau} y_{rl} \cdot \zeta_l \ge y_{r\tau} \cdot \hat{\varphi}, \ \forall r \tag{9}$$

$$\hat{\varphi} \ge 1 \tag{10}$$

$$\sum_{l \le \tau} \zeta_l = 1 \tag{11}$$

$$\mathrm{w}_{sp}^j + \mathrm{w}_t^j = 1, \ \forall j \tag{12}$$

$$\zeta_l \in \{0,1\}^{|I|} \tag{13}$$

End For

End For

The outer loop iterates over the DMUs under investigation, for $\tau \geq \mu$, so as to obtain only previous peers for each DMU(τ). The inner iterations represent incremental changes applied to the weight assigned to the spatial or temporal objective, such that $\mathrm{w}_{sp}, \mathrm{w}_t \in [0,1]$. In this way, all possible weight combinations for both dimensions are examined (results in the following section are provided for a step equal to 0.01). As already mentioned, the goal of the proposed approach is to extract a unique efficient peer, based on the weights assigned to the time and space dimension. Therefore, in the above formulation of S-T DEA, in constraint (9) which is a reformulation of constraint (3), with the exception of the use of index *l* instead of *i*, lambdas have been replaced by binary variables ζ_l which take a value of 1 if lambda value l is selected and 0 otherwise. Constraint (10) has been introduced since the projected values for the outputs of the DMU under examination cannot be worse than the original ones. This constraint implies that $\hat{\varphi} \geq 1$. Constraint (11) is introduced so as to guarantee that only one peer will be selected for the reference set of the examined DMU and constraint (12) implies that the weights assigned to the objective function are complementary. The resulting S-T DEA formulation, due to the presence of binary variables, is a Mixed Integer Linear Programming model (MILP).

The extracted solutions pertain to the selected peer (past DMU) and the calculated efficiency score ($\hat{\varphi}$) based on the efficient frontier formed by the single selected peer. Out of these extracted solutions (depending on the selected weights) one should select the peer that maximizes the efficiency score, since this implies that the DMU under study has to cover the least required distance to become efficient.

The resulting efficient frontier considering a single past peer is illustrated in Figure 2(a), where DMU(t) has in its reference set DMU(t_A) and DMU(t_B) (other DMUs are excluded for clarity). In case DMU(t_A) is selected as a peer, the $\hat{\varphi}$ value of DMU(t) will be higher than unity ($\hat{\varphi} = OA'/OA > 1$), indicating the extent by which its outputs should be improved to move onto the efficient frontier. Since $\hat{\varphi} \geq 1$ the solution will be considered as valid.

In case the extracted efficiency score ($\hat{\varphi}$) for a certain combination of weights is less than unity, as shown in Figure 2(b) where DMU(t_B) is selected as a peer and thus $\hat{\varphi} = OB'/OB < 1$, the corresponding solution should be considered as invalid due to the constraint (10). In case constraint (10) was omitted, such a solution would imply that the DMU under study would be beyond the efficient frontier formed by the selected peer.

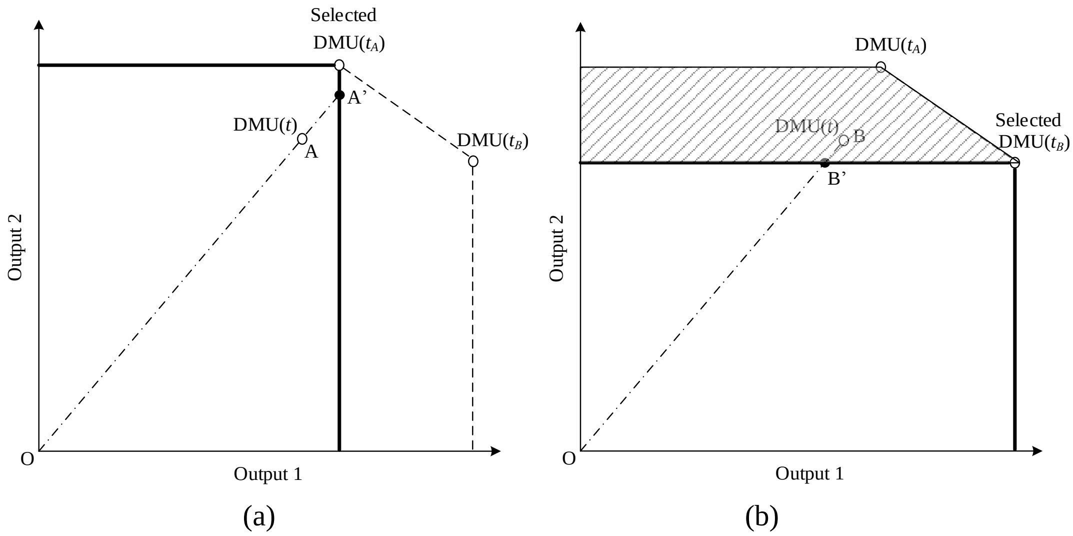

Figure 2. Efficient frontier formed by a single selected peer (illustrative example) (a) valid solution, $\hat{\varphi} \geq 1$, (b) invalid solution, $\hat{\varphi} < 1$

5.3 Introducing slacks

The aforementioned S-T DEA approach can be extended so as to provide more information about the projected values of each output, based on the spatial and temporal dimensions. A prerequisite for the calculation of the projected values is the estimation of slack values. As known, apart from the efficiency score of a DMU, the notion of slacks is necessary to determine whether a DMU is 'fully efficient' or not (Charnes et al. 1981). Slacks are input excesses or output shortages meaning that a DMU on the efficient frontier can still have room for improvement.

To this end, in the objective function (8) of the previous formulation, slacks are introduced in (14) and the goal is to minimize their values (in the weighted sum product ε represents a very small positive number, such that ε ~ 0, whereas $\hat{s}_r^+$ represents the slack value assigned to output r. A recommended range for ε is $10^{-5} \leq \varepsilon \leq 10^{-3}$. Based on the calculated slacks, the projected outputs for each inefficient DMU are calculated in (18) using variable ($\hat{y}_r$) depending on the corresponding weights in the objective function.

For $\tau = \mu, \ldots, |I|$

For $j = 1, \ldots, J$

$$\max \ \mathrm{w}_{sp}^{j} \cdot \frac{1}{\lambda_{\tau}^{MAX}} \cdot \sum_{l} a_{l}^{\tau} \cdot \zeta_{l} - \frac{1}{\delta_{\tau}^{MIN}} \cdot \mathrm{w}_{t}^{j} \sum_{l} \delta_{l}^{\tau} \cdot \zeta_{l} - \varepsilon \cdot \sum_{r} \hat{s}_{r}^{+} \quad (14)$$

s.t.

$$\sum_{l \leq \tau} y_{rl} \cdot \zeta_{l} - \hat{s}_{r}^{+} = y_{r\tau} \cdot \hat{\varphi}, \ \forall r \quad (15)$$

$$\hat{\varphi} \geq 1 \quad (16)$$

$$\sum_{l \leq \tau} \zeta_{l} = 1 \quad (17)$$

$$\hat{y}_{r} = y_{r} \cdot \hat{\varphi} + \hat{s}_{r}^{+}, \ \forall r \quad (18)$$

$$\mathrm{w}_{sp}^{j} + \mathrm{w}_{t}^{j} = 1, \ \forall j \quad (19)$$

$$\zeta_{l} \in \{0,1\}^{|I|} \quad (20)$$

$$\hat{y}_{r} \geq 0, \ \forall r \quad (21)$$

$$\hat{s}_{r}^{+} \geq 0, \ \forall r \quad (22)$$

End For

End For

5.4 Proofs

The proposed approach guarantees that there will always be a feasible solution (proofs of the following propositions are given in Appendices A and B). A feasible solution of a mathematical programming model is a solution that satisfies all the constraints of the problem. The next two propositions guarantee that the proposed model does not yield infeasible solutions for any given parameter set.

Proposition 1: *For any DMU τ and for any λ resulting from S-T DEA, and y (output), it stands that* $\varphi \geq \hat{\varphi}$

where:

φ *is the maximum efficiency derived from the LP model* (1) – (4) *and* $\hat{\varphi}$ *the maximum efficiency from the S-T DEA model* (8) – (13).

Proposition 2: *For any given parameter set* (*inputs – outputs*), *S-T DEA always provides a feasible solution.*

5.5 Handling of multiple optima

As pointed out in the literature, the problem of non-uniqueness of results in the presence of alternative optima might be encountered in the application of DEA (Banker et al. 2011). This problem, which manifests itself as alternate sets of optimal lambda values, has been extensively studied in the context of the estimation of Returns to Scale (RTS) (Seiford and Zhu 1998).

In analogy to the work of Anderson and Inman (Anderson and Inman 2011), we employ the following heuristic in order to select a unique set of lambda values in the case of alternate optima. The approach is graphically depicted in Figure 3.

Let us assume that a given DEA formulation yields multiple optima for a DMU under study, which appears at time point *τ*.

Let us denote as λ_i^* the lambda value of an efficient peer in the reference set appearing at time point *t*, which is part of a set of solutions $i' \in F$. Then, the optimum set of solutions is chosen by the combination of peers which maximizes:

$$\sum_{i' \in F} \frac{\lambda_{i'}^*}{\Delta(\tau - t)} \tag{23}$$

In the following example, we assume that for DMU(9) two sets of solutions can be extracted. It is assumed that two sets of efficient solutions are derived; the first group consists of DMUs 5 and 8, while the second group comprises DMUs 6 and 7.

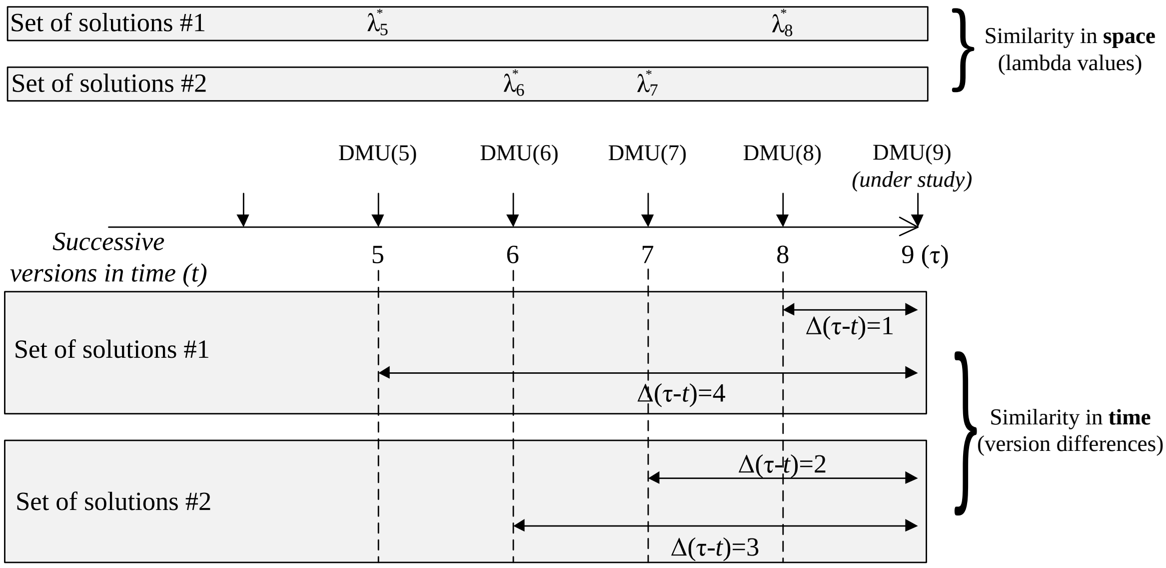

Figure 3. An approach of handling multiple optima

The optimum set of solution can be obtained by calculating which of the two solutions maximizes the corresponding sum:

$$solution\#1: \frac{\lambda_5}{\tau-5}+\frac{\lambda_8}{\tau-8}$$

$$solution\#2: \frac{\lambda_6}{\tau-6}+\frac{\lambda_7}{\tau-7}$$

The intuitive interpretation of this formula is that a distant lambda value, even it is high, will be counterweighted by the large Δt. On the other hand, a low lambda value that is close to the DMU under study will be amplified by the low Δt.

In the context of the proposed S-T DEA, checking whether multiple solutions exist should be performed at the end of the first stage.

6. Illustrative example

In this section, the results of the proposed S-T DEA model are analytically described when applied on the illustrative example, the data of which are provided in Table 1. Through the analysis, the ability of the proposed model to select the most suitable peer for an examined DMU is demonstrated.

As it can be seen in Table 2, the peers for DMU(9), which is inefficient, according to the conventional application of DEA, are DMU(5) and DMU(8). In this case, applying the S-T DEA model, DMU(5) is the only peer selected for DMU(9) when $0.83 \leq w_{sp} \leq 1$ because the lambda corresponding to DMU(5) is larger in value ($\lambda_5 = 0.543 > \lambda_8 = 0.457$). On the contrary, when $0.17 < w_t \leq 1$, DMU(8) is chosen because it is the temporally closest to DMU(9). The selected peers, depending on the applied weights, are illustrated in Figure 4.

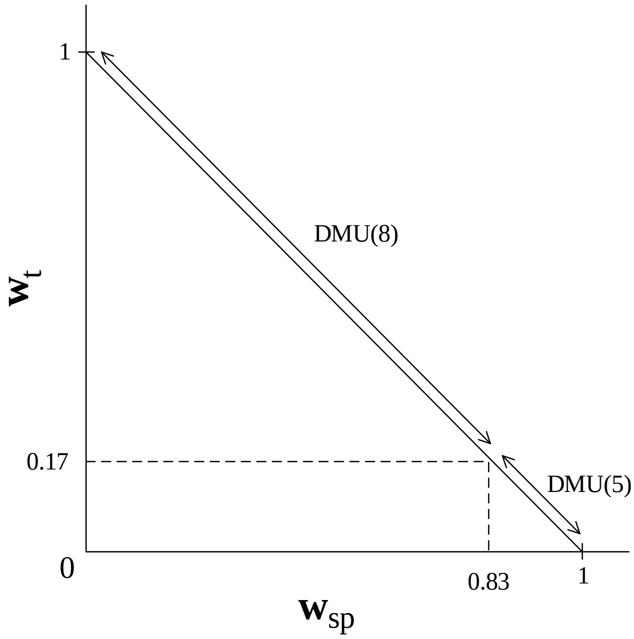

Figure 4. Selected peer according to w_{sp} and w_t of DMU(9) in the illustrative example

The proposed S-T DEA model, based on constraint (9) is capable of calculating a modified efficiency score for each DMU, which now reflects its relative position in the spatiotemporal context. This efficiency score depends on the weights assigned to the temporal and spatial dimension and thus obtains values which differ based on the selected peer. Consequently, it becomes possible to select the peer that maximizes this efficiency score providing a measure of resemblance between an examined DMU and a selected peer in the spatiotemporal context.

For the illustrative example, the extracted efficiency scores $\hat{\varphi}$ for DMU(9) are shown in Table 4. Based on Proposition 1, $\hat{\varphi}$ is now smaller in value than the original φ (for example, φ for DMU(9) was originally 1.329). Among the two potential peers, the selection of DMU(5) yields a greater efficiency score meaning that this is the closest peer to which the DMU under study should be compared.

The projected values for each of the outputs of DMU(9) based on (14) – (20) are also presented in Table 4. When the time dimension is weighted in the range $[0.17,1]$, S-T DEA model selects DMU(8), yielding for the first projected output a value of 16 and for the second a value of 9. These changes in the projected values are attributed to equation (18). When the space dimension is weighted in the range $[0,0.83]$, S-T DEA model selects DMU(5), yielding for the first projected output a value of 11 and for the second a value of 12.

Table 4. Calculated efficiency scores and projected values for DMU(9) in the spatiotemporal context based on the applied weights

Weight of the temporal dimension	**Selected Peer**	$\hat{\varphi}$	**Technical Efficiency Score** $1/\hat{\varphi}$	**Projections**
$0 \le w_t \le 0.17$	DMU(5)	1.100	0.909	output1: 10 →11
				output2: 8→12
$0.17 < w_t \le 1$	DMU(8)	1.125	0.888	output1: 10→16
				output2: 8→9

The illustrative example and the case study in Section 7 have been solved using GAMS optimization (Rosenthal 1988) and the CPLEX solver (Brooke et al. 2003).

7. Real world-application: A software example

The Eclipse Integrated Development Environment (Eclipse IDE) is one of the most popular programming platforms and Eclipse Java Development Tools (Eclipse JDT) Core is the component at the heart of the IDE. The COMETS repository (Couto et al. 2013) provides access to various metric values that characterize the evolution of Eclipse JDT Core quality over a period of 8 years (07/01/2001 – 06/14/2008). In this example, we analyze 19 versions of Eclipse JDT, taken approximately every 5 months of development. The selected versions constitute the DMUs that we wish to investigate. The metrics by which we assess the design quality of each version are coupling, cohesion, and depth of inheritance tree. Coupling quantifies the degree of inter-dependence among software modules and should be kept as low as possible; cohesion refers to the degree to which the elements of a module belong together and the goal is to maximize its value; and depth of inheritance tree expresses the average distance of a software module to the root module of the corresponding inheritance hierarchy (software is organized in inheritance hierarchies and it is preferable to have a low average depth). These three metrics determine the changeability of a software system (Samoladas et al. 2008), which is one of the sub-characteristics defined in the ISO/IEC 9126 quality model for software. Changeability refers to those attributes that reflect the effort required for software modification. These three metrics are the outputs of the corresponding DMUs. In the field of software engineering, the design quality of a software artifact is assessed by examining its metric values in isolation, rather than by contrasting them to the development effort that has been spent. In other words, the quality assurance teams or individuals in a software project would pay attention to the evolution of quality attributes (such as the ones reflected by the selected metrics) without investigating whether any improvement or deterioration should be attributed to increased or reduced development effort. Such a correlation could obviously be performed but this would be of interest to the field of software economics rather than software quality. In this example, we also focus on software quality assessment and thus we do not consider any inputs.

The overall goal is to assess the evolution of quality captured by the selected outputs and reflected in the efficiency of each version. Moreover, employing the extracted reference sets for each version, DEA enables the identification for each inefficient software version (i.e. versions for which there is room for improvement) of those past versions that can serve as reference projects. Designers can compare the current version of a given project to these reference versions, identify which aspects of the design have deteriorated and for which reason, and propose modifications for restoring the design quality.

The employed dataset for the examined versions of project Eclipse JDT Core is shown in Table 5. It should be mentioned that all listed outputs are undesirable, in the sense that the goal of the design is to minimize their values. Therefore, the actual data that have been fed to DEA are the inverse values. By taking the inverse values for all outputs, implying that the goal is to increase them, we essentially treat all outputs as desirable ones. The reason for selecting these particular outputs is that they constitute standard and widely acknowledged metrics, measurable by software engineering tools.

Table 5. Software example data (Eclipse JDT Core)

DMU	Coupling (CBO)	Lack of Cohesion (LCOM)	Depth of Inheritance Tree (DIT)
E01	12.27	132.17	3.11
E02	12.43	127.53	3.03
E03	13.29	124.07	3.21
E04	13.26	156.51	3.21
E05	13.62	166.10	3.19
E06	13.79	162.41	3.19
E07	13.91	187.72	3.08
E08	13.95	196.55	3.17
E09	14.61	237.36	3.19
E10	14.93	251.64	3.18
E11	15.11	250.86	3.13
E12	15.17	251.92	3.12
E13	15.58	249.49	2.60
E14	15.65	249.40	2.61
E15	15.68	228.47	2.60
E16	15.62	233.74	2.60
E17	15.71	240.43	2.60
E18	15.85	241.07	2.59
E19	15.86	235.36	2.62

The results of conventional DEA (VRS, output-oriented model), and more specifically the calculated efficiency scores and reference sets for each DMU of the dataset, are presented in Table 6. As it can be observed, the evolution of quality for this system does not exhibit any particular monotonous trend during the examined period. Efficient versions have only themselves in their reference set. Some of the inefficient versions (such as E04 to E12) have one past version (E02) in their reference set, and one future version (E16), which, as already explained, cannot be exploited, since the designers at the time of construction for each software version cannot take advantage of a future version for pinpointing areas that can be further improved.

The proposed S-TDEA approach is useful for cases, such as the last examined version, E19. This version has an efficiency score less than one, while its reference set contains three past versions, namely E15, E16, and E18. As already mentioned, it would be valuable to direct the designers of E19 to a single past project that can act as benchmark. However, the project of interest (E19) has a higher resemblance to the project that is further away in terms of time (E15), whereas the closest project in time (i.e. the preceding one, E18) has the lowest lambda value. In other words, the designers of E19 have to take a decision considering both the degree of similarity to the projects in the reference set (since a similar version is more appropriate for locating deficiencies), as well as the proximity in time (since a recent project will have properties that are akin to those of the examined version). A similar trade-off should be considered when assessing the reference project of choice in the case of version E17.

Table 6. Software example DEA results

DMU	Score	Reference set (lambda)
E01	1.0000	E01 (1)
E02	1.0000	E02 (1)
E03	1.0000	E03 (1)
E04	0.9406	E02 (0.985), E16 (1.51E-02)
E05	0.9324	E02 (0.895), E16 (0.104)
E06	0.9273	E02 (0.864), E16 (0.135)
E07	0.9421	E02 (0.747), E16 (0.252)
E08	0.9263	E02 (0.816), E16 (0.183)
E09	0.9047	E02 (0.708), E16 (0.291)
E10	0.8979	E02 (0.645), E16 (0.354)
E11	0.9021	E02 (0.568), E16 (0.431)
E12	0.9032	E02 (0.545), E16 (0.454)
E13	0.9994	E02 (1.35E-02), E16 (0.986)
E14	0.9970	E02 (3.94E-03), E16 (0.996)
E15	1.0000	E15 (1)
E16	1.0000	E16 (1)
E17	0.9987	E16 (0.699), E18 (0.301)
E18	1.0000	E18 (1)
E19	0.9896	E15 (0.425), E16 (0.362), E18 (0.213)

The application of the proposed S-T DEA approach yields for E19 the results which are graphically depicted in Figure 5. When the weight assigned to the time dimension (w_t) is larger than or equal to 0.37 (and correspondingly w_{sp} < 0.63), the only DMU in the reference set of E19, which is suggested as a benchmark, is E16. When the weight of the space dimension is larger than or equal to 0.63, the version which is closer in terms of lambda values, namely E15, is selected. It should be noted, that although E18 appeared in the reference set of E19 in the conventional application of DEA (as shown in Table 6), in the proposed S-T DEA approach it is not extracted as a solution because it would yield $\hat{\varphi} < 1$, violating constraint (10).

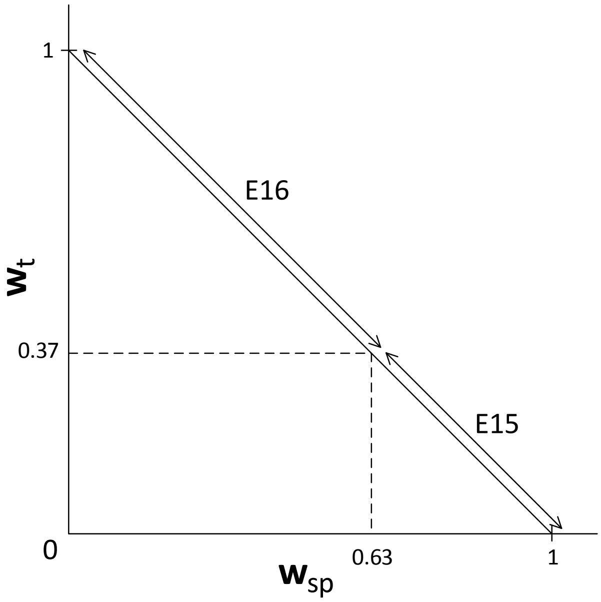

Figure 5. Selected peer according to w_{sp} and w_t of DMU E19 of software example

As already shown, the proposed S-T DEA approach is also capable of providing the efficiency scores and the projected values based on the selected weights. For example, for version E19 the efficiency scores are shown in Table 7, in relation to the DMU which is selected in the reference set. According to these results, E15 should be selected as a benchmark for comparison since its selection yields the largest efficiency for E19, even though the differences are small. The projected values indicate the extent by which the outputs should be improved when the corresponding software version is selected as a peer.

Table 7. Calculated efficiency scores and projected values for E19 in the spatiotemporal context based on the applied weights (software example)

	Weight of the temporal dimension	**Selected Peer**	$\hat{\varphi}$	**Technical Efficiency Score** $1/\hat{\varphi}$	**Projections***
Selected DMU in the reference set	$0 \leq w_t < 0.37$	E15	1.010	0.990	CBO: 15.86 →15.68 LCOM: 235.36 →228.31 DIT: 2.62→2.60
	$0.37 \leq w_t \leq 1$	E16	1.007	0.993	CBO: 15.86 →15.62 LCOM: 235.36 →233.64 DIT: 2.62→2.60

*Actual outputs have been inversed since the goal is to minimize the original metrics

8. Conclusions

Data Envelopment Analysis is one of the most powerful tools for measuring the relative efficiency of a set of units that enables benchmarking based on the corresponding reference set. However, the reference set is based only on the spatial distance of inefficient

DMUs from the efficient frontier. This property imposes a limitation when dealing with time series datasets, since it ignores, by construction, the temporal proximity among units. In this paper, we have proposed a novel two-stage DEA approach, which considers both the spatial and temporal dimension for the construction of the reference set when DMUs represent different snapshots in time of the same entity. The additional value of the proposed model lies in the ability to provide a single out of multiple DMUs as a benchmark in a unit's reference set and the recalculation of efficiency scores under this condition. The approach ensures that the obtained peer is: a) a single one, facilitating the process of comparison, b) a past version in the course of evolution, making the comparison feasible, and c) has the highest resemblance in terms of both space and time.

The proposed S-T DEA model can be applied in any context as it has been shown through the demonstration on a software development context. As an example, if different resources and outputs are available for successive time periods in the evolution of a company, the approach will yield for each inefficient period, a single past period that can be used for identifying opportunities for improvement. Since the proposed approach implies the presence of a spatiotemporal frontier, future research could investigate its relationship to the original frontier and interpret the projections of inefficient units onto the spatiotemporal one.

Acknowledgements

Konstantinos Petridis would like to acknowledge that part of this work was co-funded within the framework of the Action «State Scholarships Foundation's (IKY) mobility grants programme for the short term training in recognized scientific/research centers abroad for candidate doctoral or postdoctoral researchers in Greek universities or research» from the European Social Fund (ESF) programme "Lifelong Learning Programme 2007 – 2013".

Appendix A – Proof of Proposition 1

Let us assume that a DMU(3) is inefficient and λ_1, λ_2 are two non-zero lambdas of its reference set corresponding to DMU(1) and DMU(2) respectively, and assume that $\lambda_1 > \lambda_2$, such that:

$$\lambda_1 + \lambda_2 = 1 \quad \text{(A.1)}$$

$$0 \leq \lambda_1, \lambda_2 \leq 1 \quad \text{(A.2)}$$

According to the temporal dimension, DMU(3) is closer to DMU(2), while according to the spatial dimension DMU(3) has a higher resemblance to DMU(1).

The corresponding efficiency from (3) will be $\varphi \leq \frac{1}{y_3} \cdot (\lambda_1 \cdot y_1 + \lambda_2 \cdot y_2)$ (where y_3 is the output of the DMU under study) and due to (A.1), the above inequality will be reformulated as follows:

$$\varphi \leq \frac{1}{y_3} \cdot (\lambda_1 \cdot y_1 + \lambda_2 \cdot y_2) \Leftrightarrow$$

$$\varphi \leq \frac{1}{y_3} \cdot [\lambda_1 \cdot y_1 + (1-\lambda_1) \cdot y_2] \Leftrightarrow$$

$$\varphi \leq \frac{1}{y_3} \cdot [y_2 + (y_1 - y_2) \cdot \lambda_1] \quad \text{(A.3)}$$

The efficiency provided by the S-T DEA model will be the following:

$$\hat{\varphi} \leq \frac{y_1}{y_3}, \text{ if } w_t < w_{sp} \quad \text{(A.4)}$$

$$\hat{\varphi} \leq \frac{y_2}{y_3}, \text{ if } w_t > w_{sp} \quad \text{(A.5)}$$

Therefore, the efficiency $\hat{\varphi}$ is calculated from (9) by selecting the combination of y_1, y_2 that maximizes the value of $\hat{\varphi}$ satisfying the constraint. In the right hand side of inequalities (A.4) and (A.5), lambda values are omitted due to the constraints (9) and (11) of the S-T DEA model.

In order to prove that $\varphi \geq \hat{\varphi} \; \forall y$ the relative position of φ and $\hat{\varphi}$ by means of order relationships should be investigated. For this reason, two scenarios about the arrangement of y_1 and y_2 are examined:

- $y_1 < y_2$

In this case, $y_2 + (y_1 - y_2) \cdot \lambda_1 < y_2$ as $(y_1 - y_2) \cdot \lambda_1 < 0$ (for the sake of simplicity denominator y_3 is dropped from the analysis since it is equal to all instances). To investigate the order relation of $y_2 + (y_1 - y_2) \cdot \lambda_1$ with y_1, we reformulate the right hand side of inequality (A.3) with respect to y_1 as follows:

$$y_1 + (y_2 - y_1) \cdot \lambda_2 \quad \text{(A.6)}$$

From (A.6), it can be seen that as $(y_2 - y_1)\cdot\lambda_2 > 0$, then $y_1 + (y_2 - y_1)\cdot\lambda_2 > y_1$. Therefore, the following order relationship is formulated:

$$y_1 < y_1 + (y_2 - y_1)\cdot\lambda_2 = y_2 + (y_1 - y_2)\cdot\lambda_1 < y_2 \quad \text{(A.7)}$$

The initial assumption $y_1 < y_2$ is reformulated as $y_2 = y_1 + k$, where k is a real nonnegative number. By substitution in (A.3) the following is derived:

$$\varphi \le \frac{1}{y_3}\cdot\left[\lambda_1\cdot y_1 + \lambda_2\cdot(y_1 + k)\right] \Leftrightarrow$$

$$\varphi \le \frac{1}{y_3}\cdot\left[(\lambda_1 + \lambda_2)\cdot y_1 + \lambda_2\cdot k\right] \Leftrightarrow$$

$$\varphi \le \frac{1}{y_3}\cdot(y_1 + \lambda_2\cdot k) \quad \text{(A.8)}$$

Comparing the right hand sides of inequalities (A.4) and (A.8), it is obvious that $y_1 + \lambda_2\cdot k > y_1$, consequently, $\varphi \ge \hat{\varphi}$.

- $y_1 > y_2$

In this case $y_1 + (y_2 - y_1)\cdot\lambda_2 < y_1$ as $(y_2 - y_1)\cdot\lambda_2 < 0$. To investigate the order relationship with y_2, we reformulate (22) with respect to y_2 as follows:

$$y_2 + (y_1 - y_2)\cdot\lambda_1 \quad \text{(A.9)}$$

From (A.7), it can be seen that as $(y_2 - y_1)\cdot\lambda_2 > 0$, then $y_2 + (y_1 - y_2)\cdot\lambda_1 > y_2$. Therefore, the following order relationship is formulated:

$$y_2 < y_2 + (y_1 - y_2)\cdot\lambda_1 = y_1 + (y_2 - y_1)\cdot\lambda_2 < y_1 \quad \text{(A.10)}$$

The initial assumption $y_1 < y_2$ is reformulated as $y_1 = y_2 + m$, where m is a real nonnegative number. By substitution in (A.3) the following is derived:

$$\varphi \le \frac{1}{y_3}\cdot\left[\lambda_1\cdot(y_1 + m) + \lambda_2\cdot y_2\right] \Leftrightarrow$$

$$\varphi \le \frac{1}{y_3}\cdot\left[(\lambda_1 + \lambda_2)\cdot y_2 + \lambda_1\cdot m\right] \Leftrightarrow \quad \text{(A.11)}$$

$$\varphi \le \frac{1}{y_3}\cdot(y_2 + \lambda_1\cdot m)$$

Comparing the right hand sides of inequalities (A.5) and (A.11), it is obvious that $y_2 + \lambda_1\cdot m > y_2$, consequently, $\varphi \ge \hat{\varphi}$.

Appendix B – Proof of Proposition 2

Let $\lambda_i^{DEA} \in F^{DEA}$ be the optimal lambdas of DEA output model described by (1) – (4), $\lambda_i^{S-T\ DEA} \in F^{S-T\ DEA}$ be the optimal lambdas of S-T DEA model, described by (9) – (13) and $F^{\bullet}$ be the efficiency set of model □. By construction of tables **A** and **Δ**, a lambda that appears in F^{DEA} will also appear in $F^{S-T\ DEA}$. From the S-T DEA model, only one weight dependent solution will be selected among the lambdas provided by the initial DEA model (1) – (4) and thus the feasibility of the model is guaranteed and furthermore it stands that: $F^{S-T\ DEA} \subseteq F^{DEA}$.

Compliance with Ethical Standards

Disclosure of potential conflicts of interest

Funding: Action: «State Scholarships Foundation's (IKY) mobility grants programme for the short term training in recognized scientific/research centers abroad for candidate doctoral or postdoctoral researchers in Greek universities or research» from the European Social Fund (ESF) programme "Lifelong Learning Programme 2007 – 2013".

Conflict of Interest: The authors declare that they have no conflict of interest.

References

Andersen, P., & Petersen, N. C. (1993). A procedure for ranking efficient units in data envelopment analysis. *Management Science*, *39*(10), 1261–1264.

Anderson, T. R., & Inman, L. (2011). Resolving the issue of multiple optima in Technology Forecasting using Data Envelopment Analysis. In *Technology Management in the Energy Smart World (PICMET), 2011 Proceedings of PICMET'11:* (pp. 1–5). IEEE.

Avkiran, N. K. (2009a). Opening the black box of efficiency analysis: an illustration with UAE banks. *Omega*, *37*(4), 930–941.

Avkiran, N. K. (2009b). Removing the impact of environment with units-invariant efficient frontier analysis: An illustrative case study with intertemporal panel data. *Omega*, *37*(3), 535–544.

Avkiran, N. K., & Rowlands, T. (2008). How to better identify the true managerial performance: state of the art using DEA. *Omega*, *36*(2), 317–324.

Banker, R. D., Cooper, W. W., Seiford, L. M., & Zhu, J. (2011). Returns to scale in DEA. In W. W. Cooper, L. M. Seiford, & J. Zhu (Eds.), Handbook on Data Envelopment Analysis (pp. 41-70). New York: Springer. Bergendahl, G. (1998). DEA and benchmarks – an application to Nordic banks. Annals of Operations Research, 82(0), 233–250.

Brooke, A., Kendrick, D., Meeraus, A., & Raman, R. (2003). GAMS/CPLEX 9.0. User notes. GAMS Development Corp., Washington, DC.

Cao, Y., & Yang, Z. (2009). Using DEA window analysis and the Malmquist index to evaluate the operations of Canadian Schedule I banks. In *Service Operations, Logistics and Informatics, 2009. SOLI'09. IEEE/INFORMS International Conference on* (pp. 353–358). IEEE.

Charnes, A., Clark, C. T., Cooper, W. W., & Golany, B. (1984). A developmental study of data envelopment analysis in measuring the efficiency of maintenance units in the US air forces. *Annals of Operations Research*, *2*(1), 95–112.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, *2*(6), 429–444.

Charnes, A., Cooper, W. W., & Rhodes, E. (1981). Evaluating program and managerial efficiency: an application of data envelopment analysis to program follow through. *Management science*, *27*(6), 668–697.

Chen, P.-C., & Yu, M.-M. (2014). Total factor productivity growth and directions of technical change bias: evidence from 99 OECD and non-OECD countries. *Annals of Operations Research*, *214*(1), 143–165.

Chen, W.-C., & Johnson, A. L. (2010). The dynamics of performance space of Major League Baseball pitchers 1871–2006. *Annals of Operations Research*, *181*(1), 287–302.

Coelli, T. J., Prasada Rao, D. S., O' Donnell, C. J., & Battese, G. E. (2005). *An Introduction to Efficiency and Productivity Analysis*. Springer US.

Cook, W. D., & Seiford, L. M. (2009). Data envelopment analysis (DEA)–Thirty years on. *European Journal of Operational Research*, *192*(1), 1–17.

Cooper, W. W., Seiford, L. M., & Tone, K. (2007). *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. Springer.

Couto, C., Maffort, C., Garcia, R., & Valente, M. T. (2013). COMETS: a dataset for empirical research on software evolution using source code metrics and time series analysis. *SIGSOFT Softw. Eng. Notes*, *38*(1), 1–3.

Emrouznejad, A. (2003). An alternative DEA measure: a case of OECD countries. *Applied Economics Letters*, *10*(12), 779–782.

Emrouznejad, A., & Thanassoulis, E. (2005). A mathematical model for dynamic efficiency using data envelopment analysis. *Applied Mathematics and Computation*, *160*(2), 363–378.

Emrouznejad, A., & Thanassoulis, E. (2010). Measurement of productivity index with dynamic DEA. *International Journal of Operational Research*, *8*(2), 247–260.

Färe, R., Grosskopf, S., & Lovell, C. K. (1994). *Production frontiers*. Cambridge University Press.

Freed, N., & Glover, F. (1981). Simple but powerful goal programming models for discriminant problems. *European Journal of Operational Research*, *7*(1), 44–60.

Grifell-Tatjé, E., & Lovell, C. K. (1997). A DEA-based analysis of productivity change and intertemporal managerial performance. *Annals of Operations Research*, *73*(0), 177–189.

Hashimoto, A., & Kodama, M. (1997). Has livability of Japan gotten better for 1956–1990?: A DEA approach. *Social Indicators Research*, *40*(3), 359–373.

Inuiguchi, M., & Mizoshita, F. (2012). Qualitative and quantitative data envelopment analysis with interval data. *Annals of Operations Research*, *195*(1), 189–220.

Jain, S., Triantis, K. P., & Liu, S. (2011). Manufacturing performance measurement and target setting: A data envelopment analysis approach. *European Journal of Operational Research*, *214*(3), 616–626.

Lozano, S., & Villa, G. (2010). Gradual technical and scale efficiency improvement in DEA. *Annals of Operations Research*, *173*(1), 123–136.

Lynde, C., & Richmond, J. (1999). Productivity and efficiency in the UK: a time series application of DEA. *Economic Modelling*, *16*(1), 105–122.

Moreno, P., Lozano, S., & Gutiérrez, E. (2013). Dynamic performance analysis of U.S. wireline telecommunication companies. *Telecommunications Policy*, *37*(6–7), 469–482.

Movahedi, M. M., Saati, S., & Vahidi, A. R. (2007). Iranian railway efficiency (1971–2004): an application of DEA. *International Journal of Contemporary Mathematical Sciences*, *2*, 1569–1579.

Parnas, D. L. (1994). Software aging. In *Proceedings of the 16th international conference on Software engineering* (pp. 279–287). Los Alamitos, CA, USA: IEEE Computer Society Press.

Rosenthal, R. E. (2013). A GAMS tutorial. In GAMS Development Corporation (Eds.), GAMS: A user's guide (pp. 7-28). Washington, DC, USA.

Rutledge, R. W., Parsons, S., & Knaebel, R. (1995). Assessing hospital efficiency over time: an empirical application of data envelopment analysis. *Journal of Information Technology Management*, *6*, 13 –24.

Samoladas, I., Gousios, G., Spinellis, D., & Stamelos, I. (2008). The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation. In B. Russo, E. Damiani, S. Hissam, B. Lundell, & G. Succi (Eds.), *Open Source Development, Communities and Quality* (pp. 237–248). Springer US.

Seiford, L. M., & Zhu, J. (1998). On alternative optimal solutions in the estimation of returns to scale in DEA. *European Journal of Operational Research*, *108*(1), 149–152.