Document downloaded from:

http://hdl.handle.net/10251/166752

This paper must be cited as:

Guijarro, F.; Martínez-Gómez, M.; Visbal-Cadavid, D. (2020). A model for sector restructuring through genetic algorithm and inverse DEA. Expert Systems with Applications. 154:1-13. https://doi.org/10.1016/j.eswa.2020.113422

The final publication is available at

https://doi.org/10.1016/j.eswa.2020.113422

Copyright Elsevier

Additional Information

A model for sector restructuring through genetic algorithm and inverse DEA

Francisco Guijarro[a,*], Mónica Martínez-Gómez[b], Delimiro Visbal-Cadavid[c]

[a] *Research Institute for Pure and Applied Mathematics, Universitat Politècnica de València, 46022 Valencia, Spain*
[b] *Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, 46022 Valencia, Spain*
[c] *Facultad de Ingeniería, Universidad del Magdalena, 470004 Santa Marta, Colombia*

Abstract

The aim of this study is to devise a sector restructuring model in which all the decision making units (DMUs) satisfy a predefined global efficiency level. The proposal makes several realistic assumptions regarding the merging of DMUs under specific circumstances. The model computes the global efficiency target by giving preference to merging DMUs over saving inputs, hence considering that the affected stakeholders may be resistant to restructuring, and this resistance may have overall negative effects on the image and reputation of the companies and organizations. In addition, the number of constituents in the new entities can be limited by the decision maker after the restructuring process, so that the model also considers a constraint on cardinality. The proposal combines the inverse data envelopment analysis (InvDEA), which computes the merger's input savings, and the genetic algorithm (GA), which solves the combinatorial problem of identifying the merging units. The proposal is illustrated by two examples from banking and higher education.

Keywords: mergers, restructuring, inverse data envelopment analysis, genetic algorithm, cardinality constraint

---

*Corresponding author

Email addresses: fraguima@upvnet.upv.es (Francisco Guijarro), momargo@eio.upv.es (Mónica Martínez-Gómez), dvisbal@unimagdalena.edu.co (Delimiro Visbal-Cadavid)

1. Introduction

The business environment is often characterized by conditions that offer opportunities for synergies through mergers and acquisitions (M&As). Restructuring refers to the reorganization of the ownership, operational or other structures of a private company or public organization. The aim of restructuring is to make the organization more profitable and efficient, which can primarily be addressed by two approaches. The first is by merging two or more firms and combining their activities to create a new entity with the aim of improve global performance through the synergy of its constituents. The second alternative consists of splitting the firm into a larger number of new and independent entities (Amin et al., 2017b).

Mergers can lead the joint entity to improving its long-run productivity, saving money, freeing up resources or boosting profits (Amin et al., 2019) and can even lead to the receipt of explicit support from governments. For example, during the Japanese economic crisis that erupted in the early 1990s, the Japanese government encouraged the healthier banks to merge with financially distressed banks (Halkos et al., 2016). A similar situation occurred with Spanish banks after their financial crisis (García et al., 2010). Conversely, academics have reported that merging may imply a negative effect on employment, as shown by Beckmann and Forbes (2004); Gugler and Yurtoglu (2004); Kubo and Saito (2012).

The negative effect of mergers on employment generates distrust among employees and other stakeholders concerned by the economic and social implications of mergers. Furthermore, merging public institutions can be constrained

The following abbreviations are used in this manuscript: DMU: Decision Making Unit; DEA: Data Envelopment Analysis; InvDEA: Inverse Data Envelopment Analysis; GA: Genetic Algorithm; M&As: Mergers and Acquisitions; VRS: Variable Returns to Scale; BBC: Banker-Charnes-Cooper; CCR: Charnes-Cooper-Rhodes; GCC: Gulf Cooperation Council.

by government legislation on employees' rights. In some countries, reducing the public labour force is not permitted, hence limiting the theoretical economic benefits of restructuring on efficiency.

Data envelopment analysis (DEA) measures the relative efficiency score of Decision Making Units (DMUs) (Charnes et al., 1978; Banker et al., 1984). DEA is an analytical method which has been widely used for the merger performance evaluation of DMUs. As an example, Kohers et al. (2000) examine the influence of bank efficiencies on the market assessment of bank holding company mergers; Also in the banking sector Halkos and Tzeremes (2013) evaluate the operating efficiency gains of a potential bank merger or acquisition. Lozano and Villa (2010) use DEA as a pre-merger planning tool to estimate expected cost and profit efficiency gains. The proposed model explicitly considers the possibility of closing existing units, which is especially apt for in-market horizontal mergers according to the authors. More recently, Amin et al. (2017a) use an InvDEA model to analyze the impact of mergers on the efficient frontier, Amin and Al-Muharrami (2018) introduce a model to deal with mergers in DEA allowing for negative data, and Amin et al. (2017b) apply the inverse DEA (InvDEA) model in a firm restructuring context.

In contrast to the DEA approach, the InvDEA model determines the required inputs and outputs for a given efficiency target θ (Gattoufi et al., 2014; Amin et al., 2019). Ahuja and Orlin (2001) differentiate between optimisation problems and inverse optimisation problems: "A typical optimisation problem is a forward problem since it identifies the values of observable parameters (decision variables) given the values of the model parameters (cost coefficients, right-hand side vector, and the constraint matrix). An inverse optimisation problem consists of inferring the values of the model parameters (cost coefficients, right-hand side vector, and the constraint matrix) given the values of observable parameters (decision variables)". Emrouznejad et al. (2019) state that "unlike the standard DEA whose objective is to find the efficiency score, the InvDEA assumes the efficiency given and aims to find the levels of inputs and outputs that are required to realise the desired efficiency score." The inverse optimisation approach we use in this paper is aligned with the problem we are addressing.

As has been pointed out by Wei et al. (2000), "we have a given feasible solution which is not necessarily an optimal solution, and we wish to adjust these parameter values, inputs, and outputs, as little as possible so that the feasible solution becomes the optimal one under the adjusted parameter values". Gattoufi et al. (2014) was the first InvDEA merger application on a real data set. The authors illustrate the methodology by using the examples of 42 banking units in Gulf Corporation Council countries.

Most of the authors' practical examples are focused on merging two DMUs to generate a new entity. Restructuring a whole sector (all DMUs) when the minimum global degree of efficiency is required has not been covered in the literature. In this context, our aim was to contribute to filling this gap by introducing several realistic assumptions regarding the merging of DMUs under specific circumstances:

1. our proposal seeks a global improvement of efficiency thus all the resulting new entities must guarantee a minimum level of efficiency: the global efficiency target;
2. for non-efficient DMUs, the efficiency target θ is obtained by a) merging with other DMUs, b) reducing the inputs of the original DMU, or c) simultaneously considering both approaches;
3. whenever possible, the efficiency target should be achieved by merging DMUs instead of reducing inputs;
4. the procedure should consider cardinality constraints regarding the number of constituents of the new entities.

The cardinality constraint is used to model situations in which the decision maker wants to limit the number of constituents in the new entities, with political or social aspects in mind.

The third assumption refers to the adverse impact on the social image of companies and organisations that prioritise reducing labour costs during the restructuring process. However, governments and trade unions are often averse to downsizing, divestment and reallocation strategies, and this has overall negative effects on company reputations (Dentchev and Heene, 2004).

A final consideration is related to the concept of a major consolidation introduced by Amin et al. (2017a). Our proposal excludes major consolidations by assuming the abovementioned arguments of Amin et al. (2017a), hence promoting any mergers that do not affect the original efficiency frontier.

The following question then arises: what are the optimal combinations of DMUs that guarantee the minimum overall efficiency target, satisfying both the cardinality constraint and prioritisation? For example, if 30 DMUs were arranged in groups where the cardinality was limited to 2, the model would translate into 5.12e+17 potential solutions as shown in Section 2.4. The number of potential solutions makes the brute-force approach intractable in practice. We propose the use of a genetic algorithm (GA) model to deal with the search for units to be potentially merged. GA is then useful for the performance of an intelligent exploitation to direct the search into the region of better performance in solution space, avoiding the analysis of all possible solutions and thus reducing the search time. Hence, we used GA in our research to deal with the intractable problem of analyzing all potential solutions. The combinatorial problem of identifying the units to be merged is solved by a genetic algorithm (GA), so that the GA combinatorial search effectiveness is combined with the InvDEA model to obtain near-optimal solutions. The GA fitness function and how it deals with the aforementioned assumptions are discussed below. Despite the merge of units through DEA having been extensively addressed by the literature, the sector restructuring under the abovementioned realistic assumptions considered in our paper has not been covered. As stated by Amin et al. (2017a), "unlike other inverse optimization researches, there is a gap between theoretical developments and real world application of InvDEA". This paper also aims to fill this gap by addressing the realistic restructuring problem with two case studies.

The paper is organised as follows: Section 2 describes the InvDEA model, the GA model and describes how both models are combined to address the problem. In Section 3, two case studies from banking and higher education are given to illustrate the proposal. Finally, Section 4 summarizes the conclusions drawn from the study.

2. A combined inverse data envelopment analysis and genetic algorithm merging model

This section describes the basics of DEA (Section 2.1), the InvDEA model (Section 2.2), the GA model (Section 2.3) and how both are combined to solve the problem (Section 2.4). The latter describes the implemented fitness function and the mutation and crossover operators of GA optimization.

2.1. The DEA model

The first DEA model was proposed by Charnes et al. (1978). DEA was designed to evaluate the performance of a DMU in the presence of competitors, i.e. other DMUs. Subsequently, Banker et al. (1984) extended the model by assuming variable returns to scale (VRS), the so-called BBC Model (1). Suppose we have n DMUs and we have gathered information regarding m inputs, x_{ij} $(i = 1, \ldots, m)$, and s outputs, y_{rj} $(r = 1, \ldots, s)$. The input-oriented BCC Model 1 evaluates the efficiency of DMU_o $(o = 1 \ldots n)$, where θ_o is a scalar reporting the technical efficiency of DMU_o (Cooper et al., 2006).

According to Charnes et al. (1994) the BCC model "distinguishes between technical and scale inefficiencies by (i) estimating pure technical efficiency at the given scale of operation and (ii) identifying whether increasing decreasing, or constant returns to scale possibilities are present for further exploitation". We follow the approach of Gattoufi et al. (2014); Amin et al. (2017a,b); Amin and Al-Muharrami (2018) in the application of DEA, so that model 1 serves to determine which DMUs are (weak) efficient, $\theta = 1$. If Pareto-Koopmans efficiency was required, we should add slack variables to model 1 so that only those DMUs with $\theta = 1$ and zero value in slacks would be considered as Pareto-Koopmans efficient.

$$
\begin{array}{lll}
\min & \theta_o & \\
s.t. & & \\
& \sum_{j=1}^{n} x_{ij}\lambda_j - \theta_o x_{io} \leq 0 & i = 1 \ldots m \\
& \sum_{j=1}^{n} y_{rj}\lambda_j - y_{ro} \geq 0 & r = 1 \ldots s \\
& \sum_{j=1}^{n} \lambda_j = 1 & \\
& \lambda_j \geq 0 & j = 1 \ldots n
\end{array} \tag{1}
$$

2.2. The inverse DEA model

Wei et al. (2000) introduced the InvDEA model by discussing the following problem: if among a group of decision making units we increase certain inputs to a particular unit and assume that the DMU maintains its current efficiency level with respect to other units, how many more outputs could the unit produce? Later, Pendharkar (2002) and Amin and Emrouznejad (2007) extended the analysis of the original invDEA model by proposing inverse linear programming as an alternative to speeding up the computation of the Additive DEA model. Gattoufi et al. (2014) propose an InvDEA model for setting merger targets from both the input and output-oriented perspectives. Following the input-oriented approach, the model allows the merged entity to reach a given efficiency score, keeping all the outputs and maintaining the minimum input levels from each merging DMU. The input-oriented InvDEA model proposed by Gattoufi et al. (2014) is shown in Model 2:

$$
\begin{array}{lll}
\min & \sum_{i=1}^{m} (\alpha_{ik} + \alpha_{il}) & \\
s.t. & & \\
& \sum_{j\in F} x_{ij}\lambda_j + (\alpha_{ik} + \alpha_{il})\lambda_M - \theta(\alpha_{ik} + \alpha_{il}) \leq 0 & i = 1 \ldots m \\
& \sum_{j\in F} y_{rj}\lambda_j + (y_{rk} + y_{rl})\lambda_M \geq (y_{rk} + y_{rl}) & r = 1 \ldots s \\
& \sum_{j\in F} \lambda_j + \lambda_M = 1 & \\
& 0 \leq \alpha_{ij} \leq x_{ij} & j = k, l; i = 1 \ldots m \\
& \lambda_j \geq 0 & \forall j \in F \cup \{M\}
\end{array} \tag{2}
$$

where k and l refer to the DMUs to be merged and M to the resulting DMU; α_{ik} and α_{il} are the levels of the ith input from the merging DMU$_k$ and DMU$_l$, respectively, which is maintained by the new merged DMU$_M$; λ_j is the intensity variable, θ is the given efficiency target for the merged DMU$_M$, and F is the set of available peers in the post-merger evaluation process.

Model (2) is a nonlinear programming model, but can easily be transformed into the linear Model (3) by relaxing M from the set of its peers.

$$\begin{array}{lll} \min & \sum_{i=1}^{m}\left(\alpha_{ik}+\alpha_{il}\right) & \\ s.t. & & \\ & \sum_{j\in F} x_{ij}\lambda_j-\theta\left(\alpha_{ik}+\alpha_{il}\right)\leq 0 & i=1\ldots m \\ & \sum_{j\in F} y_{rj}\lambda_j \geq \left(y_{rk}+y_{rl}\right) & r=1\ldots s \\ & \sum_{j\in F}\lambda_j=1 & \\ & 0\leq\alpha_{ij}\leq x_{ij} & j=k,l; i=1\ldots m \\ & \lambda_j\geq 0 & \forall j\in F \end{array} \tag{3}$$

The output-oriented approach can be found in Gattoufi et al. (2014). In Section 3 we explain in detail why we opted for the input-orientation approach in our proposal.

2.3. Genetic algorithm

The GA is defined as an optimisation technique based on a heuristic search for solutions. GA models were originally proposed by Holland (1975) and are currently considered a subset of the evolutionary algorithms based on the natural evolutionary processes that enable species to adapt to their environment. Unlike other classic optimisation systems, GAs iterate by examining a set of possible solutions known as the population. These candidate solutions are encoded as strings or chromosomes. The chromosomes compete with each other for survival, but only the strongest can survive. Each iteration selects the best individuals in the current population to be part of the next population. The new generations inherit information from their parents and after a number of reproduced generations involving crossovers and mutations, the process eventually reaches convergence. The iterative process finishes when either all of the individuals in the population are essentially the same or the maximum number of iterations is reached.

Implementing the GA involves defining several concepts closely linked to the problem characteristics. Below, we describe in detail chromosome representation, the fitness function and the mutation and crossover operators.

2.3.1. The chromosome representation

The representation of each individual (chromosome) is called the genotype. Finding the appropriate genotype to represent an individual in the population is key to specifying the genetic algorithm. In the current study the genotype design aims to specify how DMUs are organized, i.e. to distinguish the DMUs that merge into new entities from those that remain unmerged.

Chromosomes are coded to contain a potential solution to the problem. In our approach, we represent chromosomes as integer strings. The length of these strings equals the number of DMUs to be merged. Each bit in the chromosome defines the group to which the DMU is assigned.

Figure 1 gives an example of chromosome representation, where 8 DMUs are arranged in 5 different mergers. As can be seen at the top of the figure, the decision process has merged DMUs 1, 3 and 8 in a new entity. Another merger contains DMUs 5 and 6, while DMUs 2, 4 and 7 remain unmerged. The genotype design of the merger example can be seen at the bottom. Each merged or unmerged DMU is assigned to a group in the vector representation, e.g. Group 1 is composed of DMUs 1, 3 and 8, DMUs 5 and 6 are joined in Group 4 and DMUs 2, 4 and 7 are labelled as Groups 2, 3 and 5, respectively.

2.3.2. The fitness function

In a given a merger configuration the fitness function computes the input savings that DMUs must accomplish to fulfil the predefined global efficiency target (Algorithm 1).

The first input (chromosome) of the algorithm contains the merging struc-

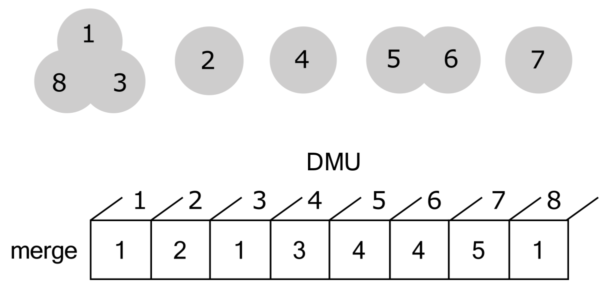

Figure 1: Example of chromosome representation.

ture of the DMUs as represented in Figure 1. The length of this vector coincides with the number of non-efficient DMUs, or those that can potentially be merged. The second parameter of the fitness function is the global efficiency target (θ) defined by the decision maker.

The algorithm iterates through the groups identified by the chromosome. The function *which* takes the chromosome as input and identifies those DMUs belonging to each group. These DMUs are saved in *merge* variable. The function iterates through each group and computes the corresponding inputs saving by using the *invDEA* function. The *invDEA* function computes the input savings to be achieved by both merged and unmerged DMUs in order to meet the efficiency target θ. The output is zero, indicating no savings when the efficiency score of the merged constituents is above the target, and thus there is no need to reduce the inputs. A zero is also obtained when dealing with an unmerged unit whose efficiency score is above the target.

As stated by Amin et al. (2017b), "identifying major consolidations in a market will help regulating and anti-trust authorities identifying those consolidations that potentially threaten the competitiveness in the market and hence thoroughly analyze those cases before any approbation. Moreover, business intelligence units in a firm may use what we propose to identify the possible threats in their business environment. One way of doing it is to use the scenario approach to identify, among all possible consolidations, those that are major in the market and hence represents the potential threat for the competitiveness of the firm." Considering that consolidations can potentially threaten competitiveness and in order to exclude major consolidation mergers, the computed input savings are penalized with a large positive penalty constant, *big*, if the new entity reaches the efficiency frontier. The *while* loop ends when the last group of DMUs has been computed. We use the *length* function to calculate the size of the *merge* variable. If length is positive, then there are DMUs in the group. A zero length indicates that the process has finished and all groups have been examined.

Once the *while* loop is finished, the algorithm returns the difference between the total input savings and the number of groups. We subtract the number of groups to get rid of unnecessary mergers, as happens when two or more DMUs are merged when all the individual efficiency scores are above the target. According to our assumptions, this undesirable behaviour is penalised by maximising the number of groups, thus promoting non-merging except where necessary.

2.3.3. The mutation operator

The mutation operator randomly selects a DMU to change its current merger. This is done by the *sample* function in Algorithm 2. Then, the DMU can be added to a pre-existing merger or isolated as an individual DMU. In the first case, the algorithm guarantees that the cardinality constraint is not violated; *i.e.* the DMU is not added to a saturated group regarding the maximum cardinality constraint. The mutation operator therefore generates feasible solutions and avoids any repair mechanism.

2.3.4. The crossover operator

The goal of the crossover operator is to obtain better chromosomes to improve the result by exchanging the information contained in the current good chromosomes (Zhang et al., 2011). The proposed crossover operator takes two parents and creates two children containing some of the genetic material from the parents. The function outlined in Algorithm 3 extracts those common merg-

Algorithm 1 Pseudo code for the fitness function.

Input: chromosome, $\theta \in [0, 1]$
Output: fit.value $\in \mathbb{R}$
Begin
inputs.saving = 0
i = 0
merge = *which*(chromosome == i+1)
n.merge = *length*(merge)
while n.merge $\geq$ 1 **do**
  inputs.saving = inputs.saving + *invDEA*(merge, θ) + big $\times$ *major*(merge)
  i = i+1
  merge = *which*(chromosome == i+1)
  n.merge = *length*(merge)
end while
return(inputs.saving - i)
End begin

ers pertaining to both parents. This common information is inherited by both children (Step 1 in Algorithm 3). The rest of the DMUs not involved in these groups are randomly merged, but observing the cardinality constraint (Step 2).

Figure 2 illustrates how the crossover operator works. Let us suppose we are dealing with 8 DMUs; the maximum merger cardinality is 2, and parents p_1 and p_2 represent possible solutions to the problem. Unlike other DMUs, we can see that DMU 4 is isolated in both parents, and DMUs 5 and 6 are joined in both parents. This common information is inherited by children c_1 and c_2. DMUs 1, 2, 3, 7 and 8 are merged randomly while observing the cardinality constraint. In this way, child c_1 joins DMUs 2 and 3, and leaves DMUs 1, 7 and 8 isolated. Child c_2 merges DMUs 1 and 2, and 3 and 7, while DMU 8 remains on its own.

In the following we explain Algorithm 3 in detail. We include as inputs the chromosomes of both parents, *parent1* and *parent2*, and the maximum cardinal-

Algorithm 2 Pseudo code for the mutation operator

```
Input: chromosome; max.card ∈ ℤ
Output: chromosome ∈ F^f
Begin
n.chromosome = length(chromosome)
i = sample(n.chromosome, 1)
feasible.merge = which.j(length(chromosome == j) ≤ max.card, j ≠ i)
chromosome[i] = sample(feasible.merge, 1)
return(chromosome)
End begin
```

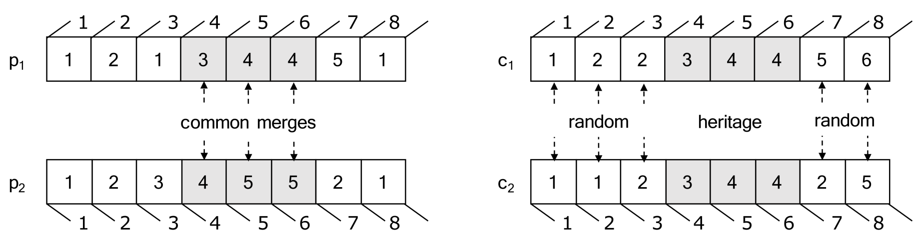

Figure 2: Example of crossover.

ity of the groups, *max.card*. The crossover operator returns the chromosomes of both children, *child1* and *child2*. The code is structured into two steps. Children inherit the common information from their parents in Step 1. The uncommon information is merged randomly, while observing the maximum number of groups *max.card* in Step 2.

Firstly, we initialise the variable *child1* with zeros. The function *common.groups* identifies which DMUs are sharing common groups in both parents. Regarding the example of Figure 2, *common.groups* would return DMUs in positions 4, 5 and 6 as an inherited merge. Then, this merge along with any other merge is inherited by *child1* and eventually transferred to *child2*. This way both children share the same common groups of DMUs.

Step 2 develops the merging of uncommon DMUs in parents. The *for* loop iterates for both children. First, we identify which positions have not been as-

Algorithm 3 Pseudo code for the crossover operator

```
Input: parent1, parent2; max.card ∈ ℤ
Output: child1, child2
Begin
# STEP 1: children inherit common groups in parents
child1 = rep(0, length(parent1))
common.merges = common.groups(parent1, parent2)
i = 0
merge = which(common.merges == i+1)
n.merge = length(merge)
while n.merge ≥ 1 do
    child1[merge] = i+1
    i = i+1
    merge = which(common.merges == i+1)
    n.merge = length(merge)
end while
child2 = child1
# STEP 2: uncommon elements are merged randomly but observing the
cardinality constraint
for child ∈ {child1 ∪ child2} do
    which.zero = which(child == 0)
    j = i
    while length(which.zero) ≥ 1 do
        merge = sample(which.zero,min(sample(max.card,1)),length(which.zero))
        child[merge] = j+1
        j = j+1
        which.zero = which(child == 0)
    end while
end for
return(child1, child2)
End begin
```

signed to any merge in Step 1. We save this information in variable *which.zero*. In the *while* loop, we use the *sample* function to randomly determine which DMUs are going to be assigned for the next merge, while observing the maximum cardinality for the mergers. Once those DMUs are assigned to a new group (child[merge] = j+1), then we search for the remaining unassigned DMUs (which.zero = *which*(child == 0)) and iterate until the stopping criterion is met. Once we have proceeded through both children, the crossover operator returns the chromosomes.

2.4. The InvDEA-GA model

A broad spectrum of papers has explored the benefits of combining data envelopment analysis with genetic algorithms in different areas. Some of the DEA-GA models proposed in the literature focus on stochastic scenarios by complementing the analysis of efficiency with the GA heuristic approach. For example, Kuah et al. (2012) evaluate the knowledge management performance in higher education. The accuracy of the efficiency scores is improved by proposing a framework which combines a Monte Carlo DEA version with GA. Udhayakumar et al. (2011) develop a stochastic simulation-based GA for solving chance constrained data envelopment analysis problems. In contrast to conventional models, which focus on deriving deterministic equivalents, the authors propose that the stochastic objective function and chance constraints be directly handled by the genetic process. Jain et al. (2015) introduce a GA-based approach to estimate weight restrictions in DEA, incorporating Decision Makers' preferences into weight restrictions. GA is used to find a set of weights which are at a minimum distance from all these preferences. Some recent developments using DEA and GA include: Lin et al. (2013); Hsu (2014); Kao et al. (2014); González et al. (2015); Fallahpour et al. (2016); Pendharkar (2018).

Despite its potential synergy, no framework integrating invDEA and GA has so far been proposed, to the best of our knowledge. This paper proposes a model combining both methodologies for a generic restructuring context when several realistic conditions have to be met. The aim is to find optimal or quasi-optimal merging solutions in a timely manner.

The number of potential solutions makes the brute-force approach intractable in practice. For example, let us suppose that the decision maker limits the cardinality of the new entities to 2. The number of possible solutions in this context is obtained through the expression $C(n) = C(n-1)+(n-1)\times C(n-2)$, where n represents the number of non-efficient DMUs involved in the analysis, $C(1) = 1$ and $C(2) = 2$. A system composed of 30 DMUs would translate into 5.12e+17 potential solutions. In addition, this figure can be dramatically increased if the cardinality constraint is relaxed. For example, if up to 3 entities are considered in the merging process, this would entail adding new potential solutions to those in the previous case, thus increasing the number of potential solutions. This is why we propose using a heuristic approach to address the restructuring problem. Due to its simplicity, GA is a popular alternative that suits the problem we are dealing with, but other approaches from Evolutionary algorithms may apply.

This is why we propose using a heuristic approach to address the restructuring problem. Due to its simplicity, GA is a popular alternative that fits the problem we are dealing with, but other approaches from Evolutionary algorithms may apply.

The GA is initialized by randomly selecting the initial population of chromosomes. The selection of the best individuals is performed by the fitness function proposed in Algorithm 1 to account for the input savings of each temporary solution. When two solutions reach the same input savings the algorithm gives preference to the case with the larger number of groups (less merging), according to the assumption given in the Introduction. The mutation operation generates offspring by randomly changing one merger, so that the mutation prevents local searches of the search space and increases the probability of finding global optima. The crossover operation generates offspring from two chosen individuals in the population (parents). The offspring inherit some characteristics from each parent. The specified maximum number of generations is considered as the termination condition.

3. Two practical applications

The efficiency of Banking and Higher Education has traditionally received considerable attention from research groups (Abbott and Doucouliagos, 2003; Nazarko and Šaparauskas, 2014; Wanke and Barros, 2014; Tsolas and Charles, 2015; Radojicic et al., 2018; Zhou et al., 2018).

This section illustrates the proposed model using two datasets as case studies. The first is composed of 46 banks in Gulf Cooperation Council (GCC) countries (Bahrain, Kuwait, Oman, Qatar, Saudi Arabia and United Arab Emirates). This dataset has previously been used by the invDEA literature (Gattoufi et al., 2014; Amin and Al-Muharrami, 2018; Amin et al., 2017a,b, 2019). However, we opted to include a second dataset to illustrate our proposal, so that future research can use both to compare different models and approaches.

This dataset includes information on 32 Colombian state universities and extends the information reported in Visbal-Cadavid et al. (2017). The two datasets, along with the technical efficiency scores under the VRS assumption are provided in Appendix A and Appendix B.

The invDEA model with the input-orientation approach is considered in both cases. We argue that reducing inputs makes more sense than increasing outputs in both the banking and higher education sectors. In the case of bank restructuring, improving the efficiency of the sector by gaining new customers is not a realistic assumption for the banking sector as a whole. In other words, in an output-oriented approach a bank can improve its efficiency only by attracting new customers from other banks. However, it is not advisable to improve overall efficiency by increasing the number of customers if we are dealing with restructuring the whole sector, and the same applies to higher education. In the Colombian educational system, increasing the number of students at state Universities may not be feasible due to budget constraints. Efficiency must be improved either by reducing the university inputs or by simply merging the DMUs as prioritised in the present proposal.

Table 1 shows the GA parameters used in both examples. These values were

Table 1: The parameters of the GA

Population size	Number of generations	Crossover ratio	Mutation ratio
150	1,000	0.3	0.5

determined by reviewing the literature on GA applications, although there is no specific rule about choosing the optimal parameters. Similar results have been obtained for other values not reported in this paper.

3.1. Banking mergers

The banking dataset is composed of 42 banks, 32 of which are non-efficient. Following Gattoufi et al. (2014), we used two inputs (Interest expenses, non-interest expenses) and two outputs (Interest income, non-interest income).

We considered 4 different values for the efficiency target in the banking dataset: 0.70, 0.75, 0.80 and 0.85 to study how banks merge according to the required level of efficiency. The second parameter is the maximum cardinality of the new entities. The simulation encompasses 3 scenarios: up to 2, 3 and 4 banks for each new entity.

We have run the algorithm 200 times to account for the standard deviation of the results. The results of one of these solutions are reported in Table 2. The DMU code and its efficiency score are shown in the first two columns. The other columns indicate the merging group assigned to each DMU for the considered scenarios. Shaded cells indicate that the DMU efficiency score is below the required efficiency target θ. A circled number indicates that the DMU has been merged, and a non-circled number that the DMU remains unmerged. For example, DMUs B002 and B027 are joined in Group 1 for $max.card = 2$ and $\theta = 0.70$, shown with a circle. DMU B004 belongs to Group 3, and is not circled. This means that DMU B004 has not been merged for $\theta = 0.70$. The merging summary is highlighted in the last rows in the table. The "Input Saving" row indicates the input reduction that DMUs must accomplish in order to meet the efficiency target requirement. A zero value indicates that global efficiency can be reached by merging DMUs and that no input reduction is needed in the new entities. For $max.card = 2$ and $\theta = 0.70$, the efficiency target is achieved by simply merging the DMUs, so that no input saving is reported. The case of $max.card = 2$ and $\theta = 0.85$ is the only one in which global efficiency cannot be reached by solely merging units. The "Alternative Solutions" row shows whether the model has returned a unique solution (No) or more than one (Yes). It can be seen that in cases where no input saving is reported, the GA finds alternative solutions. We reported only one of these solutions in each experiment. The remaining rows indicate how many groups have been designed according to the cardinality property.

The global efficiency target was reached by simply merging units in all but one case. This exemplifies the benefits of merging banks to improve the overall efficiency of the sector, excluding the input reduction imposition. The model promotes the merging of the DMUs whose efficiency is below the required level, while DMUs over the efficiency target are more likely to remain unmerged. Thus, even though Table 2 only shows one solution in the cases with multiple solutions, the decision maker can analyse the alternatives not reported in the table, and can select the specific solution that best fits any additional economic, political or social requirements.

Table 3 shows two cases in depth: $max.card = 2$ and $\theta = 0.85$, and $max.card = 4$ and $\theta = 0.80$. In the first case, the efficiency target is achieved both by merging banks and reducing inputs, while in the second case the target is reached by simply merging banks. Inter-bank synergy enables the new entities to reach the corresponding efficiency target. For example, DMUs 2 and 4 are merged in Group 1 for $max.card = 2$ and $\theta = 0.85$. DMU 2 and DMU 4 have an efficiency score of 0.677 and 0.892, respectively. The new merged entity is shown to have an efficiency level of 0.881. When both the cardinality and the efficiency target are relaxed ($max.card = 4$ and $\theta = 0.80$), the system merges DMUs whose efficiency score is furthest below the required level. For example, DMUs 2 and 12 have an efficiency score of 0.677 and 0.669, respectively. They are joined with DMU 38 (0.876) to reach an efficiency score of 0.853. Since non-merging is avoided unless really necessary, groups with a high cardinality are only designed in cases where the efficiency target cannot be obtained by forming smaller groups, as in the case of $max.card = 4$ and $\theta = 0.80$: no merger is reported with 4 DMUs.

Another interesting point in Table 3 refers to the characteristics of the unmerged units. In $max.card = 2$ and $\theta = 0.85$, the model leaves DMUs 7, 8, 25 and 29 unmerged due to the low input saving needed to achieve the efficiency target: 0.290, 4.355, 2.196 and 2.872, respectively.

These figures are very low in comparison with the input saving required for some new entities, so that the GA tends to sacrifice smaller DMUs for bigger ones in terms of inputs.

Table 4 shows the summary statistics of the input savings for the 200 simulations performed in our analysis. We have only run a case where parameters are $max.card = 2$ and $\theta = 0.85$, because in other cases the solution obtained was optimal (i.e. saving inputs were 0) -there was no dispersion in the results of the simulations-. We can see that saving ranges from 229.8 to 403.8, with a standard deviation of 37.8. Many solutions are located around 297.4, which is not far from the best solution found by the algorithm (229.8). The solution reported in Table 2 (318.613) is also close to the best solution in Table 4.

Table 2: Mergers for the banking dataset

max.card		2				3				4			
	θ	0.70	0.75	0.80	0.85	0.70	0.75	0.80	0.85	0.70	0.75	0.80	0.85
DMU	Efficiency score												
B002	0.677	1	1	1	1	1	1	1	1	1	1	1	1
B003	0.640	2	2	2	2	2	2	2	2	2	2	2	2
B004	0.892	3	2	1	1	3	3	3	1	3	3	3	3
B007	0.829	4	3	3	3	4	4	4	3	4	4	4	4
B008	0.738	5	4	4	4	5	5	5	4	5	3	5	5
B009	0.727	6	5	5	5	6	6	5	5	6	5	6	4
B011	0.939	7	6	2	6	7	1	6	6	7	6	7	4
B012	0.669	8	7	6	7	8	7	7	5	8	5	1	6
B013	0.970	9	4	7	8	9	8	8	7	9	7	8	7
B014	0.813	8	8	8	9	10	7	9	8	10	8	9	2
B015	0.953	10	9	5	10	11	6	10	8	11	9	10	2
B016	0.962	11	10	4	5	12	9	11	3	12	10	11	8
B017	0.784	2	1	8	11	13	10	9	9	13	11	10	9
B018	0.866	12	11	9	12	14	11	12	4	14	12	12	6

B019	0.770	13	12	10	13	15	12	1	10	15	13	12	1
B023	0.910	14	5	11	2	16	13	13	2	16	14	13	5
B025	0.756	15	13	9	14	17	14	11	6	17	15	11	3
B026	0.826	16	14	12	15	18	15	1	2	18	16	14	10
B027	0.678	1	15	13	9	1	16	14	9	1	6	3	1
B028	0.980	17	16	14	7	19	17	7	5	19	17	5	11
B029	0.687	7	16	15	16	18	3	2	4	19	18	2	4
B030	0.815	18	17	16	10	20	18	15	6	20	19	15	5
B033	0.838	19	18	17	8	21	19	16	7	21	20	16	5
B034	0.837	20	7	15	15	22	2	17	8	22	5	2	10
B035	0.871	21	19	13	11	2	20	2	9	8	1	6	9
B037	0.811	22	15	6	6	23	5	5	1	23	2	17	3
B038	0.876	23	20	10	13	24	21	14	10	2	21	1	1
B042	0.779	24	21	11	12	8	16	3	10	24	18	7	10
Input savings		0	0	0	318.613	0	0	0	0	0	0	0	0
Alternative solutions		Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Unmerged		20	14	8	4	22	14	9	0	20	15	8	3
Cardinality = 2		4	7	10	12	3	7	5	2	4	5	7	2

Cardinality = 3	-	-	-	-	0	0	3	8	0	1	2	3
Cardinality = 4	-	-	-	-	-	-	-	-	0	0	0	3

Note. The DMU code and its efficiency score are shown in the first two columns. The rest of the columns indicate the merging group assigned to each DMU. Shaded cells indicate that the efficiency score of the DMU is below the required efficiency target θ. A circled number indicates that the DMU has been merged, whilst a non-circled number indicates that the DMU remains unmerged.

Table 3: Two examples on merging banks and the corresponding efficiency scores and input savings obtained for the new entities

max.card = 2, $\theta = 0.85$				max.card = 4, $\theta = 0.80$			
Merge	DMUs	Inputs saving	Efficiency score	Merge	DMUs	Inputs saving	Efficiency score
1	{2,4}	0	0.881	1	{2,12,38}	0	0.853
2	{3,23}	0	0.906	2	{3,29,34}	0	0.827
3	{7}	0.290	0.829	3	{4,27}	0	0.883
4	{8}	4.355	0.738	4	{7}	0	0.829
5	{9,16}	1.461	0.823	5	{8,28}	0	0.878
6	{11,37}	0	0.930	6	{9,35}	0	0.871
7	{12,28}	0	0.887	7	{11,42}	0	0.936
8	{13,33}	0	0.891	8	{13}	0	0.970
9	{14,27}	112.587	0.794	9	{14}	0	0.813
10	{15,30}	0	0.948	10	{15,17}	0	0.864
11	{17,35}	0	0.867	11	{16,25}	0	0.843
12	{18,42}	54.601	0.835	12	{18,19}	0	0.865
13	{19,38}	0	0.879	13	{23}	0	0.910
14	{25}	2.196	0.756	14	{26}	0	0.826
15	{26,34}	140.251	0.842	15	{30}	0	0.815
16	{29}	2.872	0.687	16	{33}	0	0.838
-	-	-	-	17	{37}	0	0.838

Table 4: Summary statistics for the 200 simulations performed on the banks' case study

Parameters	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Sd. Dev.
max.card = 2 $\theta = 0.85$	229.8	297.4	297.4	324.0	356.6	403.8	37.8

3.2. Mergers in higher education

The higher education dataset is composed of 32 Colombian state universities, 20 of which are considered as non-efficient by the VRS model. We considered 3 inputs (Administrative Expenses, Full Time Equivalent Teachers, Research Staff) and 3 outputs (Number of Students, Employed Graduates, Research Papers) from the year 2018. It is worth mentioning that the Colombian Higher Education system computes some of these variables after making certain amendments. Students and teachers are measured in a different way by each University. The weighting takes into account the knowledge area, the level of training (professional, technical or technological) and the teaching methodology used (traditional learning or distance learning). For example, a student enrolled in the Open University weights 0.6 times a student attending in person. It also takes into account the development level of Colombian regions and the educational level of its population. The "Research Papers" variable is also measured through a weighting scheme. Under Colombian Law 1279, the weighted number of research papers is calculated according to the quality level of the journal.

As in the case of the banks, we have run the algorithm 200 times. The experiments were carried out using the same parameters as in the case of the banks (Cardinality and Efficiency Target) and the results for one of the experiments (which serves as example) are reported in Table 5. Although we found some scenarios in which a reduction of inputs was needed to reach the required efficiency level, a similar pattern of behaviour to the banks can be seen. As in the previous case, small universities are more likely to remain unmerged, despite the input reduction they need to reach the efficiency target. More interestingly, we can see that the GA returns the same solution as for experiments with $max.card = 3$ and $\theta = 0.85$, and $max.card = 4$ and $\theta = 0.85$. It can thus be inferred that once this solution is obtained, no cardinality relaxation can improve the solution. Any additional merger may involve a change in the efficiency frontier, which is not allowed in our model, following Amin et al. (2017a).

Finally, it should be noted that the input savings required in some cases is not as large as the level of total inputs and the efficiency gain. In the most conservative scenario ($max.card = 2$ and $\theta = 0.85$) the system needs to reduce 61,572.2 input units to reach the efficiency target. However, the total inputs for the 20 non-efficient DMUs is 641,987.4 so that the input saving represents 9.6% of the total in the most extreme scenario.

Table 7 shows the summary statistics of the input savings for the 200 simulations performed on 5 different scenarios. Results show that input savings can change because of the randomness of the GA algorithm. In order to reduce this uncertainty, the decision maker can increase the number of iterations or just to run several experiments -as we did- to search for the most accurate sub-optimal solution. However, we can observe that in all cases the median and the first quantile produce the same result. This gives an idea on how consistent the GA algorithm is as regards the solutions obtained.

Table 5: Mergers for the universities dataset

max.card		2				3				4			
	θ	0.70	0.75	0.80	0.85	0.70	0.75	0.80	0.85	0.70	0.75	0.80	0.85
DMU	Efficiency score												
Univ02	0.430	1	1	1	1	1	1	1	1	1	1	1	1
Univ03	0.682	2	2	2	2	2	2	2	2	2	2	2	2
Univ04	0.849	3	3	2	3	3	3	3	3	3	3	2	3
Univ05	0.833	4	4	3	4	4	4	4	4	4	4	3	4
Univ06	0.642	2	1	4	2	2	2	5	2	2	5	4	2
Univ07	0.803	5	5	5	1	5	5	5	3	5	6	5	3
Univ09	0.715	6	3	6	5	6	6	6	4	6	7	6	4
Univ10	0.913	7	6	7	6	7	7	7	5	7	5	7	5
Univ12	0.651	8	7	7	7	8	2	2	2	8	3	4	2
Univ13	0.617	7	6	1	8	1	1	7	5	1	1	1	5
Univ17	0.817	9	8	8	3	9	8	8	3	8	2	8	3
Univ18	0.909	10	9	9	9	10	9	6	6	9	8	6	6
Univ20	0.930	1	7	6	4	11	10	2	4	10	9	9	4
Univ21	0.796	11	10	10	6	12	11	7	6	11	10	9	6

Univ22	0.662	4	9	9	9	10	10	1	1	4	9	9	1
Univ23	0.834	8	2	10	10	8	6	5	5	12	7	4	5
Univ28	0.889	12	11	11	11	13	12	9	7	13	11	10	7
Univ29	0.995	13	12	12	12	14	13	10	8	14	12	11	8
Univ30	0.578	14	13	13	8	11	9	6	6	10	5	4	6
Univ31	0.687	14	13	4	10	1	1	1	1	1	1	1	1
Input savings		0	12,417.4	19,790.3	61,572.2	0	0	0	4,193.9	0	0	0	4,193.9
Alternative solutions		Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Unmerged		8	6	6	4	9	8	5	3	9	6	6	3
Cardinality = 2		6	7	7	8	4	3	0	1	4	4	2	1
Cardinality = 3		-	-	-	-	1	2	5	5	1	2	2	5
Cardinality = 4		-	-	-	-	-	-	-	-	0	0	1	0

Note. The DMU code and its efficiency score are shown in the first two columns. The rest of the columns indicate the merging group assigned to each DMU. Shaded cells indicate that the efficiency score of the DMU is below the required efficiency target θ. A circled number indicates that the DMU has been merged, whilst a non-circled number indicates that the DMU remains unmerged.

Table 6: Two examples on merging universities and the corresponding efficiency scores and input savings obtained for the new entities

max.card = 3, $\theta = 0.70$				max.card = 3, $\theta = 0.85$			
Merge	DMUs	Inputs saving	Efficiency score	Merge	DMUs	Inputs saving	Efficiency score
1	{2,13,31}	0	0.963	1	{2,22,31}	0	0.861
2	{3,6}	0	0.718	2	{3,6,12}	4,193.9	0.797
3	{4}	0	0.849	3	{4,7,17}	0	0.890
4	{5}	0	0.833	4	{5,9,20}	0	0.865
5	{7}	0	0.803	5	{10,13,23}	0	0.857
6	{9}	0	0.715	6	{18,21,30}	0	0.893
7	{10}	0	0.913	7	{28}	0	0.889
8	{12,23}	0	0.733	8	{29}	0	0.995
9	{17}	0	0.817				
10	{18,22}	0	0.860				
11	{20,30}	0	0.773				
12	{21}	0	0.796				
13	{28}	0	0.889				
14	{29}	0	0.995				

Table 7: Summary statistics for the 200 simulations performed on the universities' case study

Parameters	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Sd. Dev.
max.card = 2, $\theta = 0.75$	12,417	13,034	13,034	13,955	13,888	17,442	1,589.6
max.card = 2, $\theta = 0.80$	19,790	19,790	19,790	21,106	20,633	26,882	2,358.8
max.card = 2, $\theta = 0.85$	60,439	61,641	61,641	65,146	66,733	76,388	5,395.2
max.card = 3, $\theta = 0.85$	4,194	10,451	10,451	14,037	17,935	25,407	5,103.9

max.card = 4 $\theta = 0.85$	3,050	9,196	9,196	13,511	19,004	24,888	5,901.0

4. Concluding remarks

Inverse data envelopment analysis (InvDEA) aims to change the level of inputs and/or outputs of a decision making unit (DMU) to obtain a predefined efficiency target. In a merger and acquisition context, InvDEA has been applied to two or more DMUs in order to find the required levels of inputs and outputs needed from the merging entities. Unlike other optimisation models, there is a gap between theoretical developments and realistic InvDEA applications. Although the literature offers some real-world examples in which two units are merged with the aim of improving the efficiency score, there is no real-world application where InvDEA has been proposed to improve the efficiency of an entire economic sector.

This paper proposes a model that combines the InvDEA model with a genetic algorithm to deal with sector restructuring. The proposal forces all the resulting units to reach a minimum predefined efficiency level, which can be achieved by reducing the input consumption of the original DMUs and/or by merging some original DMUs into a new entity.

The proposal considers some realistic assumptions. First, we constrain the cardinality of the new entities and, secondly, we prioritise merging DMUs rather than reducing their inputs. As the former assumption may involve a very large solution space, in which it is unrealistic to expect to find the optimal solution by a brute-force approach, a genetic algorithm is proposed to solve this problem. Regarding the latter assumption, reducing employment and public services in massive restructuring processes can have a negative impact on corporate image and public organisations. Our model gives preference to solutions in which global efficiency is improved by merging DMUs with each other instead of simply reducing the input level.

We applied the proposal to two real world datasets: the first was composed of 42 banks and the second included information from 32 Colombian state universities. In most experiments global efficiency was improved by simply merging units. We also found that the genetic algorithm finds multiple solutions in many cases, which can supply the decision maker with various alternatives. In the case of universities, he can merge universities according to their degree of proximity or similar syllabi. The experiment carried out on the universities is especially interesting due to the appearance of mergers in some of the experiments.

Our proposal can help political representatives in some decision making processes. They can decide to merge some inefficient universities with outstanding ones, improving the overall efficiency but limiting negative effects such as redundancies for workers and budget cuts. In the case of Colombia, when many Universities receive funds from the Government, this would translate into a more homogeneous development of higher education progress, thus improving both professional skills and job opportunities for people living in depressed areas. By constraining mergers to minor consolidations, we prevent consolidations that potentially threaten competitiveness in the market. In some cases, when the solution involves no inputs reduction, the decision maker can handle different scenarios. The possibility of merging universities without worsening the solution allows for further discussion between those who are responsible for the restructuring process, and maybe the consideration of additional variables before making the final decision.

This approach can be extended to other variations in the fitness function of the genetic algorithm, e.g. the minimum global efficiency target can be substituted by the mean efficiency target. Another option is to maintain the minimum level of efficiency for the whole sector while minimizing the dispersion of the efficiency scores. Alternative research directions could include extending an epsilon-based approach (Toloo, 2014a,b), and considering robust optimization in those situations where variables can be subjected to uncertainty (Toloo and Mensah, 2019).

Acknowledgement

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions that have contributed to improve this paper.

Appendix A.

Table A.8: GCC banks data and efficiency scores under the VRS assumption (Gattoufi et al., 2014)

Bank	Interest expenses	Non-interest expenses	Interest incomes	Non-interest incomes	Technical efficiency scores under VRS
B001	3,956.796054	1,894.4259	9,001.0036	8,701.496886	1
B002	481.2388026	319.9764807	974.8543974	597.7262586	0.677
B003	305.2	138.6	479.8	252.2	0.640
B004	4,710.680232	3,996.258941	12,920.33718	6,060.767712	0.892
B005	1.0179	1.2818	3.0537	0.377	1
B006	954.4368435	1,208.703319	1,991.004009	7,278.09659	1
B007	3.9653867	5.0818548	13.3591183	3.0029142	0.829
B008	14.629582	16.8625182	44.658724	14.9375732	0.738
B009	11.7710586	6.5788122	22.9520892	15.1342182	0.727
B010	364.9204497	244.7502714	923.5096577	1,942.934962	1
B011	4,897.442334	2,787.180598	11,294.60684	9,363.231698	0.939
B012	14.6653	8.9726	28.1242	10.9707	0.669
B013	6.0772884	14.2491762	26.993781	10.2074844	0.970
B014	397.6273178	371.5353219	894.8452115	1,902.878236	0.813
B015	661.1197271	830.1664611	2,325.127578	1,748.531218	0.953
B016	12.1250754	7.3458486	33.5725932	19.5299268	0.962
B017	1,222.026218	1,049.479174	2,959.509429	2,651.545717	0.784
B018	931.1716014	838.3456599	2,460.797508	2,765.48501	0.866
B019	4,070.35136	2,845.497525	8,377.368148	7,726.905715	0.770
B020	3,721.233105	858.4634144	6,953.700654	2,779.716296	1
B021	16.1372658	7.080336	40.7709348	22.12605	1
B022	150.7056462	132.5044812	538.754484	129.9563181	1
B023	3,857.940464	2,894.37408	7,439.526268	10,239.08718	0.910
B024	7,994.80804	2,286.908317	14,156.194	11,261.81992	1
B025	9.6889	6.9745	22.4315	6.032	0.756
B026	3,292.736384	1,953.592256	7,041.163964	3,323.973281	0.826

Table A.8 continued from previous page

Bank	Interest expenses	Non-interest expenses	Interest incomes	Non-interest incomes	Technical efficiency scores under VRS
B027	402.7722184	321.1887946	906.2374914	775.7775119	0.678
B028	32.8350582	21.536022	97.6791354	26.55126	0.980
B029	6.7373075	7.8537756	18.4024742	4.5043713	0.687
B030	531.3947334	922.0396861	1,672.092695	1,185.164603	0.815
B031	152.5095535	190.3613222	685.3742585	769.8976255	1
B032	1.924945	4.5813691	9.1627382	5.2743493	1
B033	4.8893603	6.7373075	17.4015028	5.0818548	0.838
B034	3,233.618974	2,527.413772	7,959.733478	4,684.615848	0.837
B035	5,169.709976	5,405.975285	15,189.60922	9,830.136952	0.871
B036	6,802.565778	5,608.863431	19,958.0432	15,716.89339	1
B037	3,111.951641	2,126.012757	6,895.571804	4,869.315511	0.811
B038	3,600.983329	1,319.710512	6,547.924278	5,116.081501	0.876
B039	7,781.754225	8,486.424885	27,514.03279	14,335.67889	1
B040	4,488.665847	4,531.418617	12,157.91278	12,380.67722	1
B041	3,188.735893	1,106.153629	5,727.009354	6,194.460322	1
B042	650.8299259	307.9590502	1,265.645548	441.3589729	0.779

Appendix B.

Table B.9: State Colombian Universities data and efficiency scores under the VRS assumption (Visbal-Cadavid et al., 2017)

Uni	Admin expenses	FTE teachers	Research staff	Number of students	Employed graduates	Research papers	TE scores under VRS
U01	426,514	2,754.5	2,547	73,312.3	65,812.4	11,277,250	1
U02	13,939	608.5	135	12,888.5	11,139.5	137,550	0.430
U03	29,439	1,325.25	175	36,317.7	28,327.8	554,800	0.682
U04	52,514	862.5	144	21,816.3	16,892.4	647,300	0.849
U05	42,950	649.5	210	24,152.5	14,460.1	771,750	0.833
U06	14,295	695	198	18,621.3	14,995.73	498,100	0.642
U07	24,261	454.4	140	16,402.7	12,984.4	470,250	0.803
U08	11,192	395.5	73	15,355.5	9,697.0	373,200	1
U09	12,186	360.5	37	11,215.5	9,575.8	75,100	0.715
U10	16,127	744.25	126	20,966.7	14,427.2	618,700	0.913

Table B.9 continued from previous page

Uni	Admin expenses	FTE teachers	Research staff	Number of students	Employed graduates	Research papers	TE scores under VRS
U11	17	438.5	18	11,411.1	8,778.5	108,750	1
U12	13,568	284	49	6,629	4,027.1	123,400	0.651
U13	7,325	616.75	42	16,268.7	12,043.7	60,350	0.617
U14	8,698	330	17	5,375.5	4,319.7	190,100	1
U15	3,872	146.25	4	2,974.8	2,152.0	8,750	1
U16	91,401	3,251.5	1,354	46,377	21,486.5	6,967,200	1
U17	87,456	657	120	21,876.2	18,185.7	430,950	0.817
U18	107,499	1,227.5	716	36,249.6	28,771.3	3,013,500	0.909
U19	34,116	855.75	454	23,863	19,140.5	2,250,100	1
U20	58,687	609.5	241	18,870.7	16,636.4	1,060,000	0.930
U21	46,389	406.5	106	13,909.1	10,895.0	360,650	0.796
U22	32,805	662.25	166	22,751.2	13,641.6	417,500	0.662
U23	17,331	626.75	113	21,280.2	16,802.8	430,800	0.834
U24	9,174	308.5	54	20,762.4	17,002.3	55,250	1
U25	5,755	143.75	14	7,811.4	6,067.9	6,800	1
U26	9,864	1,146	94	28,848.6	24,680.0	362,350	1
U27	7,685	441.5	107	23,787.6	21,477.8	356,350	1
U28	15,660	701.25	15	15,812.7	12,569.5	26,900	0.889
U29	4,641	255.25	42	6,742.8	4,649.8	147,500	0.995
U30	10,661	579	79	16,615.8	13,884.2	45,150	0.578
U31	17,872	975.75	227	30,854.5	24,902.7	583,350	0.687
U32	17,513	1,291.5	44	79,302.8	48,200.2	68,300	1

References

References

Abbott, M., Doucouliagos, C., 2003. The efficiency of australian universities: a data envelopment analysis. Economics of Education review 22, 89–97.

Ahuja, R.K., Orlin, J.B., 2001. Inverse optimization. Operations Research 49, 771–783.

Amin, G.R., Al-Muharrami, S., 2018. A new inverse data envelopment analysis model for mergers with negative data. IMA Journal of Management Mathematics 29, 137–149.

Amin, G.R., Al-Muharrami, S., Toloo, M., 2019. A combined goal programming and inverse dea method for target setting in mergers. Expert Systems with Applications 115, 412–417.

Amin, G.R., Emrouznejad, A., 2007. Inverse linear programming in dea. International Journal of Operations Research 4, 105–109.

Amin, G.R., Emrouznejad, A., Gattoufi, S., 2017a. Minor and major consolidations in inverse dea: Definition and determination. Computers & Industrial Engineering 103, 193–200.

Amin, G.R., Emrouznejad, A., Gattoufi, S., 2017b. Modelling generalized firms' restructuring using inverse dea. Journal of Productivity Analysis 48, 51–61.

Banker, R.D., Charnes, A., Cooper, W.W., 1984. Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science 30, 1078–1092.

Beckmann, T., Forbes, W., 2004. An examination of takeovers, job loss and the wage decline within uk industry. European Financial Management 10, 141–165.

Charnes, A., Cooper, W.W., Lewin, A.Y., Seiford, L.M., 1994. Basic dea models, in: Data envelopment analysis: Theory, methodology, and applications. Springer, pp. 23–47.

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

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

Dentchev, N.A., Heene, A., 2004. Managing the reputation of restructuring corporations: Send the right signal to the right stakeholder. Journal of Public Affairs: An International Journal 4, 56–72.

Emrouznejad, A., Yang, G.l., Amin, G.R., 2019. A novel inverse dea model with application to allocate the co2 emissions quota to different regions in chinese manufacturing industries. Journal of the Operational Research Society 70, 1079–1090.

Fallahpour, A., Olugu, E.U., Musa, S.N., Khezrimotlagh, D., Wong, K.Y., 2016. An integrated model for green supplier selection under fuzzy environment: application of data envelopment analysis and genetic programming approach. Neural Computing and Applications 27, 707–725.

García, F., Guijarro, F., Moya, I., 2010. Ranking spanish savings banks: A multicriteria approach. Mathematical and computer modelling 52, 1058–1065.

Gattoufi, S., Amin, G.R., Emrouznejad, A., 2014. A new inverse dea method for merging banks. IMA Journal of Management Mathematics 25, 73–87.

González, M., López-Espín, J.J., Aparicio, J., Giménez, D., Pastor, J.T., 2015. Using genetic algorithms for maximizing technical efficiency in data envelopment analysis. Procedia Computer Science 51, 374–383.

Gugler, K., Yurtoglu, B.B., 2004. The effects of mergers on company employment in the usa and europe. International Journal of Industrial Organization 22, 481–502.

Halkos, G.E., Matousek, R., Tzeremes, N.G., 2016. Pre-evaluating technical efficiency gains from possible mergers and acquisitions: evidence from japanese regional banks. Review of Quantitative Finance and Accounting 46, 47–77.

Halkos, G.E., Tzeremes, N.G., 2013. Estimating the degree of operating efficiency gains from a potential bank merger and acquisition: A dea bootstrapped approach. Journal of Banking & Finance 37, 1658–1668.

Holland, J.H., 1975. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press.

Hsu, C.M., 2014. An integrated portfolio optimisation procedure based on data envelopment analysis, artificial bee colony algorithm and genetic programming. International Journal of Systems Science 45, 2645–2664.

Jain, V., Kumar, A., Kumar, S., Chandra, C., 2015. Weight restrictions in data envelopment analysis: a comprehensive genetic algorithm based approach for incorporating value judgments. Expert Systems with Applications 42, 1503–1512.

Kao, H.Y., Chan, C.Y., Wu, D.J., 2014. A multi-objective programming method for solving network dea. Applied Soft Computing 24, 406–413.

Kohers, T., Huang, M.H., Kohers, N., 2000. Market perception of efficiency in bank holding company mergers: the roles of the dea and sfa models in capturing merger potential. Review of Financial Economics 9, 101–120.

Kuah, C.T., Wong, K.Y., Wong, W.P., 2012. Monte carlo data envelopment analysis with genetic algorithm for knowledge management performance measurement. Expert Systems with Applications 39, 9348–9358.

Kubo, K., Saito, T., 2012. The effect of mergers on employment and wages: Evidence from japan. Journal of the Japanese and International Economies 26, 263–284.

Lin, R.C., Sir, M.Y., Pasupathy, K.S., 2013. Multi-objective simulation optimization using data envelopment analysis and genetic algorithm: Specific application to determining optimal resource levels in surgical services. Omega 41, 881–892.

Lozano, S., Villa, G., 2010. Dea-based pre-merger planning tool. Journal of the Operational Research Society 61, 1485–1497.

Nazarko, J., Šaparauskas, J., 2014. Application of dea method in efficiency evaluation of public higher education institutions. Technological and Economic development of Economy 20, 25–44.

Pendharkar, P.C., 2002. A potential use of data envelopment analysis for the inverse classification problem. Omega 30, 243–248.

Pendharkar, P.C., 2018. A hybrid genetic algorithm and dea approach for multi-criteria fixed cost allocation. Soft Computing 22, 7315–7324.

Radojicic, M., Savic, G., Jeremic, V., 2018. Measuring the efficiency of banks: the bootstrapped i-distance gar dea approach. Technological and Economic Development of Economy 24, 1581–1605.

Toloo, M., 2014a. An epsilon-free approach for finding the most efficient unit in dea. Applied Mathematical Modelling 38, 3182–3192.

Toloo, M., 2014b. The role of non-archimedean epsilon in finding the most efficient unit: With an application of professional tennis players. Applied Mathematical Modelling 38, 5334–5346.

Toloo, M., Mensah, E.K., 2019. Robust optimization with nonnegative decision variables: a dea approach. Computers & Industrial Engineering 127, 313–325.

Tsolas, I.E., Charles, V., 2015. Incorporating risk into bank efficiency: A satisficing dea approach to assess the greek banking crisis. Expert Systems with Applications 42, 3491–3500.

Udhayakumar, A., Charles, V., Kumar, M., 2011. Stochastic simulation based genetic algorithm for chance constrained data envelopment analysis problems. Omega 39, 387–397.

Visbal-Cadavid, D., Martínez-Gómez, M., Guijarro, F., 2017. Assessing the efficiency of public universities through dea. a case study. Sustainability 9, 1416.

Wanke, P., Barros, C., 2014. Two-stage dea: An application to major brazilian banks. Expert systems with applications 41, 2337–2344.

Wei, Q., Zhang, J., Zhang, X., 2000. An inverse dea model for inputs/outputs estimate. European Journal of Operational Research 121, 151–163.

Zhang, G., Gao, L., Shi, Y., 2011. An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications 38, 3563–3573.

Zhou, X., Xu, Z., Chai, J., Yao, L., Wang, S., Lev, B., 2018. Efficiency evaluation for banking systems under uncertainty: A multi-period three-stage dea model. Omega .