Omega 127 (2024) 103084

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Short communication

The decomposition of efficiency in parallel network production models ☆

Antonio Peyrache [a,*], Maria C.A. Silva [b]

[a] CEPA, School of Economics, The University of Queensland, Australia
[b] CEGE - Católica Porto Business School, Rua Diogo Botelho, 1327, 4169-005 Porto, Portugal

ARTICLE INFO

Keywords:
Data envelopment analysis
Free disposal hull
Networks
Directional distance function
Inefficiency

ABSTRACT

Kao (2012) proposed a method to decompose DMU efficiency into sub-unit efficiencies for parallel production systems. We provide a numerical example showing that the proposed method can yield negative sub-unit efficiency scores under variable returns to scale, against common sense and standard postulates requiring this score to be non-negative. As a solution, we propose a decomposition based on the directional distance function that does not suffer from this problem. In particular, we recognize that the overall inefficiency of the DMU is composed of the sub-units technical inefficiencies and a reallocation inefficiency component. The proposed method can be also applied to non-convex technologies, therefore providing a more general method to implement such a decomposition. Given the connection between the directional distance function and slack-based efficiency measurement, the method can easily be extended to this case as well.

1. Introduction

In a recent series of papers Professor Chiang Kao proposed a method to assess the efficiency of DMUs that are composed by a parallel network structure, and decompose this efficiency into sub-unit efficiencies (see [1–3] and see also [4–6] for extensions). In this note we shall refer to the formulas as proposed in [2] for the sake of clarity and simplicity. The method has the significant advantage of measuring the overall efficiency of the DMU by taking into account potential inefficiencies arising from mis-allocation of resources across the various nodes of the network, thus improving on models that do not account for these effects (in the tradition of Fare et al. [7]). On the other hand, the method for decomposing efficiency into sub-units inefficiency under variable returns to scale (VRS) technologies fails the basic postulate that efficiency scores should be non-negative. Moreover, since the method is based on dual shadow pricing of inputs and outputs, it is not applicable to non-convex technologies, such as the free disposal hull (FDH). Below we provide a numerical counter-example that returns a sub-unit efficiency score that is negative. This invalidates the Kao's approach to efficiency decomposition under VRS. In a recent paper (see [8]) we proposed a method that can retain the advantages of the Kao model, without being affected by negative efficiency scores at the sub-unit level. An account of the historical development of these models can be found in [9]. The method has also the advantage of distinguishing between sub-unit technical inefficiencies and reallocation inefficiencies, as well as being applicable to non-convex technologies. We provide a brief account of this alternative decomposition using the numerical example and show that all efficiency scores are indeed non-negative. Given the connection between the directional distance function and slack-based efficiency measurement, the method can easily be extended to this case as well.

2. Kao's approach

Consider an industry (or system or network) composed of a group of decision making units (DMUs) $j = 1, \dots, J$ and the production process components (or sub-DMUs) within each DMU $p = 1, \dots, P$. Production processes use $i = 1, \dots, I$ inputs to produce $r = 1, \dots, R$ outputs. The quantity of input i of sub-unit p in unit j is denoted by x_{ij}^p, and the quantity of output r of sub-unit p in unit j is denoted by y_{rj}^p. The overall quantity of input i available to DMU j is indicated with a capital letter and is equal to the sum across processes: $X_{ij} = \sum_{p=1}^{P} x_{ij}^p$. Similarly, the overall quantity of output r produced by DMU j is $Y_{rj} = \sum_{p=1}^{P} y_{rj}^p$. We assume that the dataset satisfies weak essentiality of inputs: $\sum_i x_{ij}^p > 0,\ \forall j, p$. This is the only requirement on the data, and it states that at least one input must be strictly positive for each process in each DMU.

In Fig. 1 we provide a graphical representation of the parallel network, where inputs and outputs are collected in vectors: $\mathbf{x}_j^p$ is the vector of inputs used in DMU j by process p; $\mathbf{y}_j^p$ is the vector of outputs produced by DMU j using process p ($\mathbf{X}_j = \sum_p \mathbf{x}_j^p$ and $\mathbf{Y}_j = \sum_p \mathbf{y}_j^p$).

Kao [2] proposed the following model to assess the efficiency of DMU_o under constant returns to scale (CRS) of the underlying process technologies (after eliminating redundant constraints and slack

☆ Area: Data-Driven Analytics. This manuscript was processed by Associate Editor Chiang Kao.

* Corresponding author.
E-mail addresses: a.peyrache@uq.edu.au (A. Peyrache), csilva@porto.ucp.pt (M.C.A. Silva).

https://doi.org/10.1016/j.omega.2024.103084
Received 21 June 2023; Accepted 28 March 2024
Available online 6 April 2024

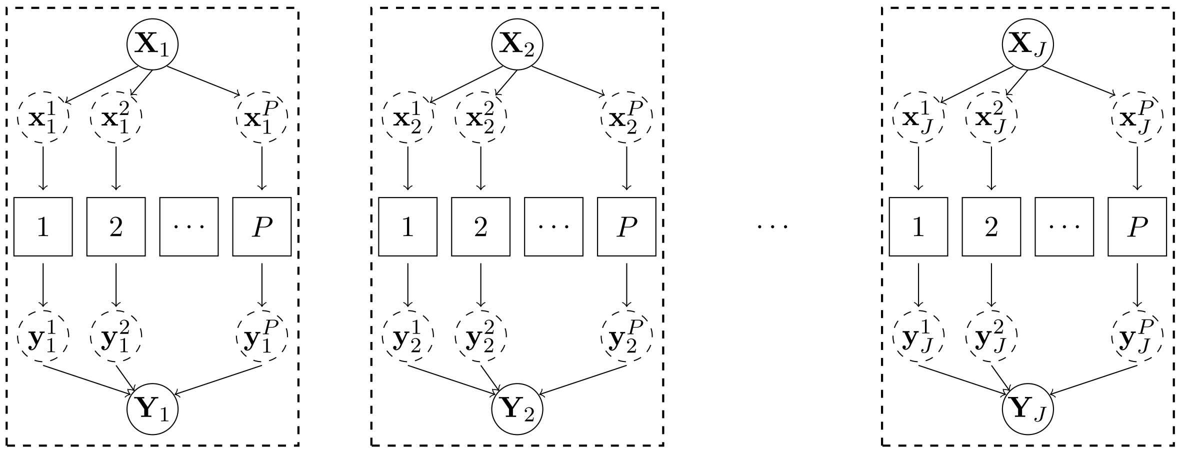

Fig. 1. Graphical representation of a parallel network system.

variables):

$$\begin{array}{ll} \min\limits_{\lambda_j^p,\theta} & \theta \\ s.t. & \sum_p \sum_j \lambda_j^p x_{ij}^p \leq \theta X_{io} \quad \forall i \\ & \sum_p \sum_j \lambda_j^p y_{rj}^p \geq Y_{ro} \quad \forall r \\ & \lambda_j^p, \theta \geq 0 \end{array} \tag{1}$$

where $X_{io} = \sum_p x_{io}^p$ is the total sum of input i available to DMU o and $Y_{ro} = \sum_p y_{ro}^p$ is the total sum of output r produced by DMU o. This model is also discussed in [1–4] as the parallel network model. The dual of this program returns the multiplier model which is here reported for completeness:

$$\begin{array}{ll} \max\limits_{u_r,v_i} & \sum_r u_r Y_{ro} \\ s.t. & \sum_r u_r y_{rj}^p - \sum_i v_i x_{ij}^p \leq 0 \quad \forall p,j \\ & \sum_i v_i X_{io} = 1 \\ & u_r, v_i \geq 0 \end{array} \tag{2}$$

where u_r is the weight (shadow price) assigned to output r and v_i is the input weight (shadow price) assigned to input i. Kao [2] proposes to use the optimal shadow prices from the dual problem to decompose the overall efficiency of the DMU into sub-unit efficiencies. In formulas:

$$e^p = \frac{\sum_r u_r y_{ro}^p}{\sum_i v_i x_{io}^p} \tag{3}$$

The computation of sub-unit efficiencies in this way allows a decomposition of the DMU efficiency:

$$E = \sum_r u_r Y_{ro} = \sum_p w^p e^p \tag{4}$$

with weights

$$w^p = \frac{\sum_i v_i x_{io}^p}{\sum_i v_i X_{io}} = \sum_i v_i x_{io}^p, \quad \forall p \tag{5}$$

where the last equality is due to the normalization constraint: $\sum_i v_i X_{io} = 1$. Notice that since the shadow prices (v_i, u_r) are non-negative, sub-unit efficiencies are always non-negative. The efficiency scores are not larger than unity, since the constraints $\sum_r u_r y_{rj}^p - \sum_i v_i x_{ij}^p \leq 0$ imply that the numerator is always lower or equal than the denominator. Therefore the sub-unit efficiency scores are contained in the unit interval: $0 \leq e^p \leq 1$. These sub-unit efficiency scores are well-defined, since at the optimal solution there is always at least one input which is strictly positive with a strictly positive shadow price for each process (thus the denominator is strictly positive). A first problem with this decomposition is that for any optimal solution of program (2), there are potentially multiple optimal solutions in terms of the weights (v_i, u_r). This means that the definition of the sub-unit efficiency scores (3) and the aggregation weights in (5) will depend on the particular choice among alternative optima. But there is another problem with this procedure that we should discuss next.

Kao [2] proposes the following model under variable returns to scale (VRS) of the underlying process technologies:

$$\begin{array}{ll} \min\limits_{\lambda_j^p,\theta} & \theta \\ s.t. & \sum_p \sum_j \lambda_j^p x_{ij}^p \leq \theta X_{io} \quad \forall i \\ & \sum_p \sum_j \lambda_j^p y_{rj}^p \geq Y_{ro} \quad \forall r \\ & \sum_j \lambda_j^p = 1 \quad \forall p \\ & \lambda_j^p, \theta \geq 0 \end{array} \tag{6}$$

with the only difference with respect to the CRS case being the constraint on the intensity variables ($\sum_j \lambda_j^p = 1$). The dual of this envelopment form is:

$$\begin{array}{ll} \max\limits_{u_r,v_i,z_p} & \sum_r u_r Y_{ro} + \sum_p z_p \\ s.t. & \sum_r u_r y_{rj}^p - \sum_i v_i x_{ij}^p + z_p \leq 0 \quad \forall p,j \\ & \sum_i v_i X_{io} = 1 \\ & u_r, v_i \geq 0 \\ & z_p \text{ free}, \forall p \end{array} \tag{7}$$

where z_p are a set of free variables. Sub-unit efficiencies are computed as:

$$e^p = \frac{\sum_r u_r y_{ro}^p + z_p}{\sum_i v_i x_{io}^p} \tag{8}$$

The weights for the aggregation into the DMU efficiency score are the same as in the CRS case, and given in Eq. (5). The DMU efficiency under VRS is:

$$E = \sum_r u_r Y_{ro} + \sum_p z_p = \sum_p w^p e^p \tag{9}$$

Notice that because of the set of constraints $\sum_r u_r y_{rj}^p - \sum_i v_i x_{ij}^p + z_p \leq 0$, the sub-unit efficiency scores in (8) are always lower or equal than unity. Unfortunately there is no guarantee that these efficiency scores are actually non-negative, since the dual shadow prices v_i, u_r are non-negative but the free variables z_p can have negative values. None of the constraints in the dual program implies that $\sum_r u_r y_{rj}^p + z_p \geq 0$, and since z_p is a free variable a negative efficiency score can be obtained for some of the processes. The numerical example in the next sub-section settles this issue in a conclusive and indisputable way.

Note that the procedure of using optimal DMU weights to assess the efficiency of sub-units has also been proposed by Kao [3] in the context of dynamic Data Envelopment Analysis models, meaning that the above mentioned problem is not only present in VRS models of parallel network systems but also in VRS versions of dynamic systems. In [4]

Table 1
Numerical example.

DMU	X_j	Y_1j	Y_2j	PROC 1			PROC 2			PROC 3		
				x_j^1	y_{1j}^1	y_{2j}^1	x_j^2	y_{1j}^2	y_{2j}^2	x_j^3	y_{1j}^3	y_{2j}^3
1	120	75	100	30	40	60	60	25	20	30	10	20
2	100	57	85	40	25	20	40	22	25	20	10	40
3	130	84	215	40	30	65	30	14	20	60	40	130
4	260	128	170	45	30	60	200	90	100	15	8	10

the author proposes a multiplicative aggregation of sub-unit efficiencies into a DMU efficiency following the above described procedure for a variety of network configurations. However, VRS versions of the models are not shown in this latest paper.

2.1. Numerical example

We use a numerical example with 4 DMUs each composed of 3 sub-units, each using a single input to produce two outputs. Table 1 shows the data for this example.

Computing sub-unit efficiencies under the VRS model using the formulas just introduced will return a negative efficiency score for process 3 of DMU 4. This invalidates the proposed decomposition (dis-aggregation) method. The application of model (7) gives rise to the optimal shadow prices reported in Table 2.

Using these shadow prices to compute sub-unit efficiencies together with the VRS formula (8) yields the results reported in Table 3, where efficiency scores are reported in percentage terms. As evidenced in this table, process 3 of DMU 4 is assigned a negative efficiency score.

For DMU 2 the weights assigned to outputs are both zero, resulting in the inability of the model to discriminate the efficiency of its processes.

3. Alternative method to decompose DMU efficiency

In this section we illustrate the method proposed in [10] to decompose DMU efficiency into meaningful components that are easy to interpret. We start by re-writing program (6) in the following way:

$$\begin{aligned}
\max_{\delta,\lambda_j^p} \quad & \delta \\
s.t. \quad & \textstyle\sum_p \sum_j \lambda_j^p x_{ij}^p \leq X_{io}(1-\delta) \quad \forall i \\
& \textstyle\sum_p \sum_j \lambda_j^p y_{rj}^p \geq Y_{ro}. \quad \forall r \\
& \textstyle\sum_j \lambda_j^p = 1 \quad \forall p \\
& \lambda_j^p, \delta \geq 0
\end{aligned} \tag{10}$$

and notice that the optimal solution of program (6) can be obtained trivially as $\theta = 1 - \delta$. We also notice that this corresponds to the directional distance function with a direction equal to $g_i = X_{io}$, therefore returning the following equivalent program:

$$\begin{aligned}
\max_{\delta,\lambda_j^p} \quad & \delta \\
s.t. \quad & \textstyle\sum_p \sum_j \lambda_j^p x_{ij}^p \leq X_{io} - \delta g_i \quad \forall i \\
& \textstyle\sum_p \sum_j \lambda_j^p y_{rj}^p \geq Y_{ro} \quad \forall r \\
& \textstyle\sum_j \lambda_j^p = 1 \quad \forall p \\
& \lambda_j^p, \delta \geq 0
\end{aligned} \tag{11}$$

The interpretation is slightly changed here, since the score δ represents the percentage by which one could reduce the input use, instead of being the reduction factor. This is in line with the directional distance function interpretation of efficiency. Apart from the interpretation, this does not change the optimal solution of the program, therefore the decomposition provided here applies to the optimal solution of the Kao [2] model.

Program (11) can be written in an equivalent way (by introducing additional decision variables) that will better emphasize the interpretation of the model. To obtain this program, first write the DMU problem (11) in standard form by adding slack variables (σ_i, τ_r) and introducing variables α^p:

$$\begin{aligned}
\max_{\alpha^p,\lambda_j^p,\sigma_i,\tau_r} \quad & \delta = \textstyle\sum_p \alpha^p \\
s.t. \quad & \textstyle\sum_p \sum_j \lambda_j^p x_{ij}^p = X_{io} - g_i \sum_p \alpha^p - \sigma_i \quad \forall i \\
& \textstyle\sum_p \sum_j \lambda_j^p y_{rj}^p = Y_{ro} + \tau_r \quad \forall r \\
& \textstyle\sum_j \lambda_j^p = 1 \quad \forall p \\
& \lambda_j^p, \alpha^p, \sigma_i, \tau_r \geq 0
\end{aligned} \tag{12}$$

We then define additional new variables $\sigma_i = \sum_p \sigma_i^p$ and $\tau_r = \sum_p \tau_r^p$, and re-arranging terms we obtain the equivalent program:

$$\begin{aligned}
\max_{\alpha^p,\lambda_j^p,\sigma_i^p,\tau_r^p} \quad & \delta = \textstyle\sum_p \alpha^p \\
s.t. \quad & \textstyle\sum_p \left(\sum_j \lambda_j^p x_{ij}^p + g_i \alpha^p + \sigma_i^p\right) = X_{io} \quad \forall i \\
& \textstyle\sum_p \left(\sum_j \lambda_j^p y_{rj}^p - \tau_r^p\right) = Y_{ro} \quad \forall r \\
& \textstyle\sum_j \lambda_j^p = 1 \quad \forall p \\
& \lambda_j^p, \alpha^p, \sigma_i, \tau_r \geq 0
\end{aligned} \tag{13}$$

Define now new decision variables $\mu_i^p = \sum_j \lambda_j^p x_{ij}^p + g_i \alpha^p + \sigma_i^p$ and $\eta_r^p = \sum_j \lambda_j^p y_{rj}^p - \tau_r^p$ and obtain the equivalent formulation:

$$\begin{aligned}
\max_{\alpha^p,\lambda_j^p,\sigma_i^p,\tau_r^p,\mu_i^p,\eta_r^p} \quad & \delta = \textstyle\sum_p \alpha^p \\
s.t. \quad & \textstyle\sum_p \mu_i^p = X_{io} \quad \forall i \\
& \mu_i^p - \alpha^p g_i - \sigma_i^p = \textstyle\sum_j \lambda_j^p x_{ij}^p \quad \forall p, i \\
& \textstyle\sum_p \eta_r^p = Y_{ro} \quad \forall r \\
& \eta_r^p + \tau_r^p = \textstyle\sum_j \lambda_j^p y_{rj}^p \quad \forall p, r \\
& \textstyle\sum_j \lambda_j^p = 1 \quad \forall p \\
& \lambda_j^p, \alpha^p, \sigma_i, \tau_r, \mu_i^p, \eta_r^p \geq 0
\end{aligned} \tag{14}$$

and finally, eliminating slack variables one obtains:

$$\begin{aligned}
\max_{\alpha^p,\lambda_j^p,\mu_i^p,\eta_r^p} \quad & \delta = \textstyle\sum_p \alpha^p \\
s.t. \quad & \textstyle\sum_p \mu_i^p = X_{io} \quad \forall i \\
& \textstyle\sum_j \lambda_j^p x_{ij}^p \leq \mu_i^p - \alpha^p g_i \quad \forall p, i \\
& \textstyle\sum_p \eta_r^p = Y_{ro} \quad \forall r \\
& \textstyle\sum_j \lambda_j^p y_{rj}^p \geq \eta_r^p \quad \forall p, r \\
& \textstyle\sum_j \lambda_j^p = 1 \quad \forall p \\
& \lambda_j^p, \alpha^p, \mu_i^p, \eta_r^p \geq 0
\end{aligned} \tag{15}$$

It should be stressed here that any problem presented in the form of program (10) can be re-stated in the form of program (15) and viceversa. In fact, any of the formulations (10)–(15) are equivalent formulations of the same problem and they will return the same optimal solution for the objective function.

It is quite natural to ask what happens to program (15) if one were to *constrain* the allocation of resources to the observed one. Setting

Table 2
VRS results for model (7).

DMU	Efficiency	u_1^*	u_2^*	v^*	z_1^*	z_2^*	z_3^*
DMU1	76.70%	0.01172	0	0.00833	−0.21875	0.0755	0.03125
DMU2	75.00%	0	0	0.01	0.3	0.3	0.15
DMU3	100%	0	0.0154	0.0077	−0.6923	−0.077	−1.54
DMU4	73.53%	0.00905	0	0.00385	−0.2466	−0.045	−0.1312

Table 3
Subunit efficiencies under model (7).

DMU	$Proc_1$	$Proc_2$	$Proc_3$
DMU1	100%	73.7%	59.375%
DMU2	75%	75%	75%
DMU3	100%	100%	100%
DMU4	14.38%	100%	−101.96%

$\mu_i^p = x_{io}^p$ and $\eta_r^p = y_{ro}^p$ will return the following program:

$$\begin{aligned} \max_{\beta^p,\lambda_j^p} \quad & \sum_p \beta^p \\ st \quad & \sum_j \lambda_j^p x_{ij}^p \le x_{io}^p - \beta^p g_i \quad \forall i,p \\ & \sum_j \lambda_j^p y_{rj}^p \ge y_{ro}^p \quad \forall r,p \\ & \sum_j \lambda_j^p = 1 \quad \forall p \\ & \lambda_j^p, \beta^p \ge 0 \end{aligned} \tag{16}$$

The optimal solution of this program will now differ from program (11) because the allocation of resources has been constrained to be the observed one. Notice also that the resource constraints ($\sum_p \mu_i^p = X_{io}$) and ($\sum_p \eta_r^p = Y_{ro}$) are trivially satisfied here and therefore can be omitted. The set of intensity variables λ_j^p is defining process specific technologies that can be used to assess the efficiency of each process separately. In other words this program can be solved as a set of P independent programs. Therefore the set of constraints of this linear program represents the production possibilities set for each production process p. This program is in all respects a standard directional distance function program, thus the objective function of (16) provides the technical inefficiency value of each process or sub-unit. This score will indicate if the sub-unit is lying inside the frontier or not. And it will provide a quantification of the excess of input that is used in the production process. It is important to stress that since program (16) has been obtained as a restriction of program (11), the optimal solution of the first will always be lower or equal than the second: $\sum_p \beta^p \le \delta$. The discrepancy between these two values is given by the fact that in program (11) we allow for reallocation of resources, while in program (16) this reallocation is prevented. Program (16) provides the overall efficiency of the system with no transfer of resources among sub-units.

In order to better understand the link between the process and the DMU programs, we can eliminate process inefficiencies altogether from the DMU input vector, thus obtaining $X_{io}^* = \sum_p (x_{io}^p - \beta^{p*} g_i)$, where β^{p*} are the scores obtained in program (16). In this way all sub-units will be technically efficient, i.e. they will lie on the production frontier. If we look now at the reallocation problem, it becomes:

$$\begin{aligned} \max_{\rho^p,\lambda_j^p,\mu_i^p,\eta_r^p} \quad & \gamma = \sum_p \rho^p \\ st \quad & \sum_j \lambda_j^p x_{ij}^p \le \mu_i^p - \rho^p g_i \quad \forall i,p \\ & \sum_j \lambda_j^p y_{rj}^p \ge \eta_r^p \quad \forall r,p \\ & \sum_j \lambda_j^p = 1 \quad \forall p \\ & \sum_p \mu_i^p = X_{io}^* \quad \forall i \\ & \sum_p \eta_r^p = Y_{ro} \quad \forall r \\ & \lambda_j^p, \rho^p, \mu_i^p, \eta_r^p \ge 0 \end{aligned} \tag{17}$$

The optimal solution of this program is equal to the difference between the solutions of programs (11) and (16). To see this, call the optimal values from program (16) β^{p*} and the optimal value from program (11) δ^*. Incidentally, these β^{p*} values are unique (there are no multiple solutions in terms of the β^p, because program (16) can be written as P separate programs with β^p as the objective function). Programs (11) to (15) are all equivalent, thus the reallocation model (17) can be equivalently written as

$$\begin{aligned} \max_{\gamma,\lambda_j^p} \quad & \gamma \\ s.t. \quad & \sum_p \sum_j \lambda_j^p x_{ij}^p \le X_{io} - \gamma g_i - g_i \sum_p \beta^{p*} \quad \forall i \\ & \sum_p \sum_j \lambda_j^p y_{rj}^p \ge Y_{ro} \quad \forall r \\ & \sum_j \lambda_j^p = 1 \quad \forall p \\ & \lambda_j^p, \gamma \ge 0 \end{aligned} \tag{18}$$

where we substituted for $X_{io}^* = \sum_p (x_{io}^p - \beta^{p*} g_i) = X_{io} - g_i \sum_p \beta^{p*}$ (the derivation between these two programs follows the derivation from (11) to (15), but in reverse order). Notice that this program is the same as program (11) with the exception that the input vector is contracted by $g_i \sum_p \beta^{p*}$. Since $X_{io} \ge g_i \sum_p \beta^{p*}$ and we are subtracting an amount that is proportional to the given directional vector $g_i = X_{io}$, the optimal value γ^* can only contract X_{io} by an amount equal to $\delta^* - \sum_p \beta^{p*}$. Notice that if the contraction factor were to be larger $\gamma > \delta^* - \sum_p \beta^{p*}$, then there would be an infeasibility (some of the binding constraints of model (11) would be violated); while if the amount where to be lower $\gamma < \delta^* - \sum_p \beta^{p*}$, it would be possible to contract further. The total optimal contraction is fixed at δ^* by program (11), which provides the maximal contraction of the input vector.

This clarifies that the discrepancy between the DMU inefficiency (δ) and the sub-unit inefficiencies ($\sum_p \beta^p$) is due to a *misallocation* of resources across the different sub-units. If one were to set the allocation variables to the efficient observed level $\mu_i^p = x_{io}^p - \beta^p g_i$ and $\eta_r^p = y_{ro}^p$, then the solution of program (17) would be $\gamma = 0$.[1] In other words, this portion of inefficiency can only be eliminated by *reallocating* resources and production across the different sub-units: without such a reallocation, *reallocation inefficiency* cannot be removed. This simple fact was noted as early as the outstanding contribution of Pachkova [11] (which, unfortunately, went unjustifiably quite unnoticed in the literature). This means that the DMU inefficiency can be *additevely* decomposed as follows (with no weights to be defined):

$$\delta = \gamma + \sum_p \beta^p \tag{19}$$

Clearly, while the technical inefficiency component ($\sum_p \beta^p$) has to do with inefficiencies arising in production at the process level, the reallocation component (γ) has to do with misallocation decisions made at the DMU level and it is therefore not a type of inefficiency which can be attributed to the individual processes (since for the processes the allocation of resources is given). The use of the directional distance function shines its best light in this setting because it can be easily aggregated in an additive fashion without requiring weights. It is also

[1] This can be seen by looking at the input constraint $\sum_j \lambda_j^p x_{ij}^p \le \mu_i^p - \rho^p g_i$ in program (17), which becomes $\sum_j \lambda_j^p x_{ij}^p \le x_{io}^p - \beta^p g_i - \rho^p g_i$. Notice now that β^p has to be optimal for program (16), the technical efficiency problem. Then at least one constraint in program (16) will state that $\sum_j \lambda_j^p x_{ij}^p = x_{io}^p - \beta^p g_i$ for some i. Finally, take this constraint and write it as $\sum_j \lambda_j^p x_{ij}^p \le x_{io}^p - \beta^p g_i - \rho^p g_i$ (i.e. in the form of program (17)): this can be true if and only if $\rho^p = 0$. Since this is true for each p, it must be that $\gamma = \sum_p \rho^p = 0$.

Table 4
DMU, Reallocation and Process inefficiencies.

	Proc 1 inefficiency	Proc 2 inefficiency	Proc 3 inefficiency	Total processes inefficiency	Reallocation inefficiency	DMU inefficiency	DMU efficiency
DMU1	0	0.1078	0.0995	0.2074	0.0257	0.2331	0.7669
DMU2	0.1	0	0	0.1	0.15	0.25	0.75
DMU3	0	0	0	0	0	0	1
DMU4	0.0577	0	0	0.0577	0.2070	0.2647	0.7353

Table 5
Percentage contribution of each component.

	Proc 1 inefficiency	Proc 2 inefficiency	Proc 3 inefficiency	Total processes inefficiency	Reallocation inefficiency	DMU inefficiency	DMU efficiency
DMU1	0	46.3%	42.7%	89%	11%	0.2331	0.7669
DMU2	40%	0	0	40%	60%	0.25	0.75
DMU3	0	0	0	0	0	0	1
DMU4	21.8%	0	0	21.8%	78.2%	0.2647	0.7353

easy to interpret since it is a contribution to the overall inefficiency of the DMU. Decomposition (19) is based on the optimal values of the objective functions of the respective programs, therefore this provides a unique way of decomposing firm inefficiency δ into reallocation inefficiency γ and technical efficiency $\sum_p \beta^p$. Nevertheless, the solution in terms of the underlying decision variables μ_i^p is not unique and there exists alternative optimal allocations of inputs and outputs that will support the given unique decomposition. Pachkova [11] provides models that can be used to select among these alternative optima in terms of the allocation decision variables.

It should also be stressed that the method does not make use of dual shadow prices, therefore nothing prevents one from using non-convex technologies. Non-convex technologies do not have in general a dual formulation, therefore multiplier forms and dual shadow prices cannot be determined in this setting. To implement the method, the only additional constraint needed in the previous programs is that the intensity variables are binary: $\lambda_j^p \in \{0,1\},\ \forall p, j$. If one were to remove the VRS constraint $\sum_p \lambda_j^p = 1$ (on the lines explained in [12,13]), this would allow also for non-convex CRS technologies. The only caveat here is computational. An enumeration algorithm exists to solve program (16) both under CRS and VRS with non-convex sets. Such an enumeration algorithm does not exist for program (11), which means such a program needs to be solved as a MILP, the computational complexity depending drastically on the number of processes P, more than the number of DMUs J (if an enumeration algorithm can be determined for this class of non-convex production problems is beyond the scope of this note). Since in the usual application the number of processes is low, computational complexity should not represent a big obstacle to the implementation of the method we are proposing in non-convex settings.

3.1. Numerical results

In Table 4 we report the results of the proposed method for the numerical example reported above. For each DMU and each process we report the inefficiency scores as computed with program (16) and the DMU inefficiency computed using program (11). The reallocation inefficiency can be either computed solving program (17) or as the difference between the DMU efficiency and the sum of the process inefficiencies. Finally, the last column reports the input radial efficiency score as determined by program (6); this score is by definition equal to one minus the DMU inefficiency reported in the second last column.

Notice that process 3 of DMU 4 which would be assigned a negative efficiency score according to the Kao model, is now deemed technical efficient (a DDF score of zero means efficiency). The bulk of DMU 4 inefficiency is clearly coming from a mis-allocation of resources, with reallocation inefficiency equal to 0.2070 and representing the largest percentage of inefficiency of the DMU. This is not surprising if one has a glance at the grossly uneven distribution of resources for DMU 4 across the 3 processes in this numerical example. Input and output targets that would implement such a reallocation can be obtained from the optimal solution of the programs. Naturally, there are various reallocations of inputs and outputs that would support the optimal solution.

Notice also the process inefficiencies for DMU 2. In the Kao model the sub-units all had the same efficiency score, while it is clear from this example that processes 2 and 3 are technically efficient (lying on the frontier) and process 1 is inefficient. Again, for DMU 2 more than half of the inefficiency is coming from a mis-allocation of resources across the sub-units.

Since the decomposition is additive and does not require weights, one has the opportunity of looking at the contribution of each component to the total inefficiency of the DMU. This is done in Table 5. For example, for DMU 4, 21.8% of the total inefficiency is contributed by production inefficiencies at the process level and 78.2% of the total inefficiency is contributed by a mis-allocation of inputs across the different production processes.

CRediT authorship contribution statement

Antonio Peyrache: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Maria C.A. Silva:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors have no conflict of interest.

Data availability

No data was used for the research described in the article.

References

[1] Kao C. Efficiency decomposition in network data envelopment analysis: a relational model. European J Oper Res 2009;192(1):949–62.
[2] Kao C. Efficiency decomposition for parallel production systems. J Oper Res Soc 2012;63(1):64–71.
[3] Kao C. Dynamic data envelopment analysis: A relational analysis. European J Oper Res 2013;227(1):325–30.
[4] Kao C. Multiplicative aggregation of division efficiencies in network data envelopment analysis. European J Oper Res 2018;270(1):328–36.
[5] Kao C. Inefficiency identification for closed series production systems. European J Oper Res 2019;275(2):599–607.
[6] Kao C. Decomposition of slacks-based efficiency measures in network data envelopment analysis. European J Oper Res 2020;283(2):588–600.
[7] Fare R, Grosskopf S, Whittaker G. Network DEA. In: Zhu J, Cook W, editors. Modelling data irregularities and structural complexities in data envelopment analysis. Springer; 2007, p. 209–40.
[8] Peyrache A, Silva MC. Efficiency decomposition for multi-level multi-components production technologies. J Prod Anal 2023;1–22.

[9] Peyrache A, Silva MC. Efficiency and productivity analysis from a system perspective: Historical overview. In: Advances in economic measurement: a volume in honour of DS Prasada Rao. Springer; 2022, p. 173–230.
[10] Peyrache A, Silva MC. Multi-level parallel production networks. Technical report, Australia: School of Economics, University of Queensland; 2021.
[11] Pachkova EV. Restricted reallocation of resources. European J Oper Res 2009;196(3):1049–57.
[12] Podinovski VV. On the linearisation of reference technologies for testing returns to scale in FDH models. European J Oper Res 2004;152(3):800–2.
[13] Briec W, Kerstens K. Input, output and graph technical efficiency measures on non-convex FDH models with various scaling laws: An integrated approach based upon implicit enumeration algorithms. Top 2006;14(1):135–66.