Centre for Efficiency and Productivity Analysis

Working Paper Series
No. WP07/2022

# A Comment on Decomposition of Efficiency in Network Production Models

Antonio Peyrache and Maria C. A. Silva

Date: June 2022

School of Economics
University of Queensland
St. Lucia, Qld. 4072
Australia

ISSN No. 1932 - 4398

# A Comment on Decomposition of Efficiency in Network Production Models[*]

Antonio Peyrache[†] and Maria C. A. Silva[‡]

**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 and 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.

Keywords: DEA; FDH; Networks; Directional Distance Function; Inefficiency.

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 Kao (2009, 2012, 2013) and see also Kao (2018, 2019, 2020) for extensions). In this note we shall refer to the formulas as proposed in Kao (2012) for the sake of clarity and simplicity. The method has the 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. (2007)). On the other hand, the method of 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 working paper (see Peyrache and Silva (2021)) we proposed a method that can retain the advantages of the Kao model,

[*]We thank two anonymous referees for rejecting one of our papers on EJOR, where the numerical example here discussed was first introduced. The comment by the two anonymous referees that a negative efficiency score was "impossible" prompted us to look deeper into this issue and make our argument sharper and more direct.

[†]CEPA, School of Economics, The University of Queensland, Australia. email: a.peyrache@uq.edu.au

[‡]CEGE - Católica Porto Business School, Rua Diogo Botelho, 1327, 4169-005 Porto, Portugal. email: csilva@porto.ucp.pt

without being affected by negative efficiency scores at the sub-unit level. 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.

# 1 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 at least one input must be strictly positive for each process in each DMU.

Kao (2012) proposed the following model to assess the efficiency of the DMU under constant returns to scale (CRS) of the underlying process technologies (after eliminating redundant constraints and slack variables):

$$\begin{aligned} \min_{\lambda_j^p, \theta} \quad & \theta \\ s.t. \quad & \textstyle\sum_p \sum_j \lambda_j^p x_{ij}^p \le \theta X_{io} \quad \forall i \\ & \textstyle\sum_p \sum_j \lambda_j^p y_{rj}^p \ge Y_{ro} \quad \forall r \\ & \lambda_j^p \ge 0 \end{aligned} \tag{1}$$

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 Kao (2009, 2012, 2013, 2018) as the parallel network model. The dual of this program returns the multiplier model which is here reported for completeness:

$$\begin{aligned} \max_{u_r, v_i} \quad & \textstyle\sum_r u_r Y_{ro} \\ s.t. \quad & \textstyle\sum_r u_r y_{rj}^p - \sum_i v_i x_{ij}^p \le 0 \quad \forall p, j \\ & \textstyle\sum_i v_i X_{io} = 1 \\ & u_r, v_i \ge 0 \end{aligned} \tag{2}$$

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 (2012) 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

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

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 $\sum_r u_r y^p_{rj} - \sum_i v_i x^p_{ij} \leq 0$ implies that the numerator is always lower or equal to the denominator. Therefore the sub-units efficiency scores are contained in the unit interval: $0 \leq e^p \leq 1$. These sub-units 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).

Kao (2012) proposes the following model under variable returns to scale (VRS) of the underlying process technologies:

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

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

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

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

$$e^p = \frac{\sum_r u_r y^p_{ro} + z_p}{\sum_i v_i x^p_{io}} \tag{8}$$

The weights for the aggregation into the DMU efficiency score are the same as in the CRS case, and given in equation (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 $\sum_r u_r y^p_{rj} - \sum_i v_i x^p_{ij} + z_p \leq 0$, the sub-unit efficiency scores in (8) are always lower or equal to 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 (2013) 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 Kao (2018) the author proposes a multiplicative aggregation of sub-units 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.

## 1.1 Numerical Example

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

Table 1: Numerical 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.

| 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 2: VRS results for model (7)

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.

| 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% |

Table 3: Subunit Efficiencies under model (7)

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.

## 2 Alternative Method to Decompose DMU Efficiency

In this section we illustrate the method proposed in Peyrache and Silva (2021) to decompose DMU efficiency into meaningful components that are easy to interpret. We start by re-writing program (6) in the following way:

$$\begin{array}{ll} \max\limits_{\delta,\lambda_j^p} & \delta \\ s.t. & \sum_p \sum_j \lambda_j^p x_{ij}^p \leq X_{io}(1-\delta) \;\; \forall i \\ & \sum_p \sum_j \lambda_j^p y_{rj}^p \;\geq Y_{ro}. \;\; \forall r \\ & \sum_j \lambda_j^p = 1 \;\; \forall p \end{array} \tag{10}$$

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:

$$\begin{array}{ll} \max\limits_{\delta,\lambda_j^p} & \delta \\ s.t. & \sum_p \sum_j \lambda_j^p x_{ij}^p \leq X_{io} - \delta g_i \;\; \forall i \\ & \sum_p \sum_j \lambda_j^p y_{rj}^p \;\geq Y_{ro} \;\; \forall r \\ & \sum_j \lambda_j^p = 1 \;\; \forall p \end{array} \tag{11}$$

The interpretation is slightly changed here, since the score $\delta$ 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 (2012) 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 $(\sigma_i, \tau_r)$ and introducing variables $\alpha^p$:

$$\begin{array}{ll} \max\limits_{\alpha^p,\lambda_j^p,\sigma_i,\tau_r} & \delta = \sum_p \alpha^p \\ s.t. & \sum_p \sum_j \lambda_j^p x_{ij}^p = X_{io} - g_i \sum_p \alpha^p - \sigma_i \;\; \forall i \\ & \sum_p \sum_j \lambda_j^p y_{rj}^p = Y_{ro} + \tau_r \;\; \forall r \\ & \sum_j \lambda_j^p = 1 \;\; \forall p \end{array} \tag{12}$$

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:

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

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:

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

and finally, eliminating slack variables one obtains:

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

In this program the variables $(\mu_i^p, \eta_r^p)$ represent the optimal allocation of resources across the $P$ sub-units that would entail the highest level of production for the DMU. If the observed allocation of resources is $(x_{io}^p, y_{io}^p)$, then $(\mu_i^p, \eta_r^p)$ is an alternative allocation of resources that optimizes the production efficiency of the DMU. The two resource constraints $(\sum_p \mu_i^p = X_{io})$ and $(\sum_p \eta_r^p = Y_{ro})$ will make sure that the total input allocation is equal to the overall input available to the DMU and the total output production at the DMU level is equal to the observed level. We call all these potential allocations of resources feasible reallocations.

It is quite natural to ask what happens to this program if one were to *constrain* the allocation of resources to the observed one. Setting $\mu_i^p = x_{oi}^p$ and $\eta_r^p = y_{or}^p$ will return the following program:

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

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 resources 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 $\lambda_j^p$ is defining process specific technologies that can be used to assess the efficiency of each process separately. Therefore the set of constraints of this linear program represents the production possibilities set for each production process. 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 to the second: $\delta \geq \sum_p \beta^p$. 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.

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 $\beta^p$ are the scores obtained in program (16). In this way all sub-units will be technically efficient, i.e. they will lay on the production frontier. If we look now at the reallocation problem, it becomes:

$$
\begin{array}{ll}
\max\limits_{\rho^p, \lambda_j^p, \mu_i^p, \eta_r^p} & \gamma = \sum_p \rho^p \\
st & \sum_j \lambda_j^p x_{ij}^p \leq \mu_i^p - \rho^p g_i \quad \forall i, p \\
 & \sum_j \lambda_j^p y_{rj}^p \geq \eta_r^p \quad \forall r, p \\
 & \sum_j \lambda_j^p = 1 \quad \forall p \\
 & \sum_p \mu_i^p = X_{io}^* \quad \forall i \\
 & \sum_p \eta_r^p = Y_{ro} \quad \forall r
\end{array}
\tag{17}
$$

The optimal solution of this program is equal to the difference between the solutions of programs (11) and (16). This clarifies that the discrepancy between the DMU inefficiency ($\delta$) 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 $\gamma = 0$. 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 (2009) (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):

$$\delta = \gamma + \sum_p \beta^p \tag{18}$$

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 ($\gamma$) 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 easy to interpret since it is a contribution to the overall inefficiency of the DMU.

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 Podinovski (2004) and Briec and Kerstens (2006)), 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.

## 2.1 Numerical Results

| | 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 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 | 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 |

Table 5: Percentage contribution of each component

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 (laying 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 inptus across the different production processes.

## References


Briec, W. and Kerstens, K. (2006). 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*, 14(1):135–166.

Fare, R., Grosskopf, S., and Whittaker, G. (2007). Network dea. In Zhu, J. and Cook, W., editors, *Modelling data irregularities and structural complexities in data envelopment analysis*, pages 209–240. Springer.

Kao, C. (2009). Efficiency decomposition in network data envelopment analysis: a relational model. *European Journal of Operational Research*, 192(1):949–962.

Kao, C. (2012). Efficiency decomposition for parallel production systems. *Journal of the Operational Research Society*, 63(1):64–71.

Kao, C. (2013). Dynamic data envelopment analysis: A relational analysis. *European Journal of Operations Research*, 227(1):325–330.

Kao, C. (2018). Multiplicative aggregation of division efficiencies in network data envelopment analysis. *European Journal of Operational Research*, 270(1):328–336.

Kao, C. (2019). Inefficiency identification for closed series production systems. *European Journal of Operational Research*, 275(2):599–607.

Kao, C. (2020). Decomposition of slacks-based efficiency measures in network data envelopment analysis. *European Journal of Operational Research*, 283(2):588–600.

Pachkova, E. V. (2009). Restricted reallocation of resources. *European Journal of Operational Research*, 196(3):1049–1057.

Peyrache, A. and Silva, M. C. (2021). Multi-level parallel production networks. Technical report, School of Economics, University of Queensland, Australia.

Podinovski, V. V. (2004). On the linearisation of reference technologies for testing returns to scale in fdh models. *European Journal of Operational Research*, 152(3):800–802.