Received: 1 December 2020 | Revised: 17 June 2021 | Accepted: 17 June 2021

DOI: 10.1111/pirs.12625

FULL ARTICLE

The multidimensional nD-GRAS method: Applications for the projection of multiregional input–output frameworks and valuation matrices

Juan Manuel Valderas-Jaramillo[1,2] | **José Manuel Rueda-Cantuche**[1]

[1]Joint Research Centre (European Commission), Spain

[2]University of Seville, Spain

Correspondence
Juan Manuel Valderas-Jaramillo, *Joint Research Centre (European Commission).*
Email: valderas@us.es

Abstract

We present a multidimensional generalization of the GRAS method (nD-GRAS) for the estimation of multiple matrices in an integrated framework. The potential applications of this method in regional and multi-regional input–output analyses based on national/regional accounts frameworks are many. We provide two real applications, a 3D-GRAS that estimates a use table at basic prices jointly with valuation matrices for Denmark; and a 4D-GRAS for estimating intercountry input–output tables with OECD data. We show that higher dimensional GRAS methods provide more consistent and accurate estimates than those with lower number of dimensions. We provide the analytical closed-form solution and the RAS-like algorithm for an easy operationalization.

KEYWORDS

GRAS, multidimensional balancing and projections, multiregional Input-Output analysis, valuation matrices, iterative proportional fitting procedure

JEL CLASSIFICATION

C61; D57; C62; C80; R15; C55

1 | INTRODUCTION

Multiple variations of biproportional techniques have been applied to the field of input–output analysis since Leontief's (1941) pioneering work, in which he used a biproportional technique to identify sources of inter-temporal change in the cells of a series of input–output tables (Lahr & de Mesnard, 2004). This also includes –broadly speaking– the RAS-family methods (Bacharach, 1965, 1970) and their extensions. A recent summary that provides a good overview and a large compilation of these methods can be found in Chapter 18 of the UN *Handbook on supply, use and input–output tables with extensions and applications* (United Nations, 2018).

Among these methods, the generalized RAS (GRAS, hereafter) is a biproportional adjustment method commonly used among input–output practitioners for matrix balancing. It can deal with positive and negative elements (Günlük-Senesen & Bates, 1988; Junius & Oosterhaven, 2003). The literature also provides several refinements of this method such as the following:

- Improving the target function in order to avoid biases (Huang et al., 2008; Lemelin, 2009; Lenzen et al., 2007)
- Dealing with row and column totals with positive and negative elements, and a non-preserving sign method (Lenzen, Moran, et al., 2014; Temurshoev et al., 2013)
- Incorporating partial information and allowing more flexibility to find a compromise solution with conflicting constraints (Lenzen et al., 2006, 2009; Paelinck & Waelbroeck, 1963).

Typically, the standard GRAS method deals with two dimensions, such as the row and column totals of a national input–output table (IOT, hereafter). In this paper, we generalize this bi-dimensional set-up into a multidimensional set-up. This new multidimensional GRAS method (nD-GRAS, hereafter) will allow us to include new sets of restrictions across every dimension.

The idea for this method stems from the fact that often in practical situations rather than simply imposing constraints summing all the elements of a matrix row-wise or column-wise (as in standard GRAS), it is necessary to rearrange a matrix representing multiregional information into arrays of a larger dimension imposing constraints on all the dimensions of the array. For instance, in multiregional frameworks, where national IOTs are split using information on bilateral exports and imports, it may be that the corresponding national use tables of imports might serve as constraints to the balancing of a multiregional IOT. Another practical situation that requires more than two dimensions is the estimation of use tables at basic prices and valuation matrices, that is, trade and transport margins tables (TTM), taxes less subsidies on products tables (TLS) to make them consistent with the use tables at purchasers´ prices. We can estimate each of those tables independently with a GRAS method, but the result of summing TTM, TLS and the use table at basic prices would be equal to the initial use table at purchasers´ prices only by chance.

The main contribution of this paper is the derivation of an analytical closed-form solution to the GRAS method in a multidimensional framework with an arbitrary number of dimensions and the algorithm to handle these problems in an accessible way. The bi-dimensional case is the standard GRAS method (2D-GRAS according to our terminology). As it will be described in the next section, the problems addressed by the nD-GRAS method can also be embedded and solved within the KRAS framework; nonetheless, our approach can easily be made operational in a RAS-like algorithm, among other differences.

In this paper, we also show two practical applications of a 3D-GRAS and a 4D-GRAS methods that can be easily implemented:

1. 3D-GRAS; the construction of use tables at basic prices, trade and transport margins tables and taxes less subsidies on products tables from a use table at purchasers´ prices for the case of Denmark and data from 2015; and
2. 4D-GRAS; an inter-country IOT (ICIOT) such as the ones published by the OECD (oe.cd/icio) provides information about commodities used by intermediate and final users by country of origin and country of destination, which actually reflect four dimensions: commodities, users, country of origin and country of destination. We will show

how the 4D-GRAS method can be used to project ICIO tables from 2014 to 2015. The so-called Extended Supply and Use[1] tables can also be seen as examples of more than four dimensions, as in a real *n*-dimensional case.

Although information might not be complete for all constraints in each dimension, this should not prevent to develop the theoretical aspects of higher dimensional nD-GRAS methods such as the 4D-GRAS showed in this paper. Actually, Leontief and Strout (1963) developed the theoretical framework of multiregional input–output tables already in the 1960s, when there were very scarce data to populate them, if any.

Finally yet importantly, another contribution of this paper is to show that projections based on a higher number of dimensions, apart from the global coherence of the estimations, lead to better performance and greater accuracy than independent projections based on a lower number of dimensions. For instance, in general, the 3D slices resulting from the projections using a 4D-GRAS method yield a better fit than the results that we could achieve using a 3D-GRAS applied independently to each of the 3D slices of the 4D array. We will also see that this statement holds for the 3D-GRAS method and the 2D matrices projections.

This paper is organized as follows. The next section frames the theoretical background of our work within the latest related literature. Section 3 introduces how several important applications can be embedded in our multidimensional approach. Section 4 contains all the theory about the nD-GRAS, and the set-up and solution of the optimization algorithm. Section 5 explores some conditions regarding feasibility issues and convergence. Section 6 provide two examples for the 3D-GRAS and 4D-GRAS methods. Finally, a concluding section provides a summary of the main findings and theoretical contributions.

2 | LITERATURE REVIEW

The RAS method and other related biproportional techniques fall under the category of what is known in other fields as iterative proportional fitting procedures (IPFP). In a bi-dimensional context, Deming and Stephan (1940) used similar methods for the estimation of contingency tables.[2] We can find other early applications of these methods in the literature by Sheleikhovskii (Bregman, 1967) and Kruithof (Lahr & de Mesnard, 2004). Generalization for three-dimensional contingency tables was done by Deming (1943),[3] and for larger multidimensional contingency tables by Darroch (1962) and Ireland and Kullback (1968). A good summary of these techniques, their implementation and basic literature references can be found in the documentation of the R-package "mipfp" developed by Barthelemy et al. (2018).

We can also find other examples in the recent literature concerning the multidimensional generalization of the RAS-family methods. Tilanus (1976) first introduced this approach in an algorithmic way, generalizing the biproportional algorithm of RAS to four dimensions. Oosterhaven et al. (1986) introduced a method for estimating an interregional input–output system in a bi-dimensional RAS set-up where the regional cells must add up to a national figure. The approach followed by Oosterhaven et al. (1986) is similar to the multiregional GRAS (MR-GRAS) method developed by Temursho, Oosterhaven, and Cardenete (2020). Both methods constitute a bi-dimensional set-up where the additional national constraint provides a sort of third dimension. In the case of the MR-GRAS, this method includes an additional set of constraints (different from the typical row-wise and column-wise sum constraints) in a multiregional framework using the same bi-dimensional objective function as the standard GRAS method. In the MR-GRAS method, the third set of linear restrictions operates across any non-overlapping subsets of elements in the multiregional IOT that must add up to a given total. As Temursho, Oosterhaven, and Cardenete (2020) show, this approach can be adapted for updating inter-national/regional or global SUTs, where the third dimension constraint is introduced for the interregional blocks add up to a given figure. The MR-GRAS approach has been extensively used for the creation of a baseline scenario called PIRAMID[4] for the 2018 Global Energy and Climate Outlook (Rey Los Santos et al., 2018; Temursho, Cardenete, et al., 2020) in the projection of national IOTs in a multiregional context for future years, ensuring the consistency of the projections with National Accounts.

In principle, the approaches of Oosterhaven et al. (1986) and Temursho, Oosterhaven, and Cardenete (2020) are not a real three-dimensional approach because the authors use a bi-dimensional set-up with constraints over these two dimensions, hence, the third dimension is not taken into account as such. Nonetheless, by introducing this additional set of restrictions, the methods of Oosterhaven et al. (1986) and Temursho, Oosterhaven, and Cardenete (2020) produce a three-dimensional solution.

The 3D-GRAS can be solved in terms of Temursho, Oosterhaven and Cardenete's approach, since it is possible to formulate a three-dimensional array as a standard matrix, thus making these two approaches correspondent. However, as Temursho, Oosterhaven, and Cardenete (2020) rightly mention, the nD-GRAS method is more general, accounting for additional sets of non-overlapping restrictions that are included to account for other dimensions. Indeed, the nD-GRAS method is more general and so we describe it in this paper in a geometrical and intuitive way. We also show a solution algorithm that makes the efficient implementation of this method straightforward, regardless the number of dimensions and without requiring the use of aggregation matrices. The nD-GRAS method was developed within the Eurostat's FIGARO[5] project (Remond-Tiedrez & Rueda-Cantuche, 2019) and it is being profusely used in the construction of the European inter-country supply, use and input–output tables (SUIOTs).

Other examples of RAS-like algorithm approaches can be found in Gilchrist and St. Louis (1999, 2004) and Lenzen et al. (2006). The TRAS algorithm introduced in the papers of Gilchrist and St. Louis sets up a sort of three-dimensional RAS that accounts for additional aggregation constraints apart from row-wise and column-wise sums. The "cRAS" algorithm proposed by Lenzen et al. (2006) generalizes the bi-dimensional RAS algorithm to account for additional aggregation constraints. In all these papers, as in Tilanus (1976), only the resolution algorithm is presented as a practical way of obtaining the estimated matrix that meets all the desired constraints without mathematical proof. Another three-dimensional approach where valuation matrices, basic and purchasers prices inputs are jointly estimated is in Dalgaard and Gysting (2004). In our opinion, the algorithm proposed by these authors does not fall into the category of multidimensional algorithms, as there is not a multidimensional constraint, such that valuation matrices and basic prices table must add up cell-wise to the purchasers' prices table. Instead, the balancing of the GDP on the output side and the demand side respecting the outputs provided by the supply table is the target of this algorithm. The basic prices and valuation matrices are derived individually using a sort of proportional adjustment and a RAS-like algorithm for balancing, and the purchasers' prices table is the sum of all of them. The algorithm continues until the desired balance is achieved.

Holý and Šafr (2020) introduced a multidimensional RAS method (DRAS) that reformulates the bi-dimensional RAS problem in a purely multidimensional set-up that deals with only non-negative arrays. In Holý and Šafr (2020), the method was introduced only in an algorithmic way, similar to Tilanus (1976). Holý and Šafr (2020) prove that this algorithm is the solution of the cross-entropy optimization model in a multidimensional generalization. The DRAS is a particular case of the nD-GRAS when no negative elements exist in the initial matrix. They apply the DRAS in a three-dimensional set-up for the estimation of regional, quarterly and domestic/imported input–output (industry-by-industry) tables of the Czech Republic. Their results show that the addition of a third dimension, apart from ensuring the consistency of national totals among the different layers (either quarterly disaggregation, regional disaggregation or domestic/imported split) allows more accuracy than the standard estimates in terms of the overall input–output structure. The application of the multidimensional RAS method to Isard's interregional input–output model also shows better results than the standard RAS method.

Another decisive development in the field of input–output projections is the continuation of methods developed from the KRAS method introduced by Lenzen et al. (2009). In the KRAS method, the vectorization of the target matrix allows the formulation of the optimization problem in a unidimensional set-up. This vectorization allows all potential constraints to be embedded, such as linear constraints on arbitrarily sized and shaped subsets of matrix elements either with unity or non-unity coefficients, in a generalized formulation. The KRAS method also incorporates other features such as the reliability of the information supplied by the constraints and the autonomous management of potential conflicts with external data in the case of inconsistent constraints.

The KRAS method is at the core of multiple applications that deal with multidimensional problems. In Geschke et al. (2011) a tool with a custom data processing language (AISHA) for the construction of series contingency tables is described. This tool performs the estimation of large dimension contingency tables. Geschke et al. (2011) describes how a multidimensional representation can be vectorized in a one-to-one correspondence with a unidimensional vector. This methodology is implemented in Lenzen et al. (2013) in the construction of the EORA MRIO database, where an eight-tiered hierarchy is used. This does not actually imply an eight-dimension problem in the same way as in an 8D-GRAS method would work. That is, even though all the elements can be embedded in an 8D-array, the constraints do not add up over all the eight dimensions together in a array. Nonetheless, some of the constraints represent 3D or 4D aggregations in an identical way to the 3D-GRAS and 4D-GRAS problems. Additional features of this methodology include the possibility of using different optimization functions apart from the GRAS optimization function. Among them, we have quadratic programming or barrier and penalty functions (Geschke et al., 2011; Lenzen et al., 2012). In addition, different solvers prepared to deal with large-scale and sparse matrices, including parallelization techniques and Cimmino algorithms (Geschke et al., 2019; Lenzen, Geschke, et al., 2014) are available. Besides, this framework includes the possibility of dealing with tailored and different regional and sectoral aggregations defined as a subset of a fixed classification (Lenzen, Geschke, et al., 2014).

This methodological proposal, deriving from the KRAS method paper, goes one step further with the development of virtual laboratories introduced in Lenzen et al. (2012, 2017), Lenzen, Geschke, et al. (2014) and Geschke and Hadjikakou (2017). In these virtual laboratories, researchers can assemble their own MRIO versions in a collaborative research environment using cloud-computing platforms enabling a multitude of input–output applications in carbon, water, ecological footprints, life-cycle assessments and trend or key driver analyses.

Undoubtedly, the problems addressed by the nD-GRAS method can be embedded and solved in this KRAS framework, as long as a feasible solution exists, using the same objective function, a set of constraints with unity coefficients and no conflicting information. This also applies, not only to the nD-GRAS method, but also to the rest of the multidimensional methods mentioned in this paper such as those proposed by Holý and Šafr (2020) or Temursho, Oosterhaven, and Cardenete (2020). Even though the advantages of the extended KRAS methodology and virtual labs proposals are obvious, other aspects also have to be taken into account.

The nD-GRAS proposal has the advantage of having a closed-form solution, at the cost of using only linear restrictions with unity coefficients. Finding a solution also involves providing non-conflicting exogenous information and a well-posed prior matrix. However, the mathematical derivation of the solution of the multidimensional problem and the associated algorithm enables a large variety of problems to be solved without IT hardware. On the other hand, the multiple features of the KRAS methodology to be handled in the process, such as reliability and clearing up conflicting information, but also of the size of the elements involved in the estimation problem in terms of regions, products and sectoral disaggregation can be a barrier to entry for users. Besides, the vectorization of the target matrix usually leads to an optimization problem that requires the management of very large and very sparse matrices that are especially demanding in terms of computing performance. In addition, the operationalization of some aspects of these methods, such as the construction of the constraints matrix, requires the use of automation and "data mining, processing, and reclassification procedures as much as possible" (Lenzen et al., 2012, p. 8376).

Optimization algorithms are also another way to address the kind of problems covered by the nD-GRAS. Jackson and Murray (2004) provide in their article an excellent review bridging between iterative techniques and optimization algorithms. The recent proliferation of RAS-based approaches already mentioned in this article, may cause an impression that these RAS extensions sacrifice simplicity for capability (KRAS is probably a good example of this). Although the generalization of the GRAS problem to multiple dimensions may seem complex, the existence of an analytical solution and an algorithm for its implementation makes it easy to deal with. Besides, this algorithm can be efficiently implemented in an easy way on widely available free software programs like R. Alternatively, more complex numerical optimization techniques may require commercial software that need to be solved with high-performance solvers embedded in this commercial software. However, it is also true that these optimization techniques can deal with a larger potential for setting complex constraints over any number of dimensions, subsets of coefficients, try different weighting patterns, penalties, and more complex objective functions. In the case of infeasibilities, in iterative process like RAS-based approaches, the reason for non-convergence of the algorithm is easier to trace compared to the complex algorithms underlying optimization routines (see Temursho, Cardenete, et al., 2020). It is also possible to inspect some qualitative aspects of the relationship between the prior and targets (as we will see in Section 5) or checking the paths followed by the updating factors in the iterative process.

3 | MULTIDIMENSIONAL GRAS BALANCING PROBLEMS IN PRACTICE

Before introducing the multidimensional GRAS method, we illustrate in Figure 1 how we can frame several practical situations into this methodology. Going from the simpler to the more general, we start with some 3D-GRAS examples.

A multiregional framework is a fertile ground for multiple 3D-GRAS applications. Let us assume that we have a matrix of a multiregional input–output framework. In this matrix, row elements (products or industries) are usually arranged by countries/regions of the multiregional framework. The same also applies column-wise.

For instance, if we think of a matrix representing bilateral trade for multiple regions. (i.e. the domestic part is voided), rows will denote exports of products by destination partner/user, and columns, imports of products by country of origin.

Figure 1 shows a matrix that schematically represents, either a standard multiregional use framework, or a bilateral international trade matrix (in such case with the elements of the main diagonal block—shaded light grey—set to zero). If we concentrate our attention on, say, trading partner (region) *k* (shaded in Figure 1), we see a description of region *k*'s imported products by region of origin. In fact element $a_{i}jr$ represents, for region *k*, the imports of product *i* by user *j* coming from region *r* (the main diagonal block equal to zero implies that for every destination region *k*, $a_{ijr} = 0$ for $r = k, \forall i,j$).

In this context, the column block of region *k*, represented in Figure 1 by elements $a_{i}jr$, could be embedded in a 3D array, as it is portrayed in the upper right-hand side of Figure 1. If we aggregate by region of origin (i.e., we aggregate $\sum_r a_{ijr} = a_{ij\bullet}, \forall i,j$) the result is a typical import use matrix of a national input–output framework, as described in Figure 1. The dark grey elements of Figure 1 depict graphically that $a_{11\bullet} = \sum_{r=1}^{k} a_{11r}$.

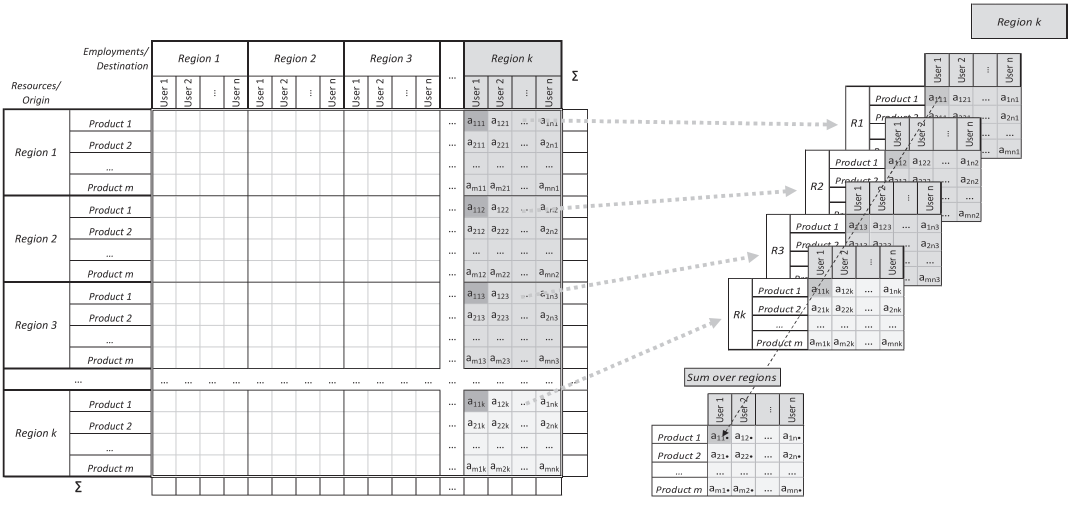

FIGURE 1 Multicountry framework and multidimensional relationships

Many practical situations in multiregional frameworks fall into this representation. For instance, balancing a multiregional framework where the totals of products and users by region are known (i.e., a typical biproportional balancing set-up). If an additional set of constraints is added (e.g., implying that the sum of the same elements by region add up to a known total), then this becomes a 3D generalization of a biproportional balancing. In addition, we can estimate a set of regional input–output tables where regional target vectors are known, adding all the regional tables up to a national one. We can solve these problems straightforwardly with the 3D-GRAS method.

This is also the situation when estimating a framework consisting of a use table at basic prices jointly with their valuation matrices. As we can see in Figure 2, the use table at basic prices, the trade and transport margin matrix and the taxes less subsidies matrix set up a 3D system with the use table at purchasers' prices being the 3D target.

We may also think that the third dimension accounts for the *territorial* dimension of the problem, and therefore, the other two dimensions are products and industries.[6]

We can also consider time as a third dimension. In this situation, for instance, the aggregated table would be an annual table, and the third dimension would account for a temporal disaggregation (e.g., quarterly). The same applies for the breakdown of total coefficients into domestic and imported figures (see Holý & Šafr, 2020).

When performing this problem, one can interpret that we are performing three bi-dimensional GRAS optimizations simultaneously: one for every layer of the system of matrices. Nonetheless, these bi-dimensional GRAS problems are not accounted for independently; on the contrary, the third dimension constraint—cell-wise sum of basic prices matrix plus valuation matrices adding up to the desired purchasers' prices values—ensures a global approach connecting all the problems into an integrated single problem that performs all the balancing simultaneously. A practical application of this approach is illustrated in Section 6.

It will always be possible to project multidimensional arrays in a bi-dimensional setup (see Figure 1, the whole multicountry frameworks constitutes a bi-dimensional representation of a 4D contingency table). However, as the number of dimensions increase, it becomes more complex to define a set of aggregator matrices to meet all constraints across all dimensions. The inclusion of new dimension implies a new different set of independent multipliers (i.e., one cell can only be used once to meet a constraint for one specific dimension and it cannot be used again to meet others). The geometrical intuition makes easier the implementation of this feature and so we describe it in this paper.

Finally, we can also find an example of application of the 4D-GRAS technique in the estimation of a global multiregional input–output framework. First, it is very important to note that, as already mentioned, even though we have expressed a multiregional framework in a bi-dimensional matrix, this must not hide the fact that it is a 4D array, since every element has to be described by four sub-indexes. Hence, the multiregional framework should be expressed as $A = [a_{ijkl}]$ where sub-index i represents the origin dimension, j represents the destination dimension, k represents the product dimension, and l represents the user dimension.

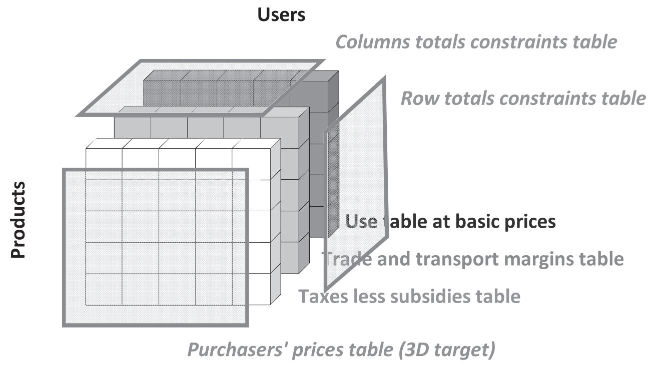

FIGURE 2 3D array and bi-dimensional margins

Our target now is the estimation of the full 4D array in one go. For this problem, it is necessary to have a prior of the 4D array, and the set of 3D array margins (i.e., sums across each of the dimensions of the 4D array). In formal language, for a 4D array, $\mathbf{A} = [a_{ijkl}]$, four 3D arrays constitute its margins:

$$\begin{aligned}
\mathbf{A}_1 &= [a_{ijk\bullet}] = \sum_l a_{ijkl} \forall i,j,k,\\
\mathbf{A}_2 &= [a_{ij\bullet l}] = \sum_k a_{ijkl} \forall i,j,l,\\
\mathbf{A}_3 &= [a_{i\bullet kl}] = \sum_j a_{ijkl} \forall i,k,l,\\
\mathbf{A}_4 &= [a_{\bullet jkl}] = \sum_i a_{ijkl} \forall j,k,l.
\end{aligned}$$

The 3D array $\mathbf{A}_1$ is an extension of a balanced view of trade including the domestic use of products, regardless of the user dimension. This matrix can be readily available in a multiregional context. Also, the $\mathbf{A}_4$ array is the use table with total values (domestic plus exported) for every destination country. Arrays $\mathbf{A}_2$ and $\mathbf{A}_3$ are less likely to be found in practice, but they all fall beyond the range of pieces to be estimated in the compilation process of a multiregional framework and at least some partial information may be available in some Statistical Offices for some countries. As long as they are available, the 4D balancing can be used to estimate the global 4D array. We will illustrate this approach in Section 6 with an example of estimation for the OECD ICIO tables.

It is important to note the difference between a 4D projection and the estimation of 4D array using 3D projections. The bi-dimensional table represented in Figure 1 is essentially a 4D array. Conditioning on a column block such as destination region k, it is a 3D array, that is, a 3D slice of the full 4D array, as represented in Figure 1.

Hence, it is possible to split the 4D array into a set of independent 3D arrays for every destination region l. As long as the bi-dimensional margins of these 3D slices are known, we could perform independent 3D-GRAS methods for each destination region l (i.e., selecting different slices every time). At the end, we will have an estimation of the full 4D array. However, the result is not necessarily consistent with all of the 3D margins $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ and $\mathbf{A}_4$. It is exactly the same situation that takes place in a 3D array. Recalling our previous example for the use basic prices and valuation matrices example described in Figure 2, if all the marginal 2D vectors of these matrices are known, it is possible to perform a 2D-GRAS method to estimate each of these bi-dimensional slices of the 3D array independently. However, this does not imply that the result is consistent with the use table at purchasers' prices matrix.

Hence, the global integrated estimation in a higher dimension adds consistency to all the elements in the system and, as we will see, this has a positive impact on the quality of the estimations.

Another advantage of the nD-GRAS method is that it does not require having a full knowledge of the constraint values, although we have assumed in this paper that they were all known across all dimensions. If some constraint values are missing, the nD-GRAS method skips them in a natural way. For instance, this would be similar to implement a standard bi-dimensional GRAS where some of the columns and/or row totals are unknown. The solution algorithm can be executed leaving such row and/or column vector totals free.

4 | GENERALIZING THE GRAS PROBLEM TO MULTIPLE DIMENSIONS (ND-GRAS)

In this section, we present the n-dimension generalization of the GRAS technique. For the sake of simplicity, and making easy to understand this problem from a mathematical point of view, we initially expose the method for a tri-dimensional array (3D-GRAS), and we will further generalize it for larger dimensions. As in a standard GRAS setup, the 3D-GRAS problem consists of three elements: the data (prior and constraints), the problem to be solved and the model that sets up the problem in a mathematical way.

As usual, a prior tridimensional array $\mathbf{A} = [a_{ijk}]$ is necessary, where $i = 1,\ldots,m$; $j = 1,\ldots,n$ and $k = 1,\ldots,s$. If we take any slice of the 3D array $\mathbf{A}$ along, say, the third dimension $(\mathbf{A}_k)$ is a bidimensional matrix. We will need two vectors as constraints for every slice, $\mathbf{u}_k$ and $\mathbf{v}_k$, that can be easily arranged into two bidimensional matrices, $\mathbf{u}$ and $\mathbf{v}$. A third matrix $\mathbf{t}$, that constitutes the constraints in the third dimension (the regional aggregated matrix) is required.

Our goal is to find a new array $\mathbf{X} = [x_{ijk}]$ that deviates least from the given array $\mathbf{A}$ that satisfies:[7]

$$\sum_{j=1}^{n} x_{ijk} = u_{ik} \forall i,k,$$
$$\sum_{i=1}^{m} x_{ijk} = v_{jk} \forall j,k,$$
$$\sum_{k=1}^{s} x_{ijk} = w_{ij} \forall i,j.$$

For that purpose, we will find updating factors $\theta_{ijk} \in \mathbb{R}^+$ which allows us to build $\mathbf{X} = [x_{ijk}]$ meeting the constraints described above. In other words:

$$\theta_{ijk} = \begin{cases} \dfrac{x_{ijk}}{a_{ijk}} & if \quad a_{ijk} \neq 0 \\ 1 & if \quad a_{ijk} = 0 \end{cases}.$$

The objective function is a tridimensional version of the improved objective function of Huang et al. (2008), based on the Kullback–Leibler cross-entropy function (Zhou et al., 2010). This objective function is preferred to the simpler version used by Lenzen et al. (2007) since it is non-negative and it can also be considered a generalization of the normalized squared differences method of Friedlander (1961) or Lecomber (1975).[8] Hence, assuming that $x_{ijk} = \theta_{ijk} \cdot a_{ijk}$ our objective function is:

$$\theta_{ijk} = \operatorname{argmin} \sum_{i,j,k} |a_{ijk}| \left[\theta_{ijk}(\ln\theta_{ijk} - 1) + 1\right] \tag{1}$$

s.t.

$$\sum_{j=1}^{n} \theta_{ijk} \cdot a_{ijk} = u_{ik} \forall i,k,$$
$$\sum_{i=1}^{m} \theta_{ijk} \cdot a_{ijk} = v_{jk} \forall j,k,$$
$$\sum_{k=1}^{s} \theta_{ijk} \cdot a_{ijk} = w_{ij} \forall i,j.$$

The related Lagrange function of this problem is:

$$\begin{aligned} L(\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\tau},\boldsymbol{\eta}) = \sum_{i,j,k} |a_{ijk}| \left[\theta_{ijk}(\ln\theta_{ijk} - 1) + 1\right] + \sum_{i,k} \lambda_{ik} \left(u_{ik} - \sum_{j=1}^{n} \theta_{ijk} \cdot a_{ijk}\right) \\ + \sum_{j,k} \tau_{ik} \left(v_{jk} - \sum_{i=1}^{m} \theta_{ijk} \cdot a_{ijk}\right) + \sum_{j,k} \eta_{ik} \left(w_{ij} - \sum_{k=1}^{s} \theta_{ijk} \cdot a_{ijk}\right). \end{aligned} \tag{2}$$

The optimized $\theta_i jk$ consists of the multiplication of three updating factors, one for each dimension.

$$\theta_{ijk} = \begin{cases} e^{\lambda_{ik}} \cdot e^{\tau_{jk}} \cdot e^{\eta_{ij}} & \text{if} \quad a_{ijk} > 0 \\ 0 & \text{if} \quad a_{ijk} = 0. \\ e^{-\lambda_{ik}} \cdot e^{-\tau_{jk}} \cdot e^{-\eta_{ij}} & \text{if} \quad a_{ijk} < 0 \end{cases}$$

Calling $r_{ik} = e^{\lambda_{ik}}$, $s_{jk} = e^{\tau_{jk}}$ and $t_{ij} = e^{\eta_{ij}}$ the target matrix is:

$$x_{ijk} = \begin{cases} a_{ijk} \cdot r_{ik} \cdot s_{jk} \cdot t_{ij} & \text{if} \quad a_{ijk} > 0 \\ 0 & \text{if} \quad a_{ijk} = 0 \cdot \\ \dfrac{a_{ijk}}{r_{ik} \cdot s_{jk} \cdot t_{ij}} & \text{if} \quad a_{ijk} < 0 \end{cases}$$

Following an analogous procedure to a standard bidimensional GRAS—see Temurshoev et al. (2013) or Lenzen et al. (2007)—the expressions for the updating factors are:

$$r_{ik} = \begin{cases} \dfrac{u_{ik} + \sqrt{u_{ik}{}^2 + 4 \cdot P_{ik}(\mathbf{s},\mathbf{t}) \cdot N_{ik}(\mathbf{s},\mathbf{t})}}{2 \cdot P_{ik}(\mathbf{s},\mathbf{t})} & \text{if} \quad P_{ik}(\mathbf{s},\mathbf{t}) > 0 \\ \dfrac{-N_{ik}(\mathbf{s},\mathbf{t})}{u_{ik}} & \text{if} \quad P_{ik}(\mathbf{s},\mathbf{t}) = 0 \end{cases}. \tag{3}$$

$$s_{jk} = \begin{cases} \dfrac{v_{jk} + \sqrt{v_{jk}{}^2 + 4 \cdot P_{jk}(\mathbf{r},\mathbf{t}) \cdot N_{jk}(\mathbf{r},\mathbf{t})}}{2 \cdot P_{jk}(\mathbf{r},\mathbf{t})} & \text{if} \quad P_{jk}(\mathbf{r},\mathbf{t}) > 0 \\ \dfrac{-N_{jk}(\mathbf{r},\mathbf{t})}{v_{jk}} & \text{if} \quad P_{jk}(\mathbf{r},\mathbf{t}) = 0 \end{cases}. \tag{4}$$

$$t_{ij} = \begin{cases} \dfrac{w_{ij} + \sqrt{w_{ij}{}^2 + 4 \cdot P_{ij}(\mathbf{r},\mathbf{s}) \cdot N_{ij}(\mathbf{r},\mathbf{s})}}{2 \cdot P_{ij}(\mathbf{r},\mathbf{s})} & \text{if} \quad P_{ij}(\mathbf{r},\mathbf{s}) > 0 \\ \dfrac{-N_{ij}(\mathbf{r},\mathbf{s})}{w_{ij}} & \text{if} \quad P_{ij}(\mathbf{r},\mathbf{s}) = 0 \end{cases}. \tag{5}$$

with

$$\begin{aligned} &P_{ik}(\mathbf{s},\mathbf{t}) = \sum\nolimits_j p_{ijk} \cdot s_{jk} \cdot t_{ij}, N_{ik}(\mathbf{s},\mathbf{t}) = \sum\nolimits_j \frac{n_{ijk}}{s_{jk} \cdot t_{ij}}, P_{jk}(\mathbf{r},\mathbf{t}) = \sum\nolimits_i p_{ijk} \cdot r_{ik} \cdot t_{ij}, \\ &N_{jk}(\mathbf{r},\mathbf{t}) = \sum\nolimits_i \frac{n_{ijk}}{r_{ik} \cdot t_{ij}}, P_{ij}(\mathbf{r},\mathbf{s}) = \sum\nolimits_k p_{ijk} \cdot r_{ik} \cdot s_{jk} \ \text{ and } \ N_{ij}(\mathbf{r},\mathbf{s}) = \sum\nolimits_k \frac{n_{ijk}}{r_{ik} \cdot s_{jk}}, \end{aligned} \tag{6}$$

where $p_{ijk} = \begin{cases} a_{ijk} & \text{if} \ \ a_{ijk} > 0 \\ 0 & \text{otherwise} \end{cases}$ and $n_{ijk} = \begin{cases} -a_{ijk} & \text{if} \ \ a_{ijk} < 0 \\ 0 & \text{otherwise} \end{cases}$.which can be solved by an iterative process in the following way.

- Step 0. Assume that all the factors are initially equal to one at beginning

$$r_{ik}(0) = s_{jk}(0) = t_{ij}(0) = 1 \qquad \forall i,j,k.$$

Iteration 1:

- Step 1. Use Equation 3 to calculate $r_ik(1)\forall i,k$, based in $s_jk(0)$ and $t_ij(0)$.
- Step 2. Move to Equation 4 to calculate $s_jk(1)\forall j,k$, based in $r_ik(1)$ and $t_ij(0)$.
- Step 3. Move to Equation 5 to calculate $t_ij(0)\forall i,j$, based in $r_ik(1)$ and $s_jk(1)$.

This algorithm continues sequentially, initiating a new iteration that repeats steps 1 to 3 for every iteration. This is represented in the flow diagram of Figure 3, being d the iteration number, for $d \geq 1$.

The iterations will continue until the algorithm converges in terms of a desired metrics. For instance, a demanding metrics could be the one that concerns the maximum absolute error of the difference between the targets and the totals of the updated matrix:

$$\max_{i,j,k}\left[\sum_{i=1}^{m}|x_{ijk}(d)-v_{jk}|,\sum_{j=1}^{n}|x_{ijk}(d)-u_{ik}|,\sum_{k=1}^{s}|x_{ijk}(d)-w_{ij}|\right]<\varepsilon,$$

$$\text{with} x_{ijk}(d)=\begin{cases} a_{ijk}\cdot r_{ik}(d)\cdot s_{jk}(d)\cdot t_{ij}(d) & \text{if} \quad a_{ijk}>0 \\ 0 & \text{if} \quad a_{ijk}=0\cdot \\ \dfrac{a_{ijk}}{r_{ik}(d)\cdot s_{jk}(d)\cdot t_{ij}(d)} & \text{if} \quad a_{ijk}<0 \end{cases}$$

As long as the error is larger than a desired threshold $\varepsilon > 0$, the process will continue starting a new iteration. The more demanding a metric or the threshold is, the higher the number of iterations and the longer the time required for the algorithm to converge. Other standard metrics consists of introducing a bound to the maximum absolute relative error:

$$\max_{i,j,k}\left[\sum_{i=1}^{m}\frac{|x_{ijk}(d)-v_{jk}|}{|v_{jk}|},\sum_{j=1}^{n}\frac{|x_{ijk}(d)-u_{ik}|}{|u_{ik}|},\sum_{k=1}^{s}\frac{|x_{ijk}(d)-w_{ij}|}{|w_{ij}|}\right]<\varepsilon.$$

In another common metrics used in Lenzen et al. (2007) or Temurshoev and Timmer (2011), the algorithm stops when the variation of factors is below a desired threshold:

$$\max_{i_j,k}\left[|r_{ik}(d)-r_{ik}(d-1)|,|s_{jk}(d)-s_{jk}(d-1)|,|t_{ij}(d)-t_{ij}(d-1)|\right]<\varepsilon. \tag{7}$$

This solution of the 3D-GRAS method generalizes the concept of biproportion (see De Mesnard, 1994) to tridimensional proportion, but in a standard setup like the standard GRAS with positive and negative figures. This is

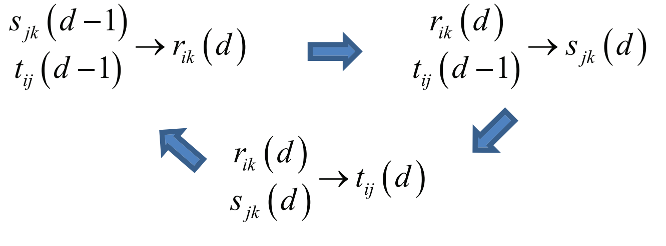

FIGURE 3 Algorithmic flow to compute updating factor in a 3D-GRAS

described in Figure 4, where element a_{125} (assumed to be positive in this example) is updated by a factor, θ_{125} consisting of the multiplication of three factors $r_1,5 \cdot s_2,5 \cdot t_1,2$.

The 3D-GRAS method can be easily generalized to higher dimensions if necessary, as well as the solution algorithm, and the result would consist of a multidimensional proportion (see de Mesnard, 2020). For instance, if we wanted to add a fourth dimension, our prior would be a 4D array and we would need to know the totals for the constrains in the four dimensions (i.e., 3D arrays). There would be a set of four factors, one for every dimension: r,s,t—in an analogous way to the 3D version—and a fourth factor, say h, for the fourth dimension. The updated value would be:

$$x_{ijkl} = \begin{cases} a_{ijkl} \cdot r_{ikl} \cdot s_{jkl} \cdot t_{ijl} \cdot h_{ijk} & \text{if} \quad a_{ijkl} > 0 \\ 0 & \text{if} \quad a_{ijkl} = 0, \\ \dfrac{a_{ijkl}}{r_{ikl} \cdot s_{jkl} \cdot t_{ijl} \cdot h_{ijk}} & \text{if} \quad a_{ijkl} < 0 \end{cases}$$

where the factors $\mathbf{r},s,t$ would have expressions analogous to (3), (4) and (5), redefining the terms P and N in (6) as accounting for the fourth dimension where every element a_ijkl is multiplicated by the rest of factors involved. This is illustrated in Equations 8 and 9 where the expression of the new factor $\mathbf{h}$ is provided:

$$h_{ijk} = \begin{cases} \dfrac{b_{ijk} + \sqrt{b_{ijk}^2 + 4 \cdot P_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t}) \cdot N_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t})}}{2 \cdot P_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t})} & \text{if} \quad P_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t}) > 0 \\ \dfrac{-N_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t})}{b_{ijk}} & \text{if} \quad P_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t}) = 0 \end{cases}, \tag{8}$$

being b_ijk the generic constraining element in the fourth dimension, and

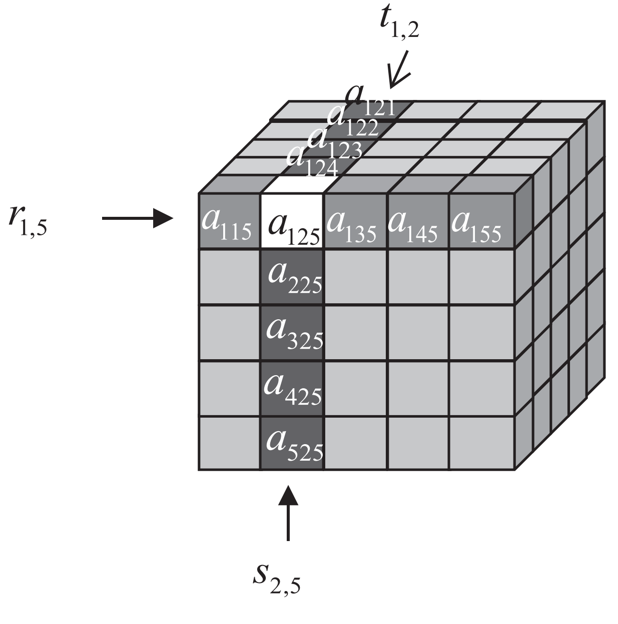

FIGURE 4 Factors in a 3D-GRAS set-up

$$P_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t}) = \sum\nolimits_l p_{ijkl} \cdot r_{ikl} \cdot s_{jkl} \cdot t_{ijl}, N_{ijk}(\mathbf{r},\mathbf{s},\mathbf{t}) = \sum\nolimits_l \frac{n_{ijkl}}{r_{ikl} \cdot s_{jkl} \cdot t_{ijl}}, \tag{9}$$

where $p_{ijkl} = \begin{cases} a_{ijkl} & \text{if } a_{ijkl} > 0 \\ 0 & \text{otherwise} \end{cases}$ and $n_{ijkl} = \begin{cases} -a_{ijkl} & \text{if } a_{ijkl} < 0 \\ 0 & \text{otherwise} \end{cases}$.

The algorithm for convergence is depicted in the flow diagram represented in Figure 5 that represents the iterative procedure to find a solution till the tolerance margin is achieved.

The solution algorithm can be implemented very efficiently in R, since it is very easy to operationalize. We provide the R scripts for the 3D-GRAS and 4D-GRAS methods upon request. They can be easily generalized for any dimension using the same sequence of updating factors for each dimension at once in each step.

5 | SOME FURTHER CONSIDERATIONS ON FEASIBILITY AND CONVERGENCE

The convergence of the nD-GRAS method is guaranteed as long as the optimization problem is well defined and a solution exists. As such, this is the case because the target function in (1) is a sum of strictly convex functions and hence, strictly convex. All the constraints are linear functions, and hence convex even though not strictly convex. The Lagrangian function in (2) is also a (strictly) convex functions since it is a sum of strictly convex and convex functions. Hence, our algorithm will converge to this solution if it exists. Besides, if a solution exists, given the characteristics of the optimization function and the constraints, according to Chiang (1984), this solution is unique.[9]

One of the main advantages of using analytical solutions and simple iterative algorithms is the possibility of controlling for problem resolution and therefore, fixing potential infeasibilities. As a result, there are some basic necessary conditions concerning feasibility that are important to highlight. These are presented as follows.

First, it is important to note that the constraints by dimension must add up to the same number regardless dimensions; otherwise, the problem would be infeasible. Bacharach (1970) proved[10] this necessary condition for bi-dimensional non-negative matrices. This necessary condition of Bacharach remains valid in the multidimensional generalization, since no solution would exist otherwise.

Second, the number of null elements in the prior is another important issue regarding convergence since they existence of zeros reduce the degrees of freedom of the system to find a solution. However, if zeroes happen to split the array into two independent sub-arrays, then the first constraint identified by Bacharach (1970)[11] can be applied independently for each sub-array. Bacharach (1970)[12] also provided for non-negative bi-dimensional matrices,

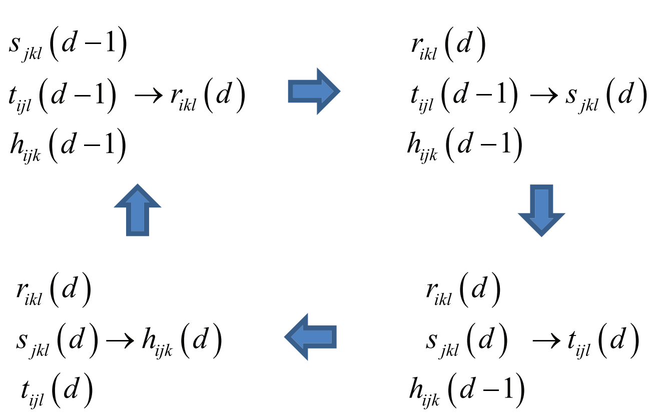

FIGURE 5 Algorithmic flow to compute updating factors in the 4D-GRAS

necessary and sufficient conditions for convergence when zeros in the prior exist. Again, this proof can be generalized for a non-negative multidimensional array. Very sparse matrices are prone to present more convergence problems if targets are not well defined.

Third, the existence of negative elements modifies the necessary conditions given by Bacharach (1970), giving the problem a sort of additional flexibility. This way, problems that were initially infeasible for non-negative arrays become feasible if negative numbers appear. As far as we know, we have not seen theorems for the convergence of general matrices with positive and negative elements.

On the contrary, at the same time, the existence of negatives leave space to new potential problems concerning the existence of a solution to the optimization problem and, consequently, the convergence of the algorithm. These problems are usually linked to sign shifts: either, sign shifts in totals, or sign shifts in individual elements between the prior and the targets.

A qualitative exploration of the prior matrix and targets is a good practice to detect potential sources of infeasibilities such as null vectors with non-zero targets or negative (resp. positive) vectors with positive (resp. negative) target. In the first situation, when all the elements of a vector[13] are zero in the prior and the sum target a value different to zero, the infeasibility arises since the nD-GRAS is a multiplicative and sign-preserving method, and zero elements in the prior will remain zero after the update. In the second case, we need to use a non-sign preserving alternative like the one described in Lenzen et al. (2014).

It is also important to check vectors that will be set to zero after the first iteration. This is the case when a target is zero and all the elements are either null and positive, or null and negative. In these situations all the elements in this vector will be set to zero after the first iteration, modifying the qualitative structure of the array in other transversal vectors that may lead to additional infeasibilities as those described in the previous paragraph.

To our knowledge, beyond the above-mentioned remarks, for general matrices that contain positive and negative elements, there are no references in the literature addressing the feasibility, existence and uniqueness of a solution of this optimization problem.

Another interesting subject is checking the amount of information required in terms of number of elements in the constraints (i.e., without taking into account the prior), in order to perform an optimization problem of this kind and some of its consequences.

In a bi-dimensional problem, if the matrix has m_1 rows and m_2 columns, $m_1 \times m_2$ elements have to be estimated while only $m_1 + m_2$ additional pieces of information are required. For a 3D problem, the 3D array consists of $m_1 \times m_2 \times m_3$ elements, and our pieces of information are the three target matrices, with $m_1 \times m_2$, $m_1 \times m_3$, $m_2 \times m_3$ elements each.

In the generalized n-dimensional problem, the multidimensional array to be estimated has $\prod_{i=1}^{n} m_i$ elements and we need n arrays of dimension $n-1$ of aditional information. This set of target arrays accrue for $\sum_{i=1}^{n} \left(\prod_{\substack{j=1 \\ j \neq i}}^{n} m_j \right)$ elements in total. Note that this number of elements equals the number of vectors inside the array in our nD-GRAS problem, and the number of factors to be estimated, and this is directly connected to the numbers of operations involved and the required time in computing a solution using the algorithm.

We can also compute the ratio of information needed for a complete nD-GRAS, as the number of factors over the number of elements to be estimated, that is, $\frac{\sum_{i=1}^{n} \left(\prod_{\substack{j=1 \\ j \neq i}}^{n} m_j \right)}{\prod_{i=1}^{n} m_i} = \sum_{i=1}^{n} \frac{1}{m_i}$.

If all the dimensions had the same number of elements, that is, $m_i = m \forall i$, the ratio of available information simplifies to $\frac{n}{m}$, namely, the dimension of the problem over the number of elements in every dimension. The sparsity of the matrix also affects this ratio, since the number of null elements reduces the degrees of freedom, i.e. the number of coefficients to be estimated. In this case, the ratio would be $\frac{\sum_{i=1}^{n}\left(\prod_{\substack{j=1 \\ j\neq i}}^{n} m_j\right)}{-z+\prod_{i=1}^{n} m_i}$ where z is the number of null elements in the prior array.

This ratio can be connected to feasibility issues. For instance, if it is equal to one (as it would be the easiest case of a 2×2 matrix) our problem is initially determined and it could be solved directly as a system of linear equations, as long as the targets are not ill-posed leading to an inconsistent problem. If the ratio were higher than one, we would have an overdetermined problem that could also be infeasible if targets are ill-posed. The closer to one this ratio, implies that our problem is more constrained, either because the sparsity of the matrix is very high or because the number of dimension is potentially high for the number of elements across all the dimensions. These situations are more prone to generate potential infeasibilities because of incompatibilities between the prior and targets.

6 | PRACTICAL APPLICATIONS OF 3D-GRAS AND 4D-GRAS METHODS

In this section, we show two examples of the nD-GRAS method for three and four dimensions. The first one is a 3D-GRAS for the estimation of valuation matrices in Denmark. And the second one, is a 4D-GRAS example for illustration purposes where we estimate the 2015 OECD Intercountry input–output (OECD-ICIO) table (oecd\icio).

6.1 | A 3D-GRAS method for the estimation of valuation matrices and use table at basic prices in Denmark

We show here how the 3D-GRAS method is used for the estimation of the 2015 Danish "use table at basic prices" (Ubp), 'table of trade and transport margins" (TTM) and "table of taxes less subsidies on products" (TLS), consistent with 2015 "use table at purchasers' prices" (Upp), all of them at current prices.

Although national statistical offices typically have all the information for the calculation of trade and transport margins and taxes less subsidies on products by product (in the format of column vectors) and therefore, output at basic prices, they often do not have the details for splitting them into fully-fledged matrices. At most, they estimate various distributive trade channels with specific margin rates for wholesalers and retailers (Rueda-Cantuche et al., 2006). At this point, it is of utmost importance to make such breakdown across the users of a use table so that there would be a full consistency between Ubp, TTM, TLS and Upp. Besides, MRIO compilers will surely require the use of the 3D-GRAS method to produce the corresponding missing use tables at basic prices.

For Denmark, the official 2010 observed tables will be our prior. Source data for 2015 and 2010 have been directly downloaded from the Eurostat website. Regarding the dimensions involved, we have $N_1 = 65$ products, $N_2 = 72$ activities, 65 industries plus a split of seven final users; and $N_3 = 3$ layers in our three-dimensional arrays, corresponding to Ubp, TTM and TLS tables. The 2010 Ubp, TTM and TLS tables will be our priors.

We have calculated the required targets from the real observed tables of 2015. In fact, we need the following matrices as targets:

- Use table at purchasers' prices of 2015.
- Basic prices use totals, trade and transport margins totals and taxes less subsidies totals by product.
- Basis prices totals, trade and transport margins totals and taxes less subsidies totals by intermediate and final users.

This set-up is equivalent to the one described in Figure 2. Our goal is to estimate the three internal layers of the cube for 2015 using 2010 tables as priors, which will require the calculation of $(65\times72)+(65\times3)+(3\times72)=5091$ updating factors. The information ration of this example initially is $\frac{1}{3}+\frac{1}{65}+\frac{1}{72}=36.3\%$, but since there are 6.347 null elements in our prior, the ratio turns 66.2%.

An important feature of the balancing problem that we want to highlight concerns trade and transport margins by users. Summing trade and transport margins over all the products by user is always zero by definition. The rationale behind this is that the amount of trade and transport margins included in all products has to be reallocated to trade and transport services. The GRAS benchmarking procedure presents the following property: if a target value is zero, the vector involved will turn zero in the iterative procedure when all the elements are only positive or negative. However, if positive and negative elements coexist, the GRAS benchmarking algorithm transforms both positive and negative elements respecting the zero-sum target. In other similar situations in an economic context, when a total is to be zero, it is illogical to think that positive and negative elements adding up to that total would be compensating. It may seem logical to require the annulation of all the elements that add up to zero. Trade and transport margins provide a good example of a zero-sum vector where its elements, far from being turned zero, need to compensate each other in the updating process.

The 2010 tables have been checked to detect and remove direct infeasibilities of the prior with respect to the targets, trying to set up reliable prior structures of the vectors affected by these direct infeasibilities. Comparing the prior with respect to the 2015 target tables, there are 34 sign shifts and 245 elements that change from zero to non-zero, or vice versa. The modifications in the prior to remove infeasibilities have been very scarce, as we can see in Table 1, accounting for less than 0.5% of the elements in the prior. The results after performing the 3D-GRAS method are summarized in the following tables.

TABLE 1 Composition of 2010 prior and 2015 target matrices

# elements		Use_{bp} table		TTM table		TLS table		Global	
2010	0	835	17.8%	3110	66.5%	2423	51.8%	6368	45.4%
	>0	3822	81.7%	1350	28.8%	2035	43.5%	7207	51.3%
	<0	23	0.5%	220	4.7%	222	4.7%	465	3.3%
2010 modified	0	821	17.5%	3108	66.4%	2418	51.7%	6347	45.2%
	>0	3842	82.1%	1360	29.1%	2044	43.7%	7246	51.6%
	<0	17	0.4%	212	4.5%	218	4.7%	447	3.2%
2015	0	835	17.9%	3106	66.4%	2417	51.6%	6358	45.3%
	>0	3825	81.9%	1366	29.2%	2089	44.6%	7280	51.9%
	<0	10	0.2%	208	4.4%	174	3.7%	392	2.8%

TABLE 2 Error measures for 3D-GRAS and 2D-GRAS estimations

	3D-GRAS estimation				Independent 2D-GRAS estimations			
	WAPE	MAPE	SWAPE	SMAPE	WAPE	MAPE	SWAPE	SMAPE
Full 3D hypermatrix	3.2%	52.2%	3.5%	33.1%	9.8%	34.6%	10.5%	24.0%
Basic prices layer	1.8%	4.4%	2.0%	3.9%	9.8%	38.9%	10.5%	29.3%
TTM layer	9.8%	20.4%	10.7%	17.5%	10.4%	16.6%	11.3%	14.0%
TLS layer	9.7%	131.9%	10.2%	78.0%	8.3%	48.4%	8.7%	28.5%
Upper theoretical bound	-	-	200%	200%	-	-	200%	200%

The left part of Table 2 shows error measures for the 3D-GRAS estimation;[14] first for the whole 3D array and then, as calculated individually for each 2D layer. It is important to note that it makes no sense to match the estimated values with respect to the 2015 use table at purchasers´ prices matrix, since this a constraint of the problem and, given that our problem has converged, this total is met by construction.

In order to assess the performance of the 3D-GRAS estimation, we also conducted three independent standard 2D-GRAS projections for each of the Ubp, TTM and TLS tables. The error measures of these estimations can also be found on the right hand side of Table 2.

A global view of Table 2 shows that results of 3D-GRAS estimation are overall very good. The weighted average percentage error (WAPE) is only 3.24%. This value increases until nearly 10% when we compare the results of the full 3D array resulting from joining the layers of the 2D-GRAS independent estimations. Besides, we must bear in mind that the 2D-GRAS estimates of Ubp, TTM and TLS tables do not add up to the Upp values.

The simple average of the relative errors (MAPE and SMAPE) indicates that some important relative errors in the 3D-GRAS estimation exist, compared their 2D-GRAS counterparts. However, in general, at the same time the WAPE and SWAPE inform us that those are concentrated in tiny and negligible values as these error measures are lower if compared to the 2D-GRAS counterparts.[15] These features indicate that the 3D-GRAS produces a better estimate in general, but, at the same time seems to produce a larger number of high relative errors concentrated in very tiny values. We can confirm this in Figure 6[16] and Table 3, where we see the error distributions for each layer in the 3D-GRAS. In the case of Ubp table, the results of the 3D-GRAS estimations clearly improve the results of the 2D-GRAS method. We can also see in this figure that the errors distribution are largely overlapping for the TTM and TLS tables but slightly better off the 3D-GRAS estimations. Distribution of relative errors of 3D-GRAS is more skewed to 0 compared to the 2D-GRAS. The median error is lower and if we look at Table 3 that provides the frequency distribution of relative errors, we can clearly see that the relative error distribution of 3D-GRAS estimates are better off that those the 2D-GRAS methods.

Hence, the main features to emphasize here are threefold. First, the use of a 3D-GRAS method ensures the consistency of the results with the Upp. The projections resulting from the three independent 2D-GRAS are not consistent with it. Besides, the individual error measures are also generally better for the tables in the 3D-GsRAS estimation. The only minor drawback seems to be the larger number of highest relative errors concentrated in small coefficients of the 3D-GRAS estimates.

This behaviour will also appear in the 4D-GRAS example, as we will see. In this example, the highest relative errors are largely concentrated in the Inventories column and TLS table in general. Inventories are characterized by a high volatility in their values and, together with the TLS elements, concentrate most of the sign changes in cells

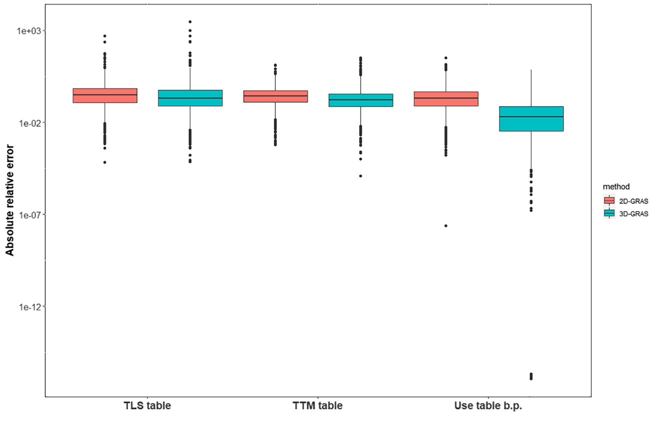

FIGURE 6 Element-wise absolute relative errors (%) distribution for Ubp, TTM and TLS matrices

TABLE 3 Relative errors element wise in Ubp, TTM and TLS tables for 3D and 2D estimations

	Usebp table				TTM table				TLS table			
	3D projection		2D projection		3D projection		2D projection		3D projection		2D projection	
Relative error	freq	%	freq	%	freq	%	freq	%	freq	%	freq	%
0% [a]	821	17.5%	821	17.5%	3,093	66.1%	3,093	66.1%	2,321	48.3%	2,321	49.6%
Lower to 0.01%	1,109	23.7%	2	0.0%	4	0.1%	0	0.0%	5	0.1%	1	0.0%
0.01%–1%	970	20.7%	119	2.5%	60	1.3%	34	0.7%	91	1.9%	52	1.1%
1%–5%	821	17.5%	478	10.2%	222	4.7%	139	3.0%	295	6.3%	208	4.4%
5%–20%	725	15.5%	1,314	28.1%	589	12.6%	486	10.4%	750	16.0%	595	12.7%
20%–50%	190	4.1%	1,045	22.3%	458	9.8%	520	11.1%	518	11.1%	690	14.7%
50%–100%	27	0.6%	507	10.8%	139	3.0%	266	5.7%	571	12.2%	633	13.5%
100%-10,000%	17	0.4%	394	8.4%	114	2.4%	142	3.0%	124	2.6%	178	3.8%
Greater than 10,000%	0	0.0%	0	0.0%	1	0.0%	0	0.0%	5	0.1%	2	0.0%

Note:
[a]Here we only account for null elements in 2010 and 2015 tables, i.e. zero in the prior and the target tables.

between 2010 and 2015. The sign preservation property of the GRAS method makes it very difficult to provide good fits under these circumstances

All in all, the additional constraint in the 3D-GRAS problem that forces the Ubp, TTM and TLS tables to add up to the Upp plays a central role in this better performance and greater accuracy for two main reasons in our opinion. First, the use of these 3D constraints is a piece of additional available information in the optimization problem. Second, this additional piece of information links all the updated tables in a single estimation process that restricts cells variation, thus avoiding more extreme solutions and increasing the global goodness of fit.

6.2 | A 4D-GRAS method for the estimation of the ICIO table of the OECD

In this subsection, we introduce an example of application of the 4D-GRAS method to estimate the 2015 OECD-ICIO. Data and all the relevant information for the interpretation of this dataset is available at oe.cd/icio.

We have considered 65 available geographical areas (64 OECD countries plus a rest of the world area) as origin and destination areas of the domestic production of these countries.[17] The OECD-ICIO table is an industry by industry input–output table. In order to avoid ambiguity, we will denominate as "products" inputs coming from different industries. Products are split into 36 categories plus an additional category for taxes less subsidies. Users are split in 36 intermediate demand industries plus six final demand categories. Hence, Origin (65 geographical areas), destination (65 geographical areas), product (37 products aggregation plus TLS row) and user (42 different users) are the four dimensions considered in our illustration. The entire table consists of 6,565,650 elements. Among them, we find 1,028,538 elements that are equal to zero (15.7% of the total), 21,976 negative elements (0.3%) and the rest positive. For this example, the number of updating factors to be estimated is $(65\times65\times37)+(65\times65\times42)+2\times(65\times37\times42)=535795$, and the information ratio is initially $\frac{1}{65}+\frac{1}{65}+\frac{1}{42}+\frac{1}{35}=8.2\%$. However, the number of null elements in the prior increases this ratio to 9.7%.

As we mentioned in Section 3, the four 3D arrays that work as constraints would be the result of the aggregation of the 4D table across each one of the four dimensions:

1. $\mathbf{A}_1$: Origin and destination of product, summed across users (balanced view of trade in goods and services) — (ODP table); country's exports of products to trading partners for all users. For instance, Belgian exports of chocolate to Germany or German imports of chocolate from Belgium.
2. $\mathbf{A}_2$: Origin and destination by user, summed across products—(ODU table); total exports of countries to specific users and trading partners, for instance, total Belgian exports to the German food industry or alternatively, total imports of Belgian products by the German food industry.
3. $\mathbf{A}_3$: Use table by country of origin, summed across destination countries—(OPU table); country's exports of products to specific users worldwide. For instance, Belgian exports of nuts to the food industry worldwide or alternatively, world food industry's imports of nuts from Belgium.
4. $\mathbf{A}_4$: Use table by importing country, summed across countries of origin— (DPU table); national use tables of imports. For instance, Belgian food industry's total imports of nuts or alternatively, total world exports of nuts to the Belgian food industry.

It might look like that data on A_1 and A_4 could be the only ones really available but in practice, trade statistics, business statistics and specific surveys for input–output tables alike can provide data on the other two layers, too. Indeed, business surveys or other alternative national sources can provide data, for instance, about how much the German food industry imports from Belgium (ODU) and how much of the world imports of nuts are produced/ exported from Belgium (OPU). Moreover, the implementation of the 4D-GRAS method does not require having full information on the corresponding 3D layers. Even if this is partial, with more degrees of freedom and less number of constraints, a solution will also exist and converge with a better performance than for individual and separate estimations of the four different layers. Nevertheless, in the total absence of information on any layer, we believe the 4D-GRAS method is still useful as it was the Leontief and Strout (1963) paper on multiregional input–output analysis when no data of this type existed by then.

We have used a random realization of the 2015 OECD-ICIO table as a prior for the 4D-GRAS optimization problem. The elements of this random prior were calculated as $v^{*}_{prior} = v_{2015} \cdot (1+r)$ with $r \in U(-0.5, 0.5)$. We opted for this simplification for several reasons. First, the OECD table provides a lot of detail on the bilateral transactions on a global scale. This makes the figures in the table very uneven, ranging from an order of magnitude of 1E+06 to largely below 1E-08[18]. If we opt to round figures to a reasonable threshold, the matrix becomes very sparse and the number of potential infeasibilities grows exponentially. These facts, together with the number of interventions required to fix the prior, led us to opt for a more transparent solution for illustration purposes.

With this random prior and the target calculated from the 2015 OECD-ICIO table, we set a convergence threshold of 1E-05 in Equation 7. The maximum absolute error in the target constraints is 144.7 units and represents only 0.01% of the desired target. The maximum relative error in the target constraints is only 0.6% and represents an absolute error of 1.8E-04 units.

In order to assess the performance of the 4D-GRAS method, we calculate the goodness of fit measures of the estimated ICIOT using the 4D-GRAS. They are summarized in Table 4. In a similar way to our 3D-GRAS example, we have also the errors of our 4D-GRAS estimations for each of the three-dimensional "slices" of the 4D array achieved by independent 3D-GRAS. It is important to note that there are 209 three-dimensional "slices" in our 4D array, arranged by:

- 65 different slices according to the Origin country dimension (DPU tables by origin);
- 65 different slices according to the Destination country dimension (OPU tables by destination);
- 37 different slices according to the Product dimension (ODU tables by product); and
- 42 different slices according to the User dimension (ODP tables by user).

Given this multiplicity, Table 4 provides averages and standard deviations for error measures for each group of slices.

The results show similar findings to those highlighted in the 3D-GRAS methods. First, the 4D-GRAS method achieve a satisfactory goodness of fit for the full OECD-ICIO table estimation. The WAPE and SWAPE of the full OECD-ICIO table is around 2.5%. These results improve the goodness of fit the full OECD-ICIO table if we join the

TABLE 4 Error measures for the estimated 2015 OECD-ICIO table

		4D-GRAS estimations				3D-GRAS estimations			
		WAPE	MAPE	SWAPE	SMAPE	WAPE	MAPE	SWAPE	SMAPE
Full OECD_ICIOT 2015		2.54%	31.41%	2.55%	31.73%	3.32%	29.89%	3.33%	30.24%
						0.2%	0.7%	0.2%	0.7%
ODP tables by user	Average	3.36%	32.18%	3.37%	32.51%	3.71%	29.55%	3.72%	29.90%
	St.Dev.	1.5%	2.5%	1.5%	2.5%	2.5%	1.4%	2.5%	1.4%
ODU tables by product	Average	2.99%	31.41%	3.00%	31.73%	3.60%	28.88%	3.61%	29.25%
	St.Dev.	2.4%	6.3%	2.4%	6.3%	2.9%	5.8%	2.9%	5.8%
OPU tables by destination	Average	4.16%	31.41%	4.16%	31.73%	5.29%	30.55%	5.29%	30.88%
	St.Dev.	1.6%	1.3%	1.6%	1.3%	1.7%	1.6%	1.7%	1.6%
DPU tables by origin	Average	4.12%	31.41%	4.13%	31.73%	5.05%	30.58%	5.06%	30.93%
	St.Dev.	1.9%	4.1%	1.9%	4.2%	2.1%	4.2%	2.1%	4.3%

results of the independent 3D-GRAS estimations into a full OECD-ICIO estimation. In addition, the 4D-GRAS estimation of the full OECD-ICIO estimation ensures that the sum over any the four dimensions matches the observed 3D projections of the OECD-ICIO.

We can also compare the goodness of fit of each of the 3D slices resulting from the 4D-GRAS estimation with their counterparts obtained with the independent 3D-GRAS estimations. This is summarized in Table 5 grouped by a kind of 3D projection. On average, the 4D-GRAS method results again show a higher accuracy compared to the results of the 3D-GRAS methods. In overall, in 86% of the situations the 3D 'slices' estimated jointly in a 4D-GRAS method are better off than their equivalent estimations using independent 3D-GRAS.

Again, as it happened for the 3D-GRAS method compared with respect to their lower dimension counterparts, using a 4D-GRAS method not only ensures the coherence of the results in the fourth dimension, it also leads to overall better estimation than using 3D-GRAS. The 3D-GRAS only outperforms the results achieved in the 4D-GRAS in 29 situations out of 209. For ODP and DPU tables, no 3D-GRAS projection improves the ones achieved in the 4D-GRAS. This result is more ambiguous for ODU and OPU tables but, in general, the 4D-GRAS projections generally outperforms their equivalent 3D-GRAS counterparts.

However, in those cases where the 3D-GRAS methods outperform the 4D-GRAS results, the difference is minimum as we can see in Figure 7, where we represent the absolute relative error distribution for some selected cases of ODP and ODU. In the left-hand side, we show some of the "slices" with a better goodness of fit for 3D-GRAS methods in ODP and ODU tables. On the right-hand side, we have the same but for 4D-GRAS methods.

TABLE 5 Assessment of 4D-GRAS projections outperforming their equivalent 3D projections

	SWAPE		WAPE	
ODP tables (42)	22	52.4%	22	52.4%
ODU tables (37)	28	75.7%	28	75.7%
OPU tables (65)	65	100.0%	65	100.0%
DPU tables (65)	65	100.0%	65	100.0%
Total (209)	180	86.1%	180	86.1%

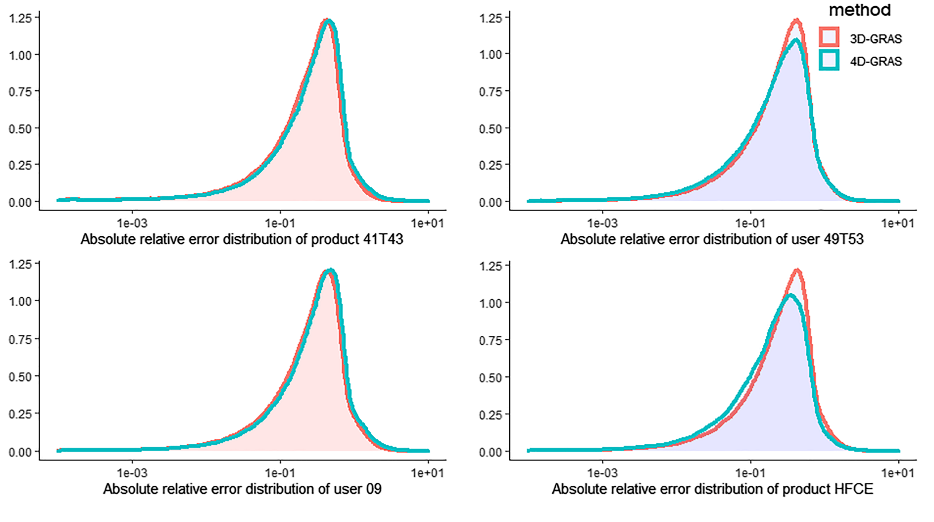

FIGURE 7 Absolute relative errors' distributions

In the upper left-hand side, we see the empirical density estimations of the best 3D-GRAS performance in ODU tables. This is the ODU table for product "41 T43." We can see that the distribution of relative errors in the 3D-GRAS is consistently closer to the origin. Nonetheless, both distributions are very close and the differences in errors are very small. On the upper right-hand side we have the distribution for the best 4D-GRAS option for a ODU table, for product "49 T53" in which the situation is quite the opposite. In this case, we clearly see that errors of 4D-GRAS are consistently smaller than for the 3D-GRAS method. The same applies for the two examples of ODP tables, for user "09"and user "HFCE," in the bottom part of Figure 7.

Finally, as it happened in the example of the 3D-GRAS Danish estimation, the proportion of large relative errors in tiny and negligible elements of the OECD-ICIO table is larger in the 4D-GRAS estimation compared to the 3D-GRAS method. This can be seen from the direct comparison of MAPE and SMAPE of the 4D-GRAS method with respect to the 3D-GRAS in Table 4. While, WAPE and SWAPE are smaller in the 4D-GRAS estimations, on MAPE and SMAPE is the opposite, which indicates that the 4D-GRAS produces a better estimate in general, but, seems again to be producing a larger number of high relative errors concentrated in very tiny values. This situation is described in Figure 8.

In Figure 8, the x-axis represents the absolute value of the element, distinguishing between positive and negative values; the y-axis represents the absolute relative error in percentage terms. Both axis are on a logarithmic scale. In Figure 8, it is easy to see a pattern of steady decrease of relative errors as the size of the element increases; this decline is more intense when this absolute value increases.

In general, it is natural to think that the largest values would concentrate the smallest relative errors and big outliers are prone to occur among the coefficients with tiniest values and/or sign shifts. The multiplicative nature of GRAS balancing techniques may seem to neutralize this effect, since the size of the coefficient in the prior does not affect the relative change of the coefficient that will depend on the size of the updating factors and the value of that coefficient in the target matrix. Nonetheless, we have appreciated in these two examples introduced in this section that larger dimensional GRAS generate a larger proportion of high relative errors concentrated in coefficients with the tiniest values.

To sum up, the conclusions drawn from these two example of the application of the multidimensional GRAS are parallel and congruent:

- If possible, it is a better option to use a higher dimensional approach since it ensures the global consistency because of the joint estimation in a larger dimension.

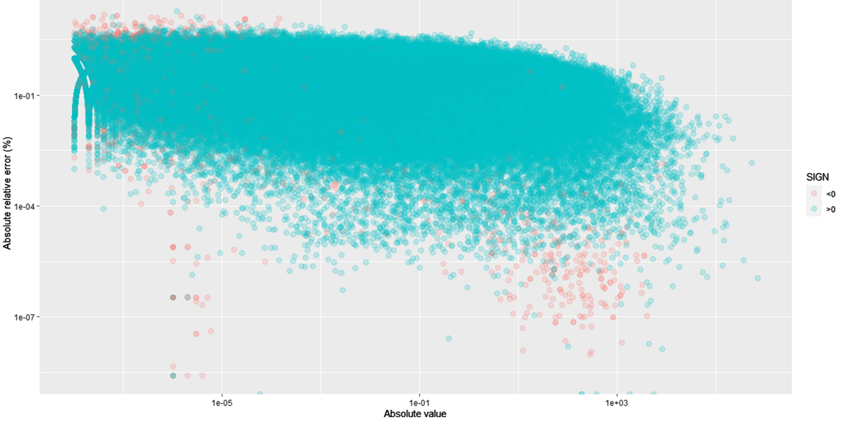

FIGURE 8 Distribution of relative errors by size of the element

- Besides, in overall, it also improves the estimation accuracy of the whole problem and of the sub-pieces, if compared with estimations done with lower dimension methods.
- Higher dimension methods, despite ensuring a better global goodness of fit, seem to generate a large proportion of higher relative errors mostly concentrated in tiny and negligible values.

7 | CONCLUSIONS

This paper generalizes the GRAS method in a multidimensional framework and provides its analytical closed-form[19] solution. We have also described an intuitive iterative algorithm that provides a handy resource-efficient operationalization of the multidimensional GRAS method, and some conditions to trace, detect and solve infeasibilities and convergence problems.

The multidimensional GRAS methods can be useful for practical applications in, broadly speaking, input–output analyses, especially when dealing with multiregional frameworks. The 3D-GRAS method can be used directly for estimating annual use tables at basic prices from an existing use table at purchasers´ prices, including the estimations of the corresponding valuation matrices (i.e., trade and transport margins and taxes less subsidies on products tables). In multiregional frameworks with information about exports and imports by trading partner, the 3D-GRAS method can also be used to split the use table of imports of a specific importer by country of origin. The 4D-GRAS method is also useful for the projection of complete multiregional IOTs, with products, users, countries of origin and countries of destination being the corresponding four dimensions. We have presented two examples that illustrate two applications for Denmark (3D-GRAS) and for the OECD inter-country IO Tables (4D-GRAS). Both applications are based on real problems being faced by MRIO compilers one way or another, although the required data to implement them might sometimes be difficult to find. In any case, although in the absence of information, the theoretical method is still valid as it was, for instance, the multiregional input–output theoretical model published by Leontief and Strout (1963) when such MRIO tables did not exist then.

We have also shown that dealing with a higher number of dimensions, apart from ensuring consistency, usually leads to a better performance than using independent estimations of the corresponding slices, with a lower number of dimensions. It is important to raise awareness about some features that are relevant for empirical applications. First, it is necessary to check for existing infeasibilities between the prior and the target constraints. In the event that those infeasibilities are not *a priori* corrected, our problem will remain infeasible and the algorithm will not provide a solution. These infeasibilities stem from the sign-preservation feature of GRAS methods and, in particular, concerns the sparsity of the prior, sign shifts and lack of match of zero and non-zero elements between priors and targets. Second, the higher dimension methods generate a larger amount of relative errors especially in tiny and negligible elements of our prior, and in elements with sign shifts between priors and targets. The KRAS method is an existing alternative for dealing automatically with infeasibilities and with many other features. However, as long as a feasible solution exists—or when we have expert information that can help us to remove infeasibilities in a manual way—our approach is easy to operationalize without the need for significant resource intensive IT infrastructures.

In sum, the multidimensional GRAS methods developed in this paper are successful for balancing multiregional IOTs with available information about the margin totals of each dimension and for balancing a full supply and use framework from purchasers´ prices to basic prices, including their valuation matrices. Moreover, they are also suitable to allow for further extensions up to as many dimensions as required, for example, multiregional IOTs split according to the domestic and foreign ownership of multinational enterprises (in extended supply and use Tables). That is, this multidimensional balancing situations can be addressed and solved with a 5D-GRAS algorithm where the fifth dimension would be that the sum of the two columns depicting domestic-owned firms' and foreign-owned firms' input structures for every industry must match the total aggregated input structure of that industry as shown in the original ICIO. However, this is clearly beyond the scope of this paper.

SAMPLE CREDIT AUTHOR STATEMENT

Juan Manuel Valderas-Jaramillo: Conceptualization, Methodology, Investigation, Software, Data curation, Formal Analysis, Writing – Original draft, Writing – Review and editing, Supervision. *José Manuel Rueda-Cantuche:* Validation, Formal Analysis, Investigation, Writing – Review and editing.

ORCID

Juan Manuel Valderas-Jaramillo https://orcid.org/0000-0003-2582-6477

ENDNOTES

[1] https://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.20/2015/July/Item_5_UNECE_NADIM.pdf

[2] Contingency tables are by definition non-negative tables since they are representations of frequency distributions or discrete joint probability distributions of several variables.

[3] Deming (1943) used least squares objective functions instead of a maximum entropy objective function.

[4] PIRAMID stands for Platform to Integrate, Reconcile and Align Model-based Input–Output Data.

[5] FIGARO stands for Full International and Global Accounts for Research in input–Output analysis.

[6] However, this approach is not enough for disaggregating national tables into full subnational tables. If we use this approach, the subnational tables coefficients will add up to the national one. However, domestic values of these subnational tables estimated with a 3D-GRAS would also be including the interregional trade coming from other regions and it would not be a real "domestic" coefficient. We would need to generalize this problem into a method that falls between a 3D-GRAS and a 4D-GRAS. This issue is explicitly mentioned and treated in Valderas-Jaramillo, et al. (2019).

[7] Our generalization of the GRAS method for higher dimensions considers that constraints are arrays in a lower dimension to the problem posed. However, this is not the only way to consider the constraints. De Mesnard (2020) poses a 3D-RAS problem where restrictions are the axes of the cube, instead of the faces. The solution is also a tri-dimensional method. This approach can also be extended to a 3D-GRAS with no difficulties. For a 4D array, it is also possible to pose the constraints in different ways to the 3D arrays. However, these approaches go beyond the scope of this paper and are not considered here.

[8] See Huang et al. (2008), Section 4 for more details.

[9] Chiang (1984 p. 342, Theorems I-III).

[10] Bacharach (1970, p. 47. Theorem I).

[11] This idea stems directly from the concept of 'connectedness' described in Bacharach (1970), p. 44).

[12] Bacharach (1970, p. 51, Theorem III).

[13] When we talk about a vector inside a multidimensional array, we mean a vector composed of all the elements taken across one of the dimensions in the multidimensional array. For instance, in a bi-dimensional matrix we can find row and column vectors. In a tridimensional array apart from row and column vectors in any of the 2D slices, there are also vectors ranging across the third dimension.

[14] These are standard error measures in this context. For a detailed explanation of all these measures and their interpretation see Valderas-Jaramillo, Rueda-Cantuche, et al. (2019)

[15] In fact, five elements in the TLS table and one observation in the TTM table have a large influence in the MAPE as they account for a relative error over 10,000% concentrated in negligible values.

[16] Due to the logarithmic scale used for the representation of the error—necessary to represent the highest errors—zero errors are not present in the Boxplot diagram. The complete frequency distribution of errors can be found in Table 6.

[17] The OECD-ICIO table splits Mexico and China into two sub-tables accounting for export processing activities and non-export processing activities. For the sake of simplicity, we have decided to aggregate these areas and work with the tables for each country as whole without distinctions.

[18] Values in the 2015 OECD-ICIO range between a maximum 2.347.855 and a minimum of −9.019.8 millions of US$. The lowest reasonable threshold from an economic point of view would be 1E-08 in absolute value (cents of dollars), as data are expressed in current million US$. However, there are around 40.000 figures falling below this threshold in the 2015 OECD-ICIO.

[19] For the 2D-GRAS and the 3D-GRAS methods, the main related references in the literature were also included for the sake of completeness.

REFERENCES

Bacharach, M. (1965). Estimating nonnegative matrices from marginal data. *International Economic Review*, *6*(3), 294–310.

Bacharach, M. (1970). *Biproportional matrices and input–output change*. Cambridge University Press.

Barthelemy, J., Suese, T., & Namazi-Rad, M. (2018). mipfp: Multidimensional iterative proportional fitting and alternative models. R-package version 3.2.1. URL: https://cran.r-project.org/web/packages/mipfp/mipfp.pdf

Bregman, L. M. (1967). Proof of convergence of Sheleikhovskii's method for a problem with transportation constraints. *USSR Computational Mathematics and Mathematical Physics*, *1*, 191–204.

Chiang, A. C. (1984). *Fundamentals methods of mathematical economics* (3rd ed.). McGraw-Hill.

Dalgaard, E., & Gysting, C. (2004). An algorithm for balancing commodity-flow systems. *Economic Systems Research*, *16*(2), 169–190. https://doi.org/10.1080/0953531042000219295

Darroch, J. N. (1962). Interactions in multi-factor contingency tables. *Journal of the Royal Statistical Society*, *Series B*, *24*(1), 251–263.

de Mesnard, L. (1994). "Unicity of biproportion", SIAM Journal on Matrix. *Analysis and Applications*, *15*(2), 490–495.

de Mesnard L. (2020). Holy-Safr 's multidimensional RAS method is not unique. DOI: https://doi.org/10.13140/RG.2.2.14584.88323. Available on researchGate.

Deming, W. E. (1943). *Statistical adjustment of data*. Wiley.

Deming, W. E., & Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, *11*(4), 427–444. https://doi.org/10.1214/aoms/1177731829

Friedlander, D. (1961). A technique for estimating a contingency table, given the marginal total and some supplementary data. *Journal of the Royal Statistical Society*, *Series a*, *124*, 412–420.

Geschke, A., & Hadjikakou, M. (2017). Virtual laboratories and MRIO analysis—an introduction. *Economic Systems Research*, *29*(2), 143–157.

Geschke, A., Lenzen, M., Kanemoto, K., & Moran, D. (2011). AISHA: A tool to construct a series of contingency tables, *19th International input–output Conference*, Alexandria, United States: International input–output Association (IIOA).

Geschke, A., Ugon, J., Lenzen, M., Kanemoto, K., & Moran, D. D. (2019). Balancing and reconciling large multi-regional input–output databases using parallel optimisation and high-performance computing. *Journal of Economic Structures*, *Pan-Pacific Association of input–output Studies (PAPAIOS)*, *8*(1), 1–24.

Gilchrist, D., & St. Louis, L. (1999). Completing input–output tables using partial information, with an application to Canadian data. *Economic Systems Research*, *11*(2), 185–194.

Gilchrist, D., & St. Louis, L. (2004). An algorithm for the consistent inclusion of partial information in the revision of input–output tables. *Economic Systems Research*, *16*(2), 149–156.

Günlük-Senesen, G., & Bates, J. M. (1988). Some experiments with methods of adjusting unbalanced data matrices. *Journal of the Royal Statistical Society*, *Series a*, *151*(3), 473–490.

Holý and Šafr. (2020). Disaggregating input–output tables by the Mmultidimensional RAS method, arXiv:1704.07814 [stat. AP].

Huang, W., Kobayashi, S., & Tanji, H. (2008). Updating an input–output matrix with sign-preservation: Some improved objective functions and their solutions. *Economic Systems Research*, *20*, 111–123.

Ireland, C. T., & Kullback, S. (1968). Contingency tables with given marginals. *Biometrika*, *55*(1), 179–188.

Jackson, R., & Murray, A. (2004). Alternative input–output matrix updating formulations. *Economic Systems Research*, *16*(2), 135–148.

Junius, T., & Oosterhaven, J. (2003). The solution of updating or regionalizing a matrix with both positive and negatives entries. *Economic Systems Research*, *15*, 87–96.

Lahr, M., & de Mesnard, L. (2004). Biproportional techniques in input–output analysis: Table updating and structural analysis. *Economic Systems Research*, *16*(2), 115–134.

Lecomber, J. R. C. (1975). A critique of methods of adjusting updating and projecting matrices. In R. I. G. Allen & W. F. Gossling (Eds.), *Estimating and Projecting input–output Coefficients* (pp. 43–56). Input–Output Publishing Company.

Lemelin, A. (2009). A GRAS variant solving for minimum information loss. *Economic Systems Research*, *21*, 399–408.

Lenzen, M., Gallego, B., & Wood, R. (2006). A flexible approach to matrix balancing under partial information. *Journal of Applied Iput–Output Analysis*, *11 & 12*, 1–24.

Lenzen, M., Gallego, B., & Wood, R. (2009). Matrix balancing under conflicting information. *Economic Systems Research*, *21*(1), 23–44.

Lenzen, M., Geschke, A., Abd Rahman, M. D., Xiao, Y., Fry, J., Reyes, J., Dietzenbacher, E., Inomata, S., Kanemoto, K., Los, B., Moran, D., Schulte in den Bäumen, H., Tukker, A., Walmsley, T., Wiedmann, T., Wood, R., & Yamano, N. (2017). The Global MRIO Lab – charting the world economy. *Economic Systems Research*, *29*(2), 158–186. https://doi.org/10.1080/09535314.2017.1301887

Lenzen, M., Geschke, A., Wiedmann, T., Lane, J., Anderson, N., Baynes, T., Boland, J., Daniels, P., Dey, C., Fry, J., Hadjikakou, M., Kenway, S., Malik, A., Moran, D., Murray, J., Nettleton, S., Poruschi, L., Reynolds, C., Rowley, H., … West, J. (2014). Compiling and using input–output frameworks through collaborative virtual laboratories. *Science of the Total Environment*, *485–486*, 241–251.

Lenzen, M., Kanemoto, K., Moran, D., & Geschke, A. (2012). Mapping the structure of the world economy. *Environmental Science and Technology*, *46*(15), 8374–8381.

Lenzen, M., Moran, D., Kanemoto, K., & Geschke, A. (2013). Building EORA: A global multi-region input–output database at high country and sector resolution. *Economic Systems Research*, *25*(1), 20–49.

Lenzen, M., Moran, D. D., Geschke, A., & Kanemoto, K. (2014). A non-sign-preserving RAS variant. *Economic Systems Research*, *26*(2), 197–208. https://doi.org/10.1080/09535314.2014.897933

Lenzen, M., Wood, R., & Gallego, B. (2007). Some comments on the GRAS method. *Economic Systems Research*, *19*(4), 461–465.

Leontief, W. (1941). *The structure of the American economy 1919–1929*. Cambridge University Press.

Leontief, W., & Strout, A. (1963). Multiregional input–output Analysis. In T. Barna (Ed.), *Structural interdependence and economic development*. Palgrave Macmillan.

Oosterhaven, J., Piek, G., & Stelder, D. (1986). Theory and practice of updating regional versus interregional interindustry tables. *Papers of the Regional Science Association*, *59*, 57–72.

Paelinck, J., & Waelbroeck, J. (1963). Etude empirique sur l'évolution de coefficients 'input–output': essai d'application de la procédure RAS de Cambridge au tableau industriel belge. *Economie Appliquee*, *16*, 81–111.

Remond-Tiedrez, I., & Rueda-Cantuche, J. M. (Eds.) (2019). *EU inter-country supply, use and input–output tables — Full international and global accounts for research in input–output analysis (FIGARO)*. European Union, Publications Office of the European Union in Luxembourg.

Rey Los Santos, L., Wojtowicz, K., Tamba, M., Vandyck, T., Weitzel, M., Saveyn, B., & Temursho, U. (2018). Global macroeconomic balances for mid-century climate analyses. Supplementary material to Global Energy and Climate Outlook 2018. JRC Technical Reports. European Commission.

Rueda-Cantuche, J. M., Titos, A., & Asensio, M. (2006). A use-side trade margins matrix for the Andalusian economy. *Journal of Applied Input–Output Analysis*, *11–12*, 121–135.

Temursho, U., Cardenete, M. A., Wojtowicz, K., Rey Los Santos, L., Weitzel, M., Vandyck, T., & Saveyn, B. (2020). *Projecting input–output tables for model baselines*". JRC Technical Report. Publications Office of the European Union, Luxembourg.

Temursho, U., Oosterhaven, J., & Cardenete, M. A. (2020). A multi-regional generalized RAS updating technique. *Spatial Economic Analysis*, 1–16. https://doi.org/10.1080/17421772.2020.1825782

Temurshoev, U., Miller, R. E., & Bouwmeester, M. C. (2013). A note on the GRAS method. *Economic Systems Research*, *25*, 342–361.

Temurshoev, U., & Timmer, M. P. (2011). Joint estimation of supply and use tables. *Papers in Regional Science*, *90*(4), 863–883.

Tilanus, C. B. (1976). The multiproportional RAS algorithm. In C. B. Tilanus (Ed.), *Quantitative methods in budgeting*. (pp. 164–165). Leiden: Martinus Nijhoff Social Sciences Division.

United Nations. (2018). *Handbook on supply, use and input-output tables with extensions and applications*. The United Nations, Series F, No. 74, Rev. 1. New York: United Nations Publication. https://unstats.un.org/unsd/nationalaccount/docs/SUT_IOT_HB_Final_Cover.pdf

Valderas-Jaramillo, J. M., Amores, A. F., Oliva Mora, J. R., & Boniquito Fernández, S. (2019). Provincialización de tablas de origen y destino SUT-RAS tridimensional (3DSUT-RAS): Desarrollo y aplicación al caso andaluz. *VIII Jornadas de Análisis input–output*, organized by the Hispanoamerican Society of input–output Analysis (SHAIO), doi: https://doi.org/10.13140/RG.2.2.25012.37763. Available on researchGate.

Valderas-Jaramillo, J. M., Rueda-Cantuche, J. M., Olmedo, E., & Beutel, J. (2019). Projecting supply and use tables: new variants and fair comparisons. *Economic Systems Research*, *31*, 423–444. https://doi.org/10.1080/09535314.2018.1545221

Zhou, P., Fan, L. W., & Zhou, D. Q. (2010). Data aggregation in constructing composite indicators: A perspective of information loss. *Expert Systems with Applications*, *37*(1), 360–365. https://doi.org/10.1016/j.eswa.2009.05.039

How to cite this article: Valderas-Jaramillo, J. M., & Rueda-Cantuche, J. M. (2021). The multidimensional nD-GRAS method: Applications for the projection of multiregional input–output frameworks and valuation matrices. *Papers in Regional Science*, *100*(6), 1599–1624. https://doi.org/10.1111/pirs.12625

DOI: 10.1111/pirs.12625

Resumen. En este artículo se presenta una generalización multidimensional del método GRAS (nD-GRAS) para la estimación de matrices múltiples en un marco integrado. Las aplicaciones potenciales de este método en los análisis *input-output* regionales y multirregionales basados en los marcos de cuentas nacionales o regionales son numerosas. Se incluyen dos aplicaciones reales, un 3D-GRAS que estima una tabla de uso a precios básicos conjuntamente con matrices de valoración para Dinamarca; y un 4D-GRAS para estimar tablas *input-output* entre países con datos de la OCDE. Se demuestra que los métodos GRAS de mayores dimensiones proporcionan estimaciones más consistentes y precisas que aquellos con un menor número de dimensiones. Para una fácil operacionalización, se proporciona la solución analítica en forma cerrada y el algoritmo tipo RAS.

抄録: 本稿では、統合フレームワークにおける複数の行列の推定のためのGRAS法(n D-GRAS)の多次元一般化モデルを示す。国別・地域別会計のフレームワークに基づく地域別・複数地域別の産業連関分析にこの手法を適用できる可能性のある方法は多くある。デンマークの評価行列と一緒に基本価格で使用表を推定する3 D-GRAS また、OECDのデータを用いて各国間の産業連関表を推定するための4 D-GRAS、以上の実際の二つの応用事例を示す。高次元のGRAS法は、低次元のGRAS法よりも、より一貫性があり正確な推定値が得られることが示された。また、解析的閉形式解と簡単な操作のためのRAS様アルゴリズムが得られた。