IOpedia Research Paper Series www.iopedia.eu

IOpedia Research Paper No. 02

A multiregional generalized RAS updating technique

Umed Temursho, Jan Oosterhaven and Manuel Alejandro Cardenete

September 2019

IOpedia

A multi-regional generalized RAS updating technique*

Umed Temursho[†a, b], Jan Oosterhaven[c] and M. Alejandro Cardenete[d]

[a]IOpedia (www.IOpedia.eu), Tomares, Seville, Spain
[b]Institute of Public Policy and Administration, Graduate School of Development, University of Central Asia, Bishkek, Kyrgyz Republic
[c]University of Groningen, The Netherlands
[d]Department of Economics, Loyola Universidad Andalucía, Seville, Spain.

September 22, 2019

Abstract

This paper presents an extension of the generalized RAS technique to a multi-regional (MR) or multi-national setting. The framework is applicable to updating/regionalizing/balancing/projecting any partitioned matrix that needs to conform the new row sums, column sums and non-overlapping aggregation constraints. The technique, which we refer to as MR-GRAS, also handles non-exhaustive constraints, in which case the missing values are endogenously recovered in the updating process. We derive the analytical solution of MR-GRAS and propose a simple iterative algorithm for its computation. Further, we discuss the main properties of the method, most of which contribute to the popularity of RAS-type balancing techniques, and we discuss normalization and interpretation of MR-GRAS multipliers. From a wide range of possible MR-GRAS applications, several updating settings, including national and global Supply and Use tables, are examined. Finally, a detailed guide on MR-GRAS implementation in practice through a worked example is presented, using our publicly available MATLAB code.

Keywords: RAS, multi-regional GRAS, partitioned matrix balancing/updating/projecting, non-survey methods

JEL Classification Codes: C02, C61, C80

*The authors would like to thank Geoffrey Hewings, Bert Saveyn, Krzysztof Wojtowicz and Luis Rey Los Santos for useful comments on earlier versions of this work.

†Corresponding author. E-mail: utemursho@gmail.com

1 Introduction

There is a large literature on updating, regionalizing, projecting, balancing and/or estimation of input-output tables (IOTs), supply and use tables (SUTs) and social accounting matrices (SAMs), which is still growing. This literature mostly considers matrix (e.g. an IOT) estimation techniques based on (non)linear programming approaches that aim at searching for a minimum "distance" between a given (i.e. available) original matrix and a new, to be estimated, matrix subject to certain constraints. However, it needs to be recognized that the very definition of "distance" could generally be considered as a rather arbitrary or subjective concept, since potentially an *infinite number of functions* can be defined and adopted as a measure of the distance between the two matrices. Nonetheless, there are a few updating and regionalization methods that are more popular among practitioners due to their attractive theoretical properties and their appeal in practical applications.

As a brief summary, the methods that are shown to be performing well in practice in estimating IOTs/SUTs/SAMs include the so-called RAS method and its different extensions (see e.g. Leontief, 1941; Stone, 1961; Bacharach, 1970), the minimum sum of cross entropies (MSCE) approach (Golan et al., 1994; Golan and Vogel, 2000) and the matrix updating methods proposed by Harthoorn and van Dalen (1987) and Kuroda (1988). Other approaches are the normalized squared differences method (Friedlander, 1961), the normalized absolute differences technique (Matuszewski et al., 1964), the squared differences approach (Almon, 1968), a univariate method of statistical correction (Tilanus, 1968), the so-called TAU and UAT methods of Snower (1990), maximum sum of cosine similarity indices (Cardenete and Sancho, 2004), mathematical programming approaches (see e.g. Harrigan and Buchanan, 1984; van der Ploeg, 1988; Canning and Wang, 2005; Geschke et al., 2019) and more complex multi-objective optimization methods (see e.g. Strømman, 2009). Finally, there are IOT and SUT variants of the Euro method (Beutel, 2002, 2008; Eurostat, 2008; United Nations, 2018), which is a largely ad-hoc updating technique (Temursho, 2019). Extensive empirical assessments of various (selected) matrix updating methods are carried out in Jackson and Murray (2004), Oosterhaven (2005), Huang et al. (2008) and Temurshoev et al. (2011). The last two studies also include improved versions of some of the above-mentioned methodologies, where "improved" refers to the treatment of negative elements and/or preservation of signs of the original data points in the derived tables.

The well-known RAS method arguably is the most popular updating method, at least, among practitioners from statistical agencies. It is a *biproportional technique* that is used to estimate a new matrix (e.g. an IOT) from an existing matrix (e.g. an IOT of earlier period) by scaling its entries row- and column-wise so that the pre-specified, exogenously given row and column totals of the updated table are respected.[1] However, the traditional RAS can only handle non-negative matrices, which limits its application to RASing non-negative matrices only. This is, indeed, a serious limitation in practice, in particular when dealing with medium to large-scale IOTs, SUTs and SAMs (or, in general, any other matrix) as these often include negative entries in such items as subsidies, net exports, reductions of inventories, trade margins, transportation margins, and depreciation. Thus, the extension of RAS, called the *generalized RAS* (GRAS) method, originally proposed by Günlük-Şenesen and Bates (1988), but re-discovered and more rigorously formalized by Junius and Oosterhaven (2003), is now a widely used bi-proportional technique for updating or balancing IOTs and SAMs with both positive and negative elements (see also Temurshoev et al., 2013). It is also possible to have *multiple* priors, which leads to the *Cell-corrected RAS* (CRAS): see Mínguez et al. (2009) for the original idea and a national time series application, and Oosterhaven and Escobedo-Cardeñoso (2011) for a more complex spatial cross section application. The SUT-RAS approach, proposed by Temurshoev and Timmer (2011), applies the GRAS updating idea to the *joint* estimation of national SUTs with different settings, such as SUTs at basic prices and purchasers' prices, and use tables separated into domestic and imported use tables.[2, 3]

In Section 2, we present an extension of the GRAS technique to a multi-regional IOT (SUT, SAM or any other partitioned matrix) setting, which necessarily implies the inclusion of additional aggregation constraints that make disaggregated, *inter-* and *intra-regional* data consistent with aggregated, *national* data. We refer to this extension as the Multi-Regional Generalized RAS, or simply the MR-GRAS method. Such extensions have been already made by Oosterhaven et al. (1986), Gilchrist and St. Louis (1999, 2004) and Lenzen et al. (2009). However, while the first three papers focus on updating non-negative matrices, the last study due to its generality loses the inherent transparency

[1]See Lahr and de Mesnard (2004) for details of the RAS method (including its history), which also gives an extensive set of references on the topic.

[2]The SUT-RAS method is used in the World Input-Output Database project (`www.wiod.org`) to construct time-series of national SUTs, which are one of the building blocks of WIOD's world IOTs.

[3]Valderas-Jaramillo et al. (2018) claim that "*none* of the existing methods in the literature reflect explicitly and consistently the taxes less subsidies on products, as in National Accounts. Therefore, we will also introduce ... adapted versions of the SUT-RAS and SUT-EURO methods that explicitly use taxes less subsidies on products (TLS) as integrated part of the updating process. ... The new methods treat TLS explicitly and separately, *unlike* the original SUT-EURO and SUT-RAS methods" (pp. 4-5, italics added). However, we must state that these statements are *not* true for, at least, the following reasons: (1) Temurshoev and Timmer (2011, herein TT) treat TLS by product explicitly, separately and as an integral part of the updating process of SUTs in purchaser's prices, see e.g. Table 5 and Section 2.4 in TT; (2) TT also consider integrated SUTs in basic prices: footnote 24 gives this framework with total uses and with explicit and separate treatment of TLS by use category, further referring to Temurshoev and Timmer (2010) for detailed discussion of an empirical assessment of this framework. Its "extension" to include separate domestic and imported uses is straightforward (see also footnotes 18 in TT and 16 in Section 3 of this paper).

and simplicity of the GRAS approach.[4] Our MR-GRAS approach contributions are the following.

First, we consider updating or projecting a multi-regional IOT/SUT/SAM (or generally a multi-partitioned matrix) which allows for adjusting both its positive and negative entries simultaneously. The inclusion of negative entries into a multi-regional updating framework is clearly an important addition as there are (far) more possibilities of having negative elements within a multi-regional IOT/SUT/SAM setting compared to a national one, due to a higher economic heterogeneity of regions or countries making up the considered economic system.

Second, we provide the complete analytical solution of the MR-GRAS approach and propose a simple iterative algorithm for its computation. There are several advantages of having such transparency:

– There is no need for mastering advanced knowledge to implement complex numerical optimization techniques.
– An analyst does not need to have access to high-performance solvers, since the proposed iterative approach can easily be programmed and applied with widely available software, such as R or Excel.
– It allows for easier control of the convergence process compared to using built-in functions of the available optimization solvers. In particular, in cases of non-convergence one may derive an approximate solution simply by increasing the threshold level of only one stopping criterion of its iterative approach. Then by studying the approximate (non-optimal) table, including the obtained multipliers, one is generally able to find the exact source(s) of such non-convergence problems. In contrast, in case of general purpose solvers one has to change many stopping rule criteria, which is often not straightforward, especially when the researcher has little knowledge of the complex algorithms underlying such optimization routines.

And third, the adjustment multipliers used in MR-GRAS have economic interpretations that could very well be the focus of research. These multipliers make up the MR-GRAS analytical solution and are directly accessible as the output of its iterative algorithm. In contrast, such information cannot be readily retrieved from the applications of numerical optimization techniques.

It must be noted that MR-GRAS is a particular case of multidimensional GRAS pre-

[4]That is, the most flexible framework, the so-called KRAS (K for *Konfliktfreies*) of Lenzen et al. (2009), generalizes the GRAS method to: (i) incorporate constraints on *arbitrary subsets* of matrix elements, including cases of constraints' coefficients being different from 1 or -1, (ii) include *reliability* of the initial estimate and the external constraints, and (iii) find a *compromise solution* between conflicting constraints. It is, however, not surprising that such flexibility comes at the cost of the update control and of rather substantial programming and computational requirements. As such, compared to (MR)GRAS, the KRAS method is *less transparent*, similar to any other numerical optimization technique, including the MSCE approach.

sented in Valderas-Jaramillo and Rueda-Cantuche (2019) as an extension of muldimensional RAS (DRAS) of Holý and Šafr (2017),[5] which are all particular cases of the KRAS method. Nonetheless, we present the details of (the simpler) MR-GRAS because: (a) in practice statisticians mostly encounter updating cases that are consistent with the MR-GRAS setting, and (b) it is likely that most practitioners will find it rather difficult to work with such concepts as hypermatrices or hypercubes when trying to implement the multidimensional (G)RAS (for update settings with 3 dimensions) or KRAS compared to simply applying MR-GRAS to familiar IOT/SUT/SAM settings.

In Section 3, given the importance of regional, national, inter-regional, inter-country and global SUTs in national accounts, from which the corresponding IOTs are derived (see e.g. United Nations, 2018), we separately discuss how the MR-GRAS technique can be applied in updating/balancing of national and inter-country SUTs. We will also discover that the SUT-RAS method is a particular case of the GRAS method, while MR-GRAS further extends the horizons of improved estimation of national SUTs by providing the possibility of incorporating known non-overlapping aggregation values on the different parts of a SUT.

Section 4 provides a detailed guide on how to implement MR-GRAS in practice. The different updating frameworks with exhaustive constraints, non-exhaustive aggregation and/or non-exhaustive disaggregate (i.e. row- and/or column-sums) constraints are explained. For all these cases through a worked example it is demonstrated how to use our MR-GRAS code written in MATLAB programming language. Section 5 concludes.

2 Multi-Regional Generalized RAS

To understand the main properties and the range of possible applications of the MR-GRAS technique, in what follows we present the MR-GRAS mathematical formulation, derive its analytical solution, propose an iterative algorithm for its computation, discuss the possibilities of including non-exhaustive constraints, explain the issue of normalization and interpretation of the MR-GRAS multipliers, and conclude with discussing the main properties of the technique.

2.1 MR-GRAS with (non)exhaustive constraints

Let x_{ij}^{0} and x_{ij} be the ij-th element of the initial (i.e. available) and the target (i.e. unknown) IOTs, respectively. The initial and the target IOTs are rectangular $m \times n$ matrices

[5] We were not aware of these papers until recently, while some of the main results of this work have already been used in the past (to be mentioned in the text).

$\mathbf{X}^0$ and $\mathbf{X}$, respectively.[6] Denote the pre-specified row sums of $\mathbf{X}$ by $u_i = \sum_j x_{ij}$ and the known column sums of $\mathbf{X}$ by $v_j = \sum_i x_{ij}$. To obtain a feasible solution, it has to be assumed that these pre-specified restrictions are mutually consistent, i.e. $\sum_i u_i = \sum_j v_j$. This consistency restriction, however, is not needed with non-exhaustive row and/or column totals, the details of which are discussed below. This is the setting of the traditional GRAS method. The MR-GRAS method, additionally, handles arbitrary non-overlapping *aggregation constraints* of the form $\sum_{i\in I,j\in J} x_{ij} = w_{IJ}$, where the uppercase indices I and J indicate the aggregate (e.g. nation-level) counterparts of the relevant disaggregated (e.g. region-level) row and column indices i's and j's, respectively, and w_{IJ}'s are the corresponding exogenously given aggregate (e.g. national) values.

Following Junius and Oosterhaven (2003), first define the ratio of the unknown (or "new") to the known (or "old") entries of the corresponding IOTs by $z_{ij} \equiv x_{ij}/x^0_{ij}$ whenever $x^0_{ij} \neq 0$. For $x^0_{ij} = 0$, this ratio should be set to unity, i.e. $z_{ij} = 1$ (Lenzen et al., 2007). Then, the MR-GRAS problem is formalized as follows:[7]

$$\min_{z_{ij}} f(\mathbf{Z}) = \sum_i \sum_j |x^0_{ij}| z_{ij} \ln\left(\frac{z_{ij}}{e}\right) \tag{1a}$$

such that

$$\sum_j x^0_{ij} z_{ij} = u_i \quad \text{for all } i = 1, \ldots, m, \tag{1b}$$

$$\sum_i x^0_{ij} z_{ij} = v_j \quad \text{for all } j = 1, \ldots, n, \tag{1c}$$

$$\sum_{i\in I,j\in J} x^0_{ij} z_{ij} = w_{IJ} \quad \text{for all } I = 1, \ldots, M < m \text{ and } J = 1, \ldots, N < n, \tag{1d}$$

where e is the base of natural algorithm. The last $M \times N$ constraints (1d) are additional for MR-GRAS compared to the regular GRAS. In case of updating a multi-regional IOT, these additional constraints ensure that the sum of the corresponding intra-regional and interregional cells of the MR-IOT add to the values of the known new national cells w_{IJ}'s.[8]

[6] Matrices are given in bold capitals; vectors in bold lower cases; and scalars in italicized lower cases. Vectors are columns by definition, row vectors are obtained by transposition, indicated by a prime.

[7] Two notes are in place with respect to the form of the (MR-)GRAS objective function. First, it is sometimes written *without* the base of the natural logarithm. This omission would *not* cause any problem and the two formulations would be entirely equivalent as long as the incorporated constraints fix the overall sum of the adjusted matrix elements, which might *not* always be the case in an MR-GRAS setting with non-exhaustive constraints. Second, Huang et al. (2008) instead propose the "Improved GRAS" function of $f_1(\mathbf{Z}) = \sum_{i,j} |x^0_{ij}| \left[z_{ij} \ln(z_{ij}/e) + 1\right]$. However, one can easily observe that this adjustment does not play any role in determining the optimal solution, hence can be safely ignored.

[8] For simplicity of exposition, in (1a)-(1d) we have *not* used additional superscripts to explicitly distinguish between the disaggregated (i.e. intra- and inter-regional) variables and the aggregated (e.g. national) data as is done in Oosterhaven et al. (1986).

The following important points regarding MR-GRAS constraints need to be stressed. First, we restrict our solution of (1) to the case wherein any disaggregated item $\{i, j\}$ is considered to be a part of *only one* aggregated set $\{I, J\}$. This means that the above-presented MR-GRAS problem only allows for *non-overlapping* aggregation constraints. This choice is favored because this type of aggregation seems to be the prevalent situation researchers and practitioners come across most often. In addition, inclusion of overlapping aggregation constraints generally makes the final solution more complicated, leading to the loss of the simplicity and transparency properties that we wish to maintain in the multi-regional version of GRAS as in RAS. In case of overlapping and possibly conflicting constraints we suggest using the KRAS approach of Lenzen et al. (2009) or some general-purpose constrained optimization solver.

Following the approach of Gilchrist and St. Louis (1999, 2004), in implementing the MR-GRAS method in practice (as shown in Section 4), we use aggregator matrices $\mathbf{G}$ and $\mathbf{Q}$ consisting of zeros and ones in order to compactly and easily (e.g. without any further vectorization) write the aggregation constraints as $\mathbf{G}(\mathbf{X}^0 \circ \mathbf{Z})\mathbf{Q} = \mathbf{GXQ} = \mathbf{W}$, where $\mathbf{W}$ is the $M \times N$ aggregation constraints matrix with typical elements w_{IJ} and the symbol $\circ$ is Hadamard product of element-wise matrix multiplication. As follows from (1d), the aggregator matrices $\mathbf{G}$ and $\mathbf{Q}$ must be of dimensions $M \times m$ and $n \times N$, respectively, where M is not necessarily equal to N. Imposing non-overlapping aggregation constraints requires that the column sums of $\mathbf{G}$ and the row sums of $\mathbf{Q}$ are all unity, i.e. that $\boldsymbol{\imath}'\mathbf{G} = \boldsymbol{\imath}'$ and $\mathbf{Q}\boldsymbol{\imath} = \boldsymbol{\imath}$, where $\boldsymbol{\imath}$ is a summation vector of ones with the appropriate dimension.

Second, if the constraints (1b)-(1d) are exhaustive, i.e. if each element of the target matrix appears in at least one constraint (hence, all the entries of $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{W}$ are known), all constraints have to be *mutually consistent* and consistent with the initial matrix $\mathbf{X}^0$. Checking these consistency requirements is not always trivial, but the following checks will help practitioners in this respect. The following two requirements on the mutual consistency of $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{W}$ need to hold:

1. Identical aggregate row sums: $\mathbf{Gu} = \mathbf{W}\boldsymbol{\imath}$, and
2. Identical aggregate column sums: $\mathbf{v}'\mathbf{Q} = \boldsymbol{\imath}'\mathbf{W}$.

Note that because of the properties of $\mathbf{G}$ and $\mathbf{Q}$ discussed above, these two requirements also imply identical overall sums of the constraints, i.e. $\boldsymbol{\imath}'\mathbf{W}\boldsymbol{\imath} = \boldsymbol{\imath}'\mathbf{u} = \boldsymbol{\imath}'\mathbf{v}$. Besides, the practitioner has to check whether the benchmark table $\mathbf{X}^0$ is consistent with the constraints $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{W}$. For example, if row i of $\mathbf{X}^0$ consists of only negative elements while the corresponding u_i is positive, then the problem is (MR-)GRAS-infeasible and does not have solution (because of MR-GRAS sign-preserving property it is impossible to sum negative entries to a positive number). A similar problem would arise if e.g. column j of $\mathbf{X}^0$ consists of only zeros, but the corresponding v_j is non-zero (because of MR-GRAS zero-preserving property), or $w_{IJ} = 0$ while the corresponding entries in the benchmark table have only one sign, making it impossible to be aggregated to a zero value.

Third, the aggregation constraints are allowed to be *non-exhaustive*. That is, it is *not* necessary that all w_{IJ}'s are known, *some* aggregation values might be missing, which could very well happen in practice. In such cases, the corresponding aggregate values are implicitly and endogenously determined within the MR-GRAS procedure, which depends on the structure of the existing aggregation constraints and the relevant row and column constraints. Section 4.2 shows how to implement MR-GRAS in practice in such circumstances. As is shown in Section 4.2, there are, however, such configurations of missing elements in $\mathbf{W}$ that could be accurately recovered, using the information in $\mathbf{u}$ and $\mathbf{v}$, which effectively boils down to the case of exhaustive aggregation constraints. Of course, in the extreme case when *all* w_{IJ}'s are missing, MR-GRAS boils down to the standard GRAS without any aggregation constraint.

Forth, similar to the previous point, the framework also allows for non-exhaustive row total and/or column total constraints, in which case the assumption $\sum_i u_i = \sum_j v_j$ is not required anymore. We explain the details in Section 4.3, but at this stage to grasp the main idea, let us consider the extreme case when only (certain) aggregation values are known but all the elements of $\mathbf{u}$ and $\mathbf{v}$ are missing. Then one can still apply the MR-GRAS framework as follows. Define new row and column totals as null vectors, i.e. $\overline{\mathbf{u}} = \mathbf{0}$ and $\overline{\mathbf{v}} = \mathbf{0}$ that are, respectively, $(m+1)$- and $(n+1)$-dimensional column vectors. Redefine the benchmark matrix as

$$\overline{\mathbf{X}}^0 = \begin{bmatrix} \mathbf{X}^0 & -\mathbf{u}^0 \\ -(\mathbf{v}^0)' & x^0 \end{bmatrix},$$

where $\mathbf{u}^0$, $\mathbf{v}^0$ and x^0 are, respectively, the row totals, column totals and the overall sum of $\mathbf{X}^0$. The new expanded – by one column and one row – aggregator matrices then could take the form (here for transparency the null vectors' dimensions are shown explicitly):

$$\overline{\mathbf{G}} = \begin{bmatrix} \mathbf{G} & \mathbf{0}_M \\ \mathbf{0}'_m & 1 \end{bmatrix} \text{ and } \overline{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \mathbf{0}_n \\ \mathbf{0}'_N & 1 \end{bmatrix}, \text{ which results in } \overline{\mathbf{G}}\,\overline{\mathbf{X}}^0\overline{\mathbf{Q}} = \begin{bmatrix} \mathbf{G}\mathbf{X}^0\mathbf{Q} & -\mathbf{G}\mathbf{u}^0 \\ -(\mathbf{v}^0)'\mathbf{Q} & x^0 \end{bmatrix}.$$

Thus, if $\mathbf{W}$ is exhaustive, then all the entries of the expanded new aggregation constraints matrix $\overline{\mathbf{W}} = \overline{\mathbf{G}}\,\overline{\mathbf{X}}\,\overline{\mathbf{Q}}$ are also known before the update procedure because of the constraints' mutual consistency requirements discussed above, i.e.

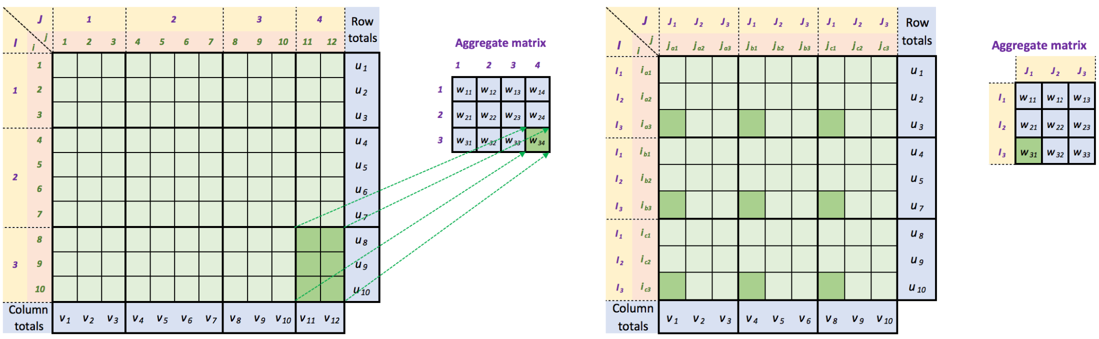
$$\overline{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & -\mathbf{W}\boldsymbol{\imath} \\ -\boldsymbol{\imath}'\mathbf{W} & \boldsymbol{\imath}'\mathbf{W}\boldsymbol{\imath} \end{bmatrix}.$$

However, in the presence of missing w_{IJ}'s, all the entries in $\overline{\mathbf{W}}$ need *not* be specified in running the MR-GRAS as they will be automatically imposed within its iterative procedure and will be determined (as corresponding residuals) by the known values in $\mathbf{W}$, $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$.

Finally, the MR-GRAS framework is also general enough to allow for different configurations of *both* non-exhaustive aggregation constraints and non-exhaustive column- and/or row-sums constraints. Such an example is considered in Section 4.4.

Two simple hypothetical examples of data structures that can be handled by MR-GRAS are illustrated in Figure 1. The first example shows international (or interregional) trade flows for a particular commodity and the second example shows an intermediate transactions matrix of a hypothetical MRIO table.

Figure 1: Examples of MR-GRAS consistent data structure



(a) International trade of one good　　**(b)** An MRIO table

Note: The row and column indices of the disaggregated data are given by i and j, respectively, whereas their corresponding aggregate sets are I and J. Dark cells visualize the components of one aggregation constraint in each case. Figure (1a) is an example of international trade data for a specific good. Figure (1b) represents a hypothetical MRIO table with three countries a, b and c, and three sectors 1, 2 and 3; for simplicity, only the intermediate transactions are shown.

To find the solution of (1), following Junius and Oosterhaven (2003), first the original matrix needs to be decomposed as $\mathbf{X}^{\mathrm{O}} = \mathbf{P}^{\mathrm{O}} - \mathbf{N}^{\mathrm{O}}$, where $\mathbf{P}^{\mathrm{O}}$ contains the positive elements of $\mathbf{X}^{\mathrm{O}}$ and $\mathbf{N}^{\mathrm{O}}$ consists of the absolute values of the negative entries in $\mathbf{X}^{\mathrm{O}}$. Then the Lagrangean function of the MR-GRAS problem (1a)-(1d) is written as follows:

$$\mathcal{L} = \sum_{(i,j)\in\mathbf{P}^{\mathrm{O}}} x_{ij}^{\mathrm{O}} z_{ij} \ln\left(\frac{z_{ij}}{e}\right) - \sum_{(i,j)\in\mathbf{N}_{\mathrm{O}}} x_{ij}^{\mathrm{O}} z_{ij} \ln\left(\frac{z_{ij}}{e}\right) + \sum_{i} \lambda_i \Big(u_i - \sum_j x_{ij}^{\mathrm{O}} z_{ij}\Big)$$
$$+ \sum_j \tau_j \Big(v_j - \sum_i x_{ij}^{\mathrm{O}} z_{ij}\Big) + \sum_{(I,J)} \mu_{IJ} \Big(w_{IJ} - \sum_{i\in I, j\in J} x_{ij}^{\mathrm{O}} z_{ij}\Big),$$

where λ_i, τ_j and μ_{IJ} are the Lagrange multipliers of the three respective sets of constraints in (1b)-(1d). The optimal solution of this function can be easily derived and compactly written, making use of the decomposition $x_{ij}^{\mathrm{O}} = p_{ij}^{\mathrm{O}} - n_{ij}^{\mathrm{O}}$, as follows:

$$x_{ij} = t_{IJ} \cdot r_i p^0_{ij} s_j - \frac{1}{t_{IJ}} \frac{1}{r_i} n^0_{ij} \frac{1}{s_j}, \quad (2)$$

where $r_i \equiv e^{\lambda_i} > 0$, $s_j \equiv e^{\tau_j} > 0$ and $t_{IJ} \equiv e^{\mu_{IJ}} > 0$. Hence, similar to the standard GRAS solution (Günlük-Şenesen and Bates, 1988; Junius and Oosterhaven, 2003), the row and column multipliers are given by r_i and s_j, respectively. However, in contrast to the GRAS method, the MR-GRAS approach is *no longer* a bi-proportional technique as the added *aggregation multipliers* t_{IJ}'s make its solution three-proportional. Not surprisingly, with $t_{IJ} = 1$ for all I and all J, i.e. without aggregation constraints, the MR-GRAS solution boils down to that of the standard GRAS approach.[9] Our choice of non-overlapping aggregation constraints implies that for any $\{i, j\}$ in (2) there is only one t_{IJ}, representing the multiplier of the aggregation constraints which includes the ij-th entry in question. If for some pair $\{I, J\}$ the aggregation constraint is not specified (or is unknown), the corresponding aggregation multiplier can be considered to be unity from the outset, $t_{IJ} = 1$.

In what follows, the analytical expressions of the MR-GRAS multipliers are derived and a simple iterative algorithm for their computation is proposed. Plugging (2) in the row sum constraints (1b) and multiplying both of its sides by r_i gives a quadratic equation in r_i, which is equation (3a) below. Similarly, substituting (2) in the MR-GRAS column sum constraints (1c) and multiplying it by s_j gives the second quadratic equation in s_j, shown as equation (3b). Finally, substitution of (2) in the aggregation constraints (1d) and its multiplication by t_{IJ} gives the third quadratic equation (3c) in t_{IJ}. Thus, the three unknown multipliers are derived from the following system of three equations:

$$p_i(s,t) r_i^2 - u_i r_i - n_i(s,t) = 0, \quad (3a)$$

$$p_j(r,t) s_j^2 - v_j s_j - n_j(r,t) = 0, \quad (3b)$$

$$p_{IJ}(r,s) t_{IJ}^2 - w_{IJ} t_{IJ} - n_{IJ}(r,s) = 0, \quad (3c)$$

where the quadratic coefficient and the constant term in each of the above equations are defined, respectively, as follows:

$$p_i(s,t) = \sum_j p^0_{ij} s_j t_{IJ} \quad \text{and} \quad n_i(s,t) = \sum_j \frac{n^0_{ij}}{s_j t_{IJ}}, \quad (4a)$$

$$p_j(r,t) = \sum_i r_i p^0_{ij} t_{IJ} \quad \text{and} \quad n_j(r,t) = \sum_i \frac{n^0_{ij}}{r_i t_{IJ}}, \quad (4b)$$

[9] Hence within the GRAS framework, "... the procedure RAS is appropriate for positive elements, but needs to be replaced by (1/R)A(1/S) for negative elements" (Günlük-Şenesen and Bates, 1988, p. 476). The *complete* analytical expressions of the GRAS multipliers r_i and s_j are presented in Temurshoev et al. (2013).

$$p_{IJ}(r,s) = \sum_{i\in I, j\in J} r_i p^0_{ij} s_j \quad \text{and} \quad n_{IJ}(r,s) = \sum_{i\in I, j\in J} \frac{n^0_{ij}}{r_i s_j}. \tag{4c}$$

Note that the appearance of r, s and/or t within the brackets above explicitly indicates the dependence of the corresponding coefficients on the row, column and/or aggregation multipliers, respectively. For simplicity, however, the dependence of these terms on the observed, exogenous elements, i.e. p^0_{ij} or n^0_{ij}, is suppressed.

It can be easily verified that each of the equations (3a)-(3c) has two real roots, one positive and one negative. However, since the three sets of multipliers r_i, s_j and t_{IJ} must be positive by construction as follows from (2), the relevant solutions of interest here are only the positive ones. Thus, using the well-known quadratic formula, the closed-form solutions of the three sets of multipliers are given by the following expressions:

$$r_i = \begin{cases} \dfrac{u_i + \sqrt{u_i^2 + 4p_i(s,t)n_i(s,t)}}{2p_i(s,t)} & \text{for } p_i(s,t) > 0, \\ -\dfrac{n_i(s,t)}{u_i} & \text{for } p_i(s,t) = 0, \end{cases} \tag{5}$$

$$s_j = \begin{cases} \dfrac{v_j + \sqrt{v_j^2 + 4p_j(r,t)n_j(r,t)}}{2p_j(r,t)} & \text{for } p_j(r,t) > 0, \\ -\dfrac{n_j(r,t)}{v_j} & \text{for } p_j(r,t) = 0, \end{cases} \tag{6}$$

$$t_{IJ} = \begin{cases} \dfrac{w_{IJ} + \sqrt{w_{IJ}^2 + 4p_{IJ}(r,s)n_{IJ}(r,s)}}{2p_{IJ}(r,s)} & \text{for } p_{IJ}(r,s) > 0, \\ -\dfrac{n_{IJ}(r,s)}{w_{IJ}} & \text{for } p_{IJ}(r,s) = 0. \end{cases} \tag{7}$$

Note that each multiplier in (5)-(7) is defined by two expressions: the first one is valid when the corresponding row(s) and/or column(s) of the original matrix include(s) at least one positive element, while the second solution is used instead when the relevant elements of the old matrix consist(s) of only non-positive entries with at least one negative element (for details on the relevance of this issue, which becomes even more relevant in a multiregional setting, see Temurshoev et al., 2013). Note from (5) that when $p_i(s,t) = 0$ but $n_i(s,t) > 0$ (i.e. row i in $\mathbf{X}^0$ includes only negative non-zero entries), the row multiplier r_i is also positive (as it expected to be), because then it also must be true from the requirement of consistent benchmark matrix and constraints that $u_i < 0$. Similarly, whenever $p_j(r,t) = 0$ and $n_j(r,t) > 0$, we have again positive multiplier $s_j > 0$ because then it must be the case that $v_j < 0$. And, finally, in case for a particular pair of aggregate sets $\{I, J\}$ all the corresponding elements in the original matrix are non-positive with at least one negative entry, then the consistent aggregation constraint must have $w_{IJ} < 0$ for the pair $\{I, J\}$ under consideration. The last will also ensure that the corresponding aggregation multiplier t_{IJ} is positive.

There is, however, one special case – uninteresting and unlikely to be encountered in an IOT/SUT/SAM updating – when the MR-GRAS iterative procedure results in *zero* multipliers. We refer to this case as *weak sign preservation* property of the MR-GRAS method. This possibility is discussed later in Sections 2.3 and 4.5.

In order to compute the required multipliers and the new adjusted matrix $\mathbf{X}$, we propose the following simple iterative algorithm:

- Iteration $iter = 0$: Set $r_i(iter = 0) = 1$ for all i and $t_{IJ}(iter = 0) = 1$ for all I and all J. This initialization essentially means that one starts the adjustment procedure from the original matrix.
- Iteration $iter = 1, 2, ..., k$: Perform the following sequence of computations.

  (a) Calculate $s_j(iter)$ using (6) as a function of $r_i(iter - 1)$ and $t_{IJ}(iter - 1)$, both derived from the previous iteration;

  (b) Calculate $r_i(iter)$ using (5) as a function of $s_j(iter)$ of the current iteration and $t_{IJ}(iter - 1)$ from the previous iteration;

  (c) Calculate $t_{IJ}(iter)$ using (7) as a function of $r_i(iter)$ and $s_j(iter)$, both of current iteration.

- Iteration $iter = k$: Stop when the multipliers converge for a sufficiently small tolerance level $\epsilon > 0$, i.e. when $s_j(k) - s_j(k-1) < \epsilon$ for all j, $r_i(k) - r_i(k-1) < \epsilon$ for all i, and $t_{IJ}(k) - t_{IJ}(k-1) < \epsilon$ for all I and J.
- Finally, using the last iteration values of multipliers $r_i(k)$, $s_j(k)$ and $t_{IJ}(k)$ derive the adjusted entries of the new matrix, x_{ij}'s, using the MR-GRAS solution (2).

2.2 Normalization and interpretation of MR-GRAS multipliers

How to interpret the row, column and aggregation multipliers of the MR-GRAS solution? In the IO literature, the uniform changes along any row and down any column of an intermediate input coefficients matrix are interpreted to reflect the *substitution effects* and *fabrication effects*, respectively (Stone, 1961). That is, it is often claimed that r_i measures whether input i has been replaced by other inputs (if $r_i < 1$) or has replaced other inputs (if $r_i > 1$) in the updated vs. the original IO matrix. The fabrication factor s_j indicates whether sector j absorbs more (if $s_j > 1$) or less (if $s_j < 1$) intermediate inputs compared to primary inputs (Miller and Blair, 2009, Chap. 7.4.4). Oosterhaven et al. (1986) interpret the aggregation multiplier t_{IJ} as *technology effect* "indicating the general rise or fall in importance" of product i for sector j (p. 62). More generally, the aggregation multipliers could be termed as sector-specific or non-sector-specific *intra- or inter-regional effects* that account for all common factors identically affecting the corresponding intra- or inter-regional/country trade flows. In fact, with such interpretation of t_{IJ}'s, the MR-GRAS approach has been recently applied in estimating inter-regional trade flows among different regions in the US by Zhao and Squibb III (2019).[10]

However, any interpretation of the standard (G)RAS multipliers require a normalization since r_i's and s_j's are unique only up to a scalar (see e.g. Toh, 1998; Lahr and de Mesnard, 2004; de Mesnard, 2002, 2004). One encounters the same non-uniqueness problem with MR-GRAS multipliers. From (2) it is easy to verify that if r_i, s_j and t_{IJ} are the MR-GRAS multipliers, so are e.g. $\delta \times r_i$, s_j/δ and t_{IJ}, or alternatively $\delta^{1/2} \times r_i$, s_j/δ and $\delta^{1/2} \times t_{IJ}$ for any $\delta > 0$. In all these cases the updated matrix $\mathbf{X}$ is exactly the same, but the underlying multipliers are unique only up to a scalar. Therefore, as follows from (2), what is uniquely identified is the product $t_{IJ} r_i s_j$ for all $i \in I$ and all $j \in J$. To overcome the problem of the non-uniqueness of multipliers, the use of a scaling equation (i.e. normalization) is required. In general, it is preferred that the choice of normalization is justifiable on theoretical and/or empirical grounds for a setting within which the (MR)GRAS technique is applied. For example, within the IO framework, van der Linden and Dietzenbacher (2000) propose a normalization based on a sound economic reasoning which states that for the whole system under consideration the global substitution effect should be zero. The latter simply means that the system-wide use of intermediate inputs with substitution effects is equal to the system-wide intermediate use with no substitution.[11] However, in general the required scaling equation will *not* hold for r_i's which are the solution of the MR-GRAS algorithm, hence they need to to be normalized. If we denote the normalized multipliers by a star superscript, then the first normalization task is to convert r_i into r_i^* for all i such that $\sum_i (u_i/r_i^*) = \sum_i u_i$. Let us denote the weighted harmonic mean of *non*-normalized row multipliers r_i's by

$$\bar{r}_H \equiv \frac{\sum_i u_i}{\sum_i (u_i/r_i)},$$

[10]The MR-GRAS has been also recently used by Joint Research Centre of the European Commission is projecting MR-IOTs for the Baseline scenario that represents a projection of the world economy under the assumption of current climate and energy policies and also realization of Nationally Determined Contributions (NDC) in line with Paris agreement. For details, see Rey Los Santos et al. (2018). The resulting IOTs projections are available for download at `http://data.europa.eu/89h/3ffc59a1-edff-491f-8894-3147d2202e42`.

[11]In mathematical terms, this later normalization is equivalent to $\sum_i (u_i/r_i) = \sum_i u_i$, where its left-hand side, $\sum_i (u_i/r_i)$, is the updated/new total intermediate deliveries corrected for substitution effects and its right-hand side, $\sum_i u_i$, is the new total intermediate deliveries without substitution (i.e. $r_i = 1$ for all i). Note that this scaling equation is equivalent to unitary "weighted harmonic mean of average substitution effects r_i" (van der Linden and Dietzenbacher, 2000, p. 2212), i.e. $\frac{\sum_i u_i}{\sum_i (u_i/r_i)} = 1$, where the weights are represented by sectoral intermediate deliveries u_i's.

which will generally be different from unity. Then adopting the normalization of van der Linden and Dietzenbacher (2000) implies that after the MR-GRAS algorithm, the normalized multipliers are obtained as follows:

$$r_i^* = \frac{r_i}{\bar{r}_H}, \quad s_j^* = s_j \times \bar{r}_H \quad \text{and} \quad t_{IJ}^* = t_{IJ}. \tag{8}$$

It is easy to show that the substitution effects r_i^*'s in (8) indeed satisfy $\sum_i (u_i / r_i^*) = \sum_i u_i$, or equivalently, the weighted harmonic mean of normalized row multipliers is unity, i.e. $\bar{r}_H^* = 1$. Hence, by restricting the average substitution effects, the normalized column multipliers, $s_j^* = s_j \times \bar{r}_H$, would cover the fabrication effects of substitution of intermediate inputs for primary factor inputs.[12]

In other settings, e.g. when updating the flows of migrants, social networks, flows of financial assets, etc., the interpretation of the MR-GRAS multipliers would be different which implies that other normalization (most likely) could be more relevant. That is, it is the multipliers interpretation which is critical in choosing the most appropriate scaling equation(s) for the problem at hand.

2.3 The main properties of MR-GRAS method

The (G)RAS method has several attractive properties that make it popular and (arguably) the most widely used updating method among practitioners. One of such properties is the existence of closed-form solution, which makes GRASing a rather simple procedure to implement, requiring no advanced optimization and programming knowledge. As follows from Section 2.1, this property also holds for the MR-GRAS approach. Being a rescaling method, the (MR-G)RAS is also *transparent* as there is a simple relation (2) between the updated and benchmark matrices.

Two important economic structure-maintaining properties of the (MR-G)RAS technique are its *sign-preserving* and *zero-preserving* properties. This follows from the MR-GRAS analytical solution (2) and the implied strictly positive scaling multipliers. We refer to this property as *strong sign preservation*. However, compared to the analytical solution, the MR-GRAS iterative procedure gives a possibility of what we call as *weak sign preservation*. In the standard RAS one can have a row with at least one strictly positive number and a constraint for that row equal to zero. Then the basic iterative RAS solution sets

[12] It should be noted that, in general, there are two options for normalization: (1) a chosen scaling equation is used within each individual iteration of the (MR-G)RAS procedure, or (2) normalization is applied to multipliers after the (MR-G)RAS iterative procedure. de Mesnard (2004) calls these approaches, respectively, as *ex ante* normalization and *ex post* normalization. As shown by de Mesnard (2004), for *ex ante* normalization it is impossible to analytically derive the normalized solution and convergence must be proved at each step of normalization. For these reasons, here the *ex post* normalization approach is adopted.

that whole row equal to zero. This possibility can be verified using (5): without any negative element along the non-zero row i, i.e. $p_i(s,t) > 0$ but $n_i(s,t) = 0$, the zero row sum requirement $u_i = 0$ implies zero multiplier, $r_i = 0$. So all strictly positive numbers along row i of the original table change into a semi-positive number (actually 0). Thus, "weak sign preservation". Using (6), similar possibility can be seen for a non-zero and non-negative column j that is constrained to sum to zero, i.e. $v_j = 0$.

The same possibility arises within the MR-GRAS setting: if in the original table all the elements $\{i, j\}$ corresponding to one specific aggregate constraint $\{I, J\}$ are non-negative with at least one positive element and are constrained to sum to zero, $w_{IJ} = 0$, then all the original non-zero elements turn into a semi-positive number, 0. This again can easily be verified using (7). However, it must be recognized that such cases of turning positive elements into zeros due to zero row-, column- and/or aggregate sums is completely uninteresting for updating purposes. Why would one ever want to nullify a set of positive entries in an IOT/SUT/SAM update? Economies grow and become more complex over time, thus positive flows turning into zero is not very likely. But even if such cases arise (e.g. old industries cease to exist), simply nullifying (manually) the corresponding original entries does the job (see Section 4.5). As such one can safely ignore the relevance of weak sign preservation for IOT/SUT/SAM updating.

A case that is relevant for updating purposes is constraining elements of both signs to sum to zero. As follows from (5)-(7), in the presence of both positive and negative elements with corresponding zero summation constraints, the multipliers will always be strictly positive. Thus, the underlying positive and negative elements will keep their signs in its strict sense and will be updated (or equally contribute) to sum to zero. Finally, as follows from our discussions in Section 2.1, if the constrained elements include only negative non-zero values, these do *not* change into zeros. In such cases, MR-GRAS is infeasible: the second expressions in (5)-(7) are not defined unless the corresponding constraints' values are also strictly negative.

To sum, positive and negative elements in the benchmark matrix keep their original signs after the updating procedure, while zero benchmark entries remain zeros in the adjusted matrix. The usefulness of these properties could be assessed from two perspectives. On the one hand, if one is really interested in keeping the structure of the old matrix in terms of its elements' signs and zero values in the projected matrix, then MR-GRAS is an ideal updating technique to use. On the other hand, if the problem at hand finds it *critical* to allow for changing signs and switching between zero to non-zero values, then applying MR-GRAS is not recommended. However, if one has information on which entries would change their signs or become (non)zero, this could readily be introduced within the benchmark table and the "desirable" outcome would be obtained from the MR-GRAS procedure. Within the IO framework, often economic structure changes slowly over short period of time, hence projections using MR-GRAS are generally acceptable. For long-term projections, however, the option of introducing changes in the economic structure of the benchmark data needs to be considered seriously.

The zero-preserving property of the MR-GRAS procedure also allows one to incorporate additional information on the updated matrix in exactly the same way as is implemented in the so-called modified RAS approach.[13] This property of introducing additional correct exogenous data is critically important since it is generally recommended to use all available information in the updating procedure. As de Mesnard and Miller (2006) put it "[a]s a general rule, introduction of accurate exogenous information into RAS improves the resulting estimates, and counterexamples should probably not be taken too seriously" (p. 517).

McDougall (1999) makes a detailed comparison of RAS and other entropy-theoretic methods, including the minimum sum of cross entropies (MSCE) technique (Golan et al., 1994; Golan and Vogel, 2000),[14] and argues that, in general, RAS remains the preferable matrix balancing technique. In particular, he states that there is "one important and desirable property that the RAS method has and MSCE does not: ... the RAS preserves the ordering of input intensities across industries, ... [while] in general, the MSCE estimates do not preserve the intensity ordering" (p. 10, Proposition 5). By preserving input intensity ordering it is meant that if for any pair of industries h and k in the base table one has the relation

$$\frac{x^0_{i_1h}}{x^0_{i_2h}} \lesseqgtr \frac{x^0_{i_1k}}{x^0_{i_2k}}, \quad \text{then in the target table it is also valid that} \quad \frac{x_{i_1h}}{x_{i_2h}} \lesseqgtr \frac{x_{i_1k}}{x_{i_2k}}.$$

Note that not only the input intensities are maintained, but also the output intensities, which may be important if applications of the supply-driven IO model are foreseen (see Oosterhaven, 2012, for the latest warning to do so). That is, applied to RASed value added components, it means that the ordering of the capital-to-labor ratios across industries in the original data is preserved after RAS balancing. This is indeed an important property, since "there is nothing in the new data to support any reversal in relative input intensities; there cannot be, since the new data contain no industry-specific information about cost structures" (McDougall, 1999, p. 11). However, the same *inputs-intensity-ordering pre-*

[13]The modified MR-GRAS approach would work as follows: nullify all cells of the benchmark table which are known in $\mathbf{X}$; adjust the corresponding row, column and aggregation totals for this existing information; run the MR-GRAS procedure; and finally add back the known information into the updated matrix (see e.g. Miller and Blair, 2009, Chap. 7.4.5).

[14]Temursho (2018) applies MSCE to a *benchmarking problem*, where less precise high-frequency data need to be adjusted to match the more reliable low-frequency data.

serving property *only partially* holds for the MR-GRAS approach: it only holds for MR-GRAS if the industries h and k under consideration are subject to the same aggregation constraint set J of purchasers of inputs.[15]

There are also two other properties of the traditional RAS that are not kept in its generalized versions when the benchmark matrix includes at least one negative element. First, Dietzenbacher and Miller (2009, hereafter DM) formally showed that for the RAS outcome it does *not* make a difference whether one is updating a transactions matrix, or the corresponding input or output coefficients matrices. In fact, the proof is also valid for MR-RAS because their main conclusion is not affected by adding other restrictions to the basic RAS problem, such as when "the sum of a set of elements may be known *a priori*, or inequalities for sets of elements may be imposed" (DM, p. 559). This "uniqueness" property of "generating the same answer whether updating the transactions or the coefficients is a very attractive property that holds exclusively for RAS, at least within the set of commonly applied updating procedures" (DM, p. 564).

This attractive property of RAS, however, does not hold for the GRAS technique when negative elements are present in $\mathbf{X}^0$. So, it will not hold for MR-GRAS either. Following DM's approach, this can be proven as follows. If updating is done on the transactions matrix $\mathbf{X}^0 = \mathbf{P}_x^0 - \mathbf{N}_x^0$, then the GRAS solution $\mathbf{X}$ will satisfy the following three conditions:

$$\mathbf{X}_x = \hat{\mathbf{r}}_x \mathbf{P}_x^0 \hat{\mathbf{s}}_x - \hat{\mathbf{r}}_x^{-1} \mathbf{N}_x^0 \hat{\mathbf{s}}_x^{-1}, \tag{9a}$$

$$\mathbf{X}_x \boldsymbol{\iota} = \mathbf{u}, \tag{9b}$$

$$\boldsymbol{\iota}' \mathbf{X}_x = \mathbf{v}', \tag{9c}$$

where subscript x refers to the case of updating transactions, $\mathbf{r}_x$ and $\mathbf{s}_x$ are the corresponding row and column multipliers, and a "hat" is used to indicate a diagonal matrix.

Next, consider updating "input coefficients" matrix defined as $\mathbf{A}^0 = \mathbf{X}^0(\hat{\mathbf{v}}^0)^{-1} = \mathbf{P}_a^0 - \mathbf{N}_a^0$. The GRAS solution in this case, $\mathbf{A}_a$, satisfies the following three conditions:

[15] The proof is as follows. Assume that $x^0_{i_1 h}$, $x^0_{i_2 h}$, $x^0_{i_1 k}$ and $x^0_{i_2 k}$ are all strictly positive. Then

$$\frac{x_{i_1 h}}{x_{i_2 h}} - \frac{x_{i_1 k}}{x_{i_2 k}} = \frac{r_{i_1}}{r_{i_2}} \left[\frac{t_{I_{i_1} J_h}}{t_{I_{i_2} J_h}} \times \frac{x^0_{i_1 h}}{x^0_{i_2 h}} - \frac{t_{I_{i_1} J_k}}{t_{I_{i_2} J_k}} \times \frac{x^0_{i_1 k}}{x^0_{i_2 k}} \right],$$

where e.g. the aggregation index I_{i_2} means that $i_2 \in I_{i_2}$. If h and k belong to the same aggregation constraint set J indicating purchasers of inputs, i.e. $J_h = J_k$, the above relation boils down to:

$$\frac{x_{i_1 h}}{x_{i_2 h}} - \frac{x_{i_1 k}}{x_{i_2 k}} = \frac{r_{i_1}}{r_{i_2}} \frac{t_{I_{i_1} J_h}}{t_{I_{i_2} J_h}} \left[\frac{x^0_{i_1 h}}{x^0_{i_2 h}} - \frac{x^0_{i_1 k}}{x^0_{i_2 k}} \right],$$

which guarantees the preservation of original inputs intensities in the final estimates. (Similar derivations hold for other cases of negative entries only or the combinations of positive and negative elements.) Otherwise, as long as $J_h \neq J_k$, no such guarantee exists in general as follows from the first expression above.

$$\mathbf{A}_a = \hat{\mathbf{r}}_a \mathbf{P}_a^0 \hat{\mathbf{s}}_a - \hat{\mathbf{r}}_a^{-1} \mathbf{N}_a^0 \hat{\mathbf{s}}_a^{-1}, \tag{10a}$$

$$\mathbf{A}_a \mathbf{v} = \mathbf{u}, \tag{10b}$$

$$\boldsymbol{\imath}' \mathbf{A}_a = \boldsymbol{\imath}'. \tag{10c}$$

To clearly see the relation between (9) and (10), write (10) in terms of the transactions matrix, which after GRASing is obtained from $\mathbf{X}_a = \mathbf{A}_a \hat{\mathbf{v}}$. Then, the three conditions in (10) are equivalent to the following three conditions, respectively:

$$\mathbf{X}_a = \hat{\mathbf{r}}_a \mathbf{P}_x^0 (\hat{\mathbf{v}}^0)^{-1} \hat{\mathbf{s}}_a \hat{\mathbf{v}} - \hat{\mathbf{r}}_a^{-1} \mathbf{N}_x^0 (\hat{\mathbf{v}}^0)^{-1} \hat{\mathbf{s}}_a^{-1} \hat{\mathbf{v}}, \tag{11a}$$

$$\mathbf{X}_a \boldsymbol{\imath} = \mathbf{u}, \tag{11b}$$

$$\boldsymbol{\imath}' \mathbf{X}_a = \mathbf{v}'. \tag{11c}$$

where we have used $\mathbf{P}_a^0 \hat{\mathbf{v}}^0 = \mathbf{P}_x^0$ and $\mathbf{N}_a^0 \hat{\mathbf{v}}^0 = \mathbf{N}_x^0$. Although the last two conditions in (11) are similar in nature to their counterparts in (9), the first conditions (9a) and (11a) are *not* identical. To see this, denote $\hat{\mathbf{s}}_1 \equiv (\hat{\mathbf{v}}^0)^{-1} \hat{\mathbf{s}}_a \hat{\mathbf{v}}$ and $\hat{\mathbf{s}}_2^{-1} \equiv (\hat{\mathbf{v}}^0)^{-1} \hat{\mathbf{s}}_a^{-1} \hat{\mathbf{v}}$, hence (11a) can be rewritten as $\mathbf{X}_a = \hat{\mathbf{r}}_a \mathbf{P}_x^0 \hat{\mathbf{s}}_1 - \hat{\mathbf{r}}_a^{-1} \mathbf{N}_x^0 \hat{\mathbf{s}}_2^{-1}$. Note that $\mathbf{s}_1 = [(\hat{\mathbf{v}}^0)^{-1} \hat{\mathbf{v}}]^2 \mathbf{s}_2$. Thus, given that in general $\mathbf{s}_1 \neq \mathbf{s}_2$ (unless $v_i = v_i^0$ for all i), the first conditions (9a) and (11a) are not the same, i.e. $\mathbf{X}_x \neq \mathbf{X}_a$. A similar conclusion can be obtained if instead of input coefficients, the corresponding output (or allocation) coefficients matrix were used. Similar to DM discussions (p. 559), adding additional restrictions, including aggregation constraints, does not alter the last conclusion.

Finally, on the base of the principle of insufficient reason (or Laplace criterion), it could be argued that another desirable property of a matrix updating technique might be a *homotheticity property*: if all the exogenous constraints are multiplied by the same scalar k, then the new/updated matrix equals the same multiple of the old matrix, i.e. $\mathbf{X} = k\mathbf{X}^0$. Motorin (2017) showed that while RAS passes this homothetic test, GRAS does not. Hence, this statement also holds true for MR-GRAS when there is at least one negative element in the benchmark table.

3 Updating national and multi-regional/country SUTs

By now it is clear that the MR-GRAS method can be used for constructing/balancing/updating *any* partitioned matrix to conform to new, exogenously given exhaustive or non-exhaustive constraints (where the aggregation constraints are non-overlapping), while is applicable for wide variety of problems. Given the importance of supply and use tables (SUTs) in national accounts, this section discusses MR-GRAS applied to updating SUTs.

With the appropriate formulation of the benchmark matrix $\mathbf{X}^0$ and the corresponding constraints $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{W}$, MR-GRAS can be readily used to jointly update/project SUTs, both within national and inter-regional/country settings. This is also true with respect to any SUTs framework, be it in basic prices or in purchasers prices, or where total Use is separated into domestic and imported Use. In fact, it turns out that one may already use the GRAS approach to estimate SUTs when total outputs by product are not available. This was one of the main motivations for introducing the SUT-RAS approach by Temurshoev and Timmer (2011, hereafter TT), who use *directly and without any transformation* the main components of SUTs and derive the closed-form expressions for these components.

Consider as an example the national SUTs framework in basic prices. For such a setting we could formulate the benchmark matrix as follows (for simplicity, zero superscripts denoting the benchmark character of the components of SUTs are suppressed on the right-hand side):

$$\mathbf{X}^0 = \begin{bmatrix} -\mathbf{S}^d & \mathbf{0} & \mathbf{U}_b^d & \mathbf{Y}_b^d \\ \mathbf{O} & -\mathbf{m} & \mathbf{U}_b^m & \mathbf{Y}_b^m \\ \mathbf{0}' & 0 & \mathbf{t}_u' & \mathbf{t}_y' \end{bmatrix}, \tag{12}$$

where $\mathbf{O}$ and $\mathbf{0}$ are, respectively, the null matrix and null vector of appropriate dimensions,

- $\mathbf{S}^d$ denotes the (domestic) supply matrix of dimension product by industry ($p \times s$), which is a transpose of the make matrix often denoted by $\mathbf{V}$;
- $\mathbf{m}$ is the p-dimensional vector of total imports priced at CIF;
- $\mathbf{U}_b^d$ and $\mathbf{U}_b^m$ are $p \times s$ matrices of, respectively, domestic and imported intermediate uses at basic prices;
- $\mathbf{Y}_b^d$ and $\mathbf{Y}_b^m$ are, respectively, $p \times f$ matrices of domestic and imported final uses at basic prices (where f is the number of final use categories); and
- $\mathbf{t}_u$ and $\mathbf{t}_y$ are, respectively, s- and f-dimensional vectors of total taxes less subsidies on products for intermediate and final uses.

Note that for any balanced SUTs framework, the row sums of the above benchmark matrix, except for its last row, are all zeros because of product-level supply-use accounting balances of domestic production and imports, i.e. $\mathbf{S}^d\boldsymbol{\imath} = \mathbf{U}_b^d\boldsymbol{\imath} + \mathbf{Y}_b^d\boldsymbol{\imath}$ and $\mathbf{m} = \mathbf{U}_b^m\boldsymbol{\imath} + \mathbf{Y}_b^m\boldsymbol{\imath}$.

Let us now assume, similar to TT, that for a new matrix (to be projected on base of $\mathbf{X}^0$) only the following information is available: total output by industry at basic prices $\mathbf{x}' = \boldsymbol{\imath}'\mathbf{S}^d$, the vector of gross value-added (GVA) by industry at basic prices $\mathbf{w}$, the total (domestic and imported) final uses by final demand category at purchasers' prices $\mathbf{y}_f$, and the economy-wide totals of imports (m_{tot}) and net taxes on products (t_{tot}). To be able to use the GRAS method for the updating purpose, the row and column sums of the new integrated SUT matrix $\mathbf{X}$ are defined as follows (the subscript $2p$ of the null vector indicates its dimension):

$$\mathbf{u}' = \begin{bmatrix} \mathbf{0}'_{2p} & t_{tot} \end{bmatrix} \quad \text{and} \quad \mathbf{v}' = \begin{bmatrix} -\mathbf{x}' & -m_{tot} & \mathbf{x}' - \mathbf{w}' & \mathbf{y}'_f \end{bmatrix}.$$

GRASing the above $\mathbf{X}^0$ to satisfy the above constraints $\mathbf{u}$ and $\mathbf{v}$ yields exactly the same SUTs as those produced by the SUT-RAS approach: the outcome of these two approaches must be the same as the two problems are equivalent, i.e. they have the same objective function and constraints. Nonetheless, in Appendix A we show this equivalence.[16]

All other frameworks of national SUTs can be updated/estimated using the GRAS approach by appropriate reformulation of $\mathbf{X}^0$, $\mathbf{u}$ and $\mathbf{v}$, including the cases when gross outputs by industry, economy-wide totals of net taxes on products and/or of imports, total intermediate uses, or total final uses are not known. We do not show the corresponding $\mathbf{X}^0$, $\mathbf{u}$ and $\mathbf{v}$ under the latter (and other) circumstances as by now – after the above formulation of SUTs at basic prices – it should be straightforward to make the corresponding changes.

All in all, the SUT-RAS method is a particular case of the GRAS approach. The practical advantage of the SUT-RAS method is or might be that the main components of SUTs are used individually and there is no need to put all of them within one integrated matrix framework such as $\mathbf{X}^0$ above with its corresponding row- and column-sums constraints, including changing the signs of certain components of SUTs.[17]

If there is additional information on (non-overlapping) aggregation values of certain parts of national SUTs, then it is generally preferable to use MR-GRAS instead of GRAS or SUT-RAS on the ground that incorporating existing information on new SUTs often improves the final estimates. The MR-GRAS is particularly useful for updating inter-

[16] The SUT-RAS basic price framework given in footnote 24 in TT (p. 880) can be "extended" straightforwardly to *separately* include domestic and imported uses by a little reformulation of the expanded benchmark make and use matrices only. The latter and the exogenously given expanded total outputs and total uses (as in TT) are, respectively:

$$\overline{\mathbf{V}}^0 = \begin{bmatrix} \mathbf{V} & \mathbf{O} & \mathbf{0} \\ \mathbf{0}' & \mathbf{m}' & 0 \\ \mathbf{0}' & \mathbf{0}' & t_{tot} \end{bmatrix}, \quad \overline{\mathbf{U}}^0 = \begin{bmatrix} \mathbf{U}_b^d & \mathbf{Y}_b^d \\ \mathbf{U}_b^m & \mathbf{Y}_b^m \\ \mathbf{t}'_u & \mathbf{t}'_y \end{bmatrix}, \quad \overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ m_{tot} \\ t_{tot} \end{bmatrix} \text{ and } \overline{\mathbf{u}} = \begin{bmatrix} \mathbf{x} - \mathbf{w} \\ \mathbf{y}_f \end{bmatrix}.$$

Then one can readily use Theorem 1 and the corresponding three multipliers' expressions (equations 8-10) in TT to obtain the SUTs at basic prices. The multipliers of all components are also readily available: e.g. the first two (consecutive) p-dimensional vectors in $\mathbf{r}_u$ in TT (the row multiplier of $\overline{\mathbf{U}}$) are, respectively, the row multipliers of the domestic and imported uses, while the last element of $\mathbf{r}_u$ is the multiplier of taxes less subsidies on products. Alternatively, with little change of matrices of domestic uses and make, one could use the detailed purchaser price SUTs framework in TT (Section 2.4) to derive all the components of SUTs in basic prices, including net taxes on products by use categories.

[17] It should be noted that we could have opted for a second formulation of $\mathbf{X}^0$, $\mathbf{u}$ and $\mathbf{v}$ above by changing the signs of all its components (from minus to plus, and from plus to minus), which is however irrelevant for the final estimates of the SUT components. This is related to footnote 16 in TT (p. 873) where alternative SUT-RAS expressions are presented.

regional/national or global SUTs. Consider a global SUTs framework in basic prices, which distinguishes between the origin and destination countries of the intermediate and final uses (i.e., the "purchases only" member of the family of interregional SUTs in Oosterhaven, 1984). Such a SUT with c countries (regions), within the MR-GRAS setting, can be compactly written as:

$$\mathbf{X}^0 = \left[\begin{array}{cccc|cccc|cccc} -\mathbf{S}^1 & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{U}^{11} & \mathbf{U}^{12} & \cdots & \mathbf{U}^{1c} & \mathbf{Y}^{11} & \mathbf{Y}^{12} & \cdots & \mathbf{Y}^{1c} \\ \mathbf{O} & -\mathbf{S}^2 & \cdots & \mathbf{O} & \mathbf{U}^{21} & \mathbf{U}^{22} & \cdots & \mathbf{U}^{2c} & \mathbf{Y}^{21} & \mathbf{Y}^{22} & \cdots & \mathbf{Y}^{2c} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & -\mathbf{S}^c & \mathbf{U}^{c1} & \mathbf{U}^{c2} & \cdots & \mathbf{U}^{cc} & \mathbf{Y}^{c1} & \mathbf{Y}^{c2} & \cdots & \mathbf{Y}^{cc} \\ \hline \mathbf{0}' & \mathbf{0}' & \cdots & \mathbf{0}' & (\mathbf{t}_u^1)' & (\mathbf{t}_u^2)' & \cdots & (\mathbf{t}_u^c)' & (\mathbf{t}_u^1)' & (\mathbf{t}_y^2)' & \cdots & (\mathbf{t}_y^c)' \end{array}\right], \tag{13}$$

where $\mathbf{S}^r$ is the product-by-industry domestic supply matrix of country r, the ij-th element of $\mathbf{U}^{rs}$ indicates the amount of intermediate product i from country r used by industry j in country s, while the if-th element of $\mathbf{Y}^{rs}$ indicates the amount of final product i from country r used by final user f in country s, and the j-th element of $\mathbf{t}_u^s$ (resp. f-th entry of $\mathbf{t}_y^s$) denotes total taxes less subsidies on products paid by intermediate user j (resp. final user f) in country s. Note that compared to (12) imports from the Rest of the World $\mathbf{m}$ are absent here.

The MR-GRAS row and column sums constraints corresponding to (13) then take the form:

$$\mathbf{u}' = \begin{bmatrix} \mathbf{0}'_{2c} & t_{world} \end{bmatrix} \quad \text{and}$$

$$\mathbf{v}' = \left[\begin{array}{cccc|cccc|cccc} -(\mathbf{x}^1)' & -(\mathbf{x}^1)' & \cdots & -(\mathbf{x}^c)' & (\mathbf{u}^1)' & (\mathbf{u}^1)' & \cdots & (\mathbf{u}^c)' & (\mathbf{y}_f^1)' & (\mathbf{y}_f^1)' & \cdots & (\mathbf{y}_f^c)' \end{array}\right],$$

where t_{world} is the value of net taxes on products at the world level, and $\mathbf{u}^c$ and $\mathbf{y}_f^c$ denote, respectively, the vectors of total intermediate inputs and total final uses at purchasers' prices in country c. In such a setting, imposing non-overlapping aggregation constraints is particularly relevant, especially to ensure that the different (parts of) components of SUTs of each individual country sum to the corresponding aggregate value. This information then needs to be incorporated in the aggregator matrices $\mathbf{G}$ and $\mathbf{Q}$ with the corresponding aggregation values included in $\mathbf{W}$, also when there is only partial information available.

Assume one has information on $\boldsymbol{\iota}'\mathbf{U}^{rs}\boldsymbol{\iota} = u^{rs}$, $\boldsymbol{\iota}'\mathbf{Y}^{rs}\boldsymbol{\iota} = y^{rs}$, $\boldsymbol{\iota}'\mathbf{t}_u^s = t_u^s$ and $\boldsymbol{\iota}'\mathbf{t}_y^s = t_y^s$ for all r and all s. Then the corresponding aggregator matrices take the following form:

$$\underset{(c+1)\times(cp+1)}{\mathbf{G}} = \left[\begin{array}{c|c} \mathbf{I}_c \otimes \boldsymbol{\iota}'_p & \mathbf{0}_c \\ \hline \mathbf{0}'_{cp} & 1 \end{array}\right] \quad \text{and} \quad \underset{c(2s+f)\times(c+f)}{\mathbf{Q}} = \left[\begin{array}{c|c} \mathbf{O}_{cs\times c} & \mathbf{O}_{cs\times f} \\ \hline \mathbf{I}_c \otimes \boldsymbol{\iota}_s & \mathbf{O}_{cs\times f} \\ \hline \mathbf{O}_{cf\times c} & \mathbf{I}_c \otimes \boldsymbol{\iota}_f \end{array}\right],$$

where subscripts denote the dimension of the corresponding vector/matrix and $\otimes$ is the Kronecker product. In Section 4 we explain how to deal with cases of missing w_{IJ}'s.

In practice it is often the case that for each country only the overall total taxes less subsidies on products are known without the corresponding totals on intermediate and final uses, i.e. only t^s are given for each s but not its two individual components t_u^s and t_y^s (where $t^s = t_u^s + t_y^s$). In such a setting, the net taxes-related part of the global SUTs in (13) can be further expanded as follows:

$$\mathbf{X}^0 = \left[\begin{array}{cccc|cccc|cccc} -\mathbf{S}^1 & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{U}^{11} & \mathbf{U}^{12} & \cdots & \mathbf{U}^{1c} & \mathbf{Y}^{11} & \mathbf{Y}^{12} & \cdots & \mathbf{Y}^{1c} \\ \mathbf{O} & -\mathbf{S}^2 & \cdots & \mathbf{O} & \mathbf{U}^{21} & \mathbf{U}^{22} & \cdots & \mathbf{U}^{2c} & \mathbf{Y}^{21} & \mathbf{Y}^{22} & \cdots & \mathbf{Y}^{2c} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & -\mathbf{S}^c & \mathbf{U}^{c1} & \mathbf{U}^{c2} & \cdots & \mathbf{U}^{cc} & \mathbf{Y}^{c1} & \mathbf{Y}^{c2} & \cdots & \mathbf{Y}^{cc} \\ \hline \mathbf{0}' & \mathbf{0}' & \cdots & \mathbf{0}' & (\mathbf{t}_u^1)' & \mathbf{0}' & \cdots & \mathbf{0}' & (\mathbf{t}_u^1)' & \mathbf{0}' & \cdots & \mathbf{0}' \\ \mathbf{0}' & \mathbf{0}' & \cdots & \mathbf{0}' & \mathbf{0}' & (\mathbf{t}_u^2)' & \cdots & \mathbf{0}' & \mathbf{0}' & (\mathbf{t}_y^2)' & \cdots & \mathbf{0}' \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}' & \mathbf{0}' & \cdots & \mathbf{0}' & \mathbf{0}' & \mathbf{0}' & \cdots & (\mathbf{t}_u^c)' & \mathbf{0}' & \mathbf{0}' & \cdots & (\mathbf{t}_y^c)' \end{array}\right], \quad (14)$$

which ensures that the known country-level totals of net taxes on products are explicitly incorporated in the corresponding row sums vector $\mathbf{u}$.

All in all, different formulations of MR-GRAS settings for national, inter-regional or global SUTs are possible, which depend on what components of the new SUTs are known or not known, and/or on the aims of the researcher.

4 A guide to MR-GRAS implementation in practice

In this section we explain how to implement MR-GRAS in practice for cases of exhaustive and non-exhaustive constraints through a worked example, making use of our MR-GRAS program written in MATLAB language. The code is given in Appendix B, and can be directly (and easily after going through this section) be used by anyone interested in MRGRASing, or can be adopted by users of other similar software.

Consider a hypothetical simple benchmark table presented in Table 1. This table was generated randomly in MATLAB by entering at the command line of MATLAB command window the following statement: `X0=randi([-50,100],6)`. The latter command returns a 6-by-6 matrix containing pseudorandom integer values drawn from the discrete uniform distribution within the interval ranging from -50 to 100. Although, due to the random locations of negatives, this example may not literally be interpreted as a "3-sector and 2-region IOT", it fully suffices for our purposes of demonstration of some of wide range of possibilities of MR-GRAS applications.

Table 1: The benchmark 3-sector, 2-region "intermediate transactions" table, $\mathbf{X}^0$

		Region A			Region B			Total sales of outputs
		sec1	sec2	sec3	sec1	sec2	sec3	
Region A	Sector 1 (sec1)	63	9	14	9	-18	75	152
	Sector 2 (sec2)	-14	53	-10	66	69	66	230
	Sector 3 (sec3)	16	56	-21	9	93	-25	128
Region B	sec1	53	16	74	72	-1	80	294
	sec2	4	-48	14	64	51	99	184
	sec3	61	-1	84	6	16	27	193
Total uses of inputs		183	85	155	226	210	322	1181

To get the national (aggregate) intermediate flows, the following aggregator matrices need to be created:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Such aggregator matrices for large-scale IOTs can easily be generated in MATLAB using its built-in function of Kronecker product. In this simple case, the above matrices were created as follows: `G = kron([1 1],eye(3))` and `Q = kron([1;1],eye(3))`. Since we consider only non-overlapping aggregation constraints, the column sums of $\mathbf{G}$ and the row sums of $\mathbf{Q}$ must be all unity which is the case here, i.e. $\boldsymbol{\iota}'\mathbf{G} = \boldsymbol{\iota}'$ and $\mathbf{Q}\boldsymbol{\iota} = \boldsymbol{\iota}$. Note also that the typical elements of $\mathbf{G}$ and $\mathbf{Q}$ are, respectively, g_{Ii} and q_{jJ} so that one can readily see from the aggregator matrices which disaggregate items or sectors i's (resp. j's) belong to a certain aggregation index I (resp. J).

The national intermediate flow matrix in the base year then can be obtained from:

$$\mathbf{W}^0 = \mathbf{G}\mathbf{X}^0\mathbf{Q} = \begin{bmatrix} 197 & 6 & 243 \\ 120 & 125 & 169 \\ 92 & 164 & 65 \end{bmatrix}.$$

4.1 Updating with exhaustive constraints

Let us now update the random base-year inter-regional IOT, $\mathbf{X}^0$, to the following new row sums, column sums and non-overlapping aggregation constraints (or new national IOT):

$$\mathbf{u} = \begin{bmatrix} 160 \\ 194 \\ 145 \\ 320 \\ 134 \\ 151 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 197 \\ 71 \\ 151 \\ 242 \\ 178 \\ 265 \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} 230 & 0 & 250 \\ 123 & 75 & 130 \\ 86 & 174 & 36 \end{bmatrix}.$$

Before running the MR-GRAS, we need to assure the consistency of all the constraints among themselves and with the benchmark IOT. Following the recommendations made in Section 2.1, we confirm that indeed the requirements of identical aggregate row sums (i.e. $\mathbf{G}\mathbf{u} = \mathbf{W}\boldsymbol{\iota}$) and identical aggregate column sums (i.e. $\mathbf{v}'\mathbf{Q} = \boldsymbol{\iota}'\mathbf{W}$) hold true, thus the above three constraints are mutually consistent. Next, in checking the consistency of constraints with $\mathbf{X}^0$ in Table 1, note that we have imposed a zero aggregate value, $w_{12} = 0$. The corresponding i's and j's are: for $I = 1$ from $\mathbf{G}$ we find ones in $i = \{1, 4\}$ and for $J = 2$ from $\mathbf{Q}$ the constrained disaggregate indices are $j = \{2, 5\}$. Thus for $w_{12} = 0$ to be achievable, in all the corresponding transactions flowing from sector 1 (sec1) to sector 2 (sec2) we should have, at least, one positive and one negative entry.[18] These base-year elements are: $x^0_{12} = 9$ and $x^0_{45} = -1$ (i.e. domestic sec1-to-sec2 intermediate flows in regions A and B, respectively), $x^0_{15} = -18$ and $x^0_{42} = 16$ (i.e. inter-regional sec1-to-sec2 flows from A to

[18] Recall our discussions on strong vs. weak sign preservation in Section 2.3. Ignoring for now the uninteresting case of weak sign preservation (but covered in Section 4.5), there is the necessity of having both negative and positive entries if elements are be aggregated to a zero value.

B and from B to A, respectively). As there are two negative and two positive benchmark elements, then in principle it is feasible to constrain them to sum to zero in the new, updated IOT. Similarly, other aggregate values feasibility could be checked vis-à-vis the base-year IOT.

To run the MR-GRAS code, first make sure that the file `mrgras.m` is stored in the current working directory. Then, in case only the updated matrix $\mathbf{X}$ is needed, type at the command line the following statement:

```
>>  X = mrgras(X0,u,v,G,Q,W)
```

If additionally all the three multipliers r_i's, s_j's and t_{IJ}'s are required, enter instead:

```
>>  [X,r,s,T] = mrgras(X0,u,v,G,Q,W)
```

The notations of all the input and output vectors/matrices are entirely consistent with those used in this paper (except for not being bold for obvious reasons). One can also add a seventh input argument – a small positive scalar after `W` – indicating the algorithm convergence tolerance level; if empty (as above), the default threshold of `eps=0.1e-5` (=0.000001) is used automatically. The possibility of choosing your own stopping criteria is important because in cases when the program does not converge (the number of iterations will be displayed in the command window when the code is running), one can choose a higher threshold level in order to explore and find out the problem (if an approximate solution is reached), e.g. by looking closer into the sectors with relative extreme values of multipliers. In other cases, the result obtained at a higher threshold level might be perfectly acceptable for the purposes at hand.

If we use the default threshold value (which is also used in all the follow-up exercises), the MR-GRAS algorithm converges in 11 iterations; the resulting new matrix, whose entries are rounded to one decimal place, is shown in Table 2. The mean absolute percentage error (MAPE) and weighted absolute percentage error (WAPE) indicators (see e.g. Temurshoev et al., 2011) of the new matrix $\mathbf{X}$ vs. the old matrix $\mathbf{X}^0$ equal, respectively, 14.7 and 14.9. This difference largely reflects the variability dictated by the exogenously specified row sums, column sums and aggregation constraints: MAPE and WAPE are, respectively, equal to 15.3 and 15.2 for $\mathbf{u}$ vs. $\mathbf{u}^0$, 11.1 and 11.6 for $\mathbf{v}$ vs. $\mathbf{v}^0$, and 26.9 and 15.5 for $\mathbf{W}$ vs. $\mathbf{W}^0$.

The resulting row sums multipliers $\mathbf{r}$, column sums multipliers $\mathbf{s}$, aggregation multipliers $\mathbf{T}$, substitution factors $\mathbf{r}^*$, fabrication factors $\mathbf{s}^*$ and the technology or regional effects $\mathbf{T}^*$, all rounded to three decimal places, are given below. The latter three starred multipliers are obtained using the scaling equations in (8), assuming that $\mathbf{X}^0$ is a use matrix and ignoring the problem/plausibility of having negative numbers within the Use

Table 2: The updated matrix, $\mathbf{X}$: The case of exhaustive constraints

		Region A			Region B			Total sales of outputs
		sec1	sec2	sec3	sec1	sec2	sec3	
Region A	sec1	74.2	8.2	16.4	10.6	-21.5	72.1	160.0
	sec2	-13.4	44.4	-10.4	68.5	52.8	52.2	194.0
	sec3	18.8	64.8	-19.3	10.5	98.3	-28.0	145.0
Region B	sec1	61.7	14.5	85.5	83.5	-1.2	76.0	320.0
	sec2	4.0	-59.6	12.9	63.9	37.5	75.3	134.0
	sec3	51.7	-1.2	65.9	5.1	12.2	17.4	151.0
Total uses of inputs		197.0	71.0	151.0	242.0	178.0	265.0	1104.0

table.

$$\mathbf{r} = \begin{bmatrix} 1.114 \\ 0.932 \\ 1.146 \\ 1.101 \\ 0.897 \\ 0.827 \end{bmatrix}, \quad \mathbf{r}^* = \begin{bmatrix} 1.111 \\ 0.930 \\ 1.143 \\ 1.098 \\ 0.894 \\ 0.825 \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 1.071 \\ 0.941 \\ 1.012 \\ 1.066 \\ 0.860 \\ 0.832 \end{bmatrix}, \quad \mathbf{s}^* = \begin{bmatrix} 1.074 \\ 0.943 \\ 1.015 \\ 1.069 \\ 0.862 \\ 0.834 \end{bmatrix},$$

$$\text{and} \quad \mathbf{T}^* = \mathbf{T} = \begin{bmatrix} 0.988 & 0.874 & 1.037 \\ 1.044 & 0.954 & 1.019 \\ 0.956 & 1.072 & 0.937 \end{bmatrix}.$$

In this particular case normalization (8) does not really change the meaning (and size) of the multipliers effects. This is because the weighted harmonic mean of the non-normalized multipliers is already quite close to one, $\bar{r}_H = 1.0026$. However, in principle, one cannot rule out that the difference between the normalized and non-normalized factors could be significant even qualitatively, e.g. it might be the case that $r_i > 1$ but $r_i^* < 1$.

4.2 Non-exhaustive (non-overlapping) aggregation constraints

Sometimes, if not often, it may be the case that certain w_{IJ}'s (but not all) are unknown and the researcher decides not to make uninformed guesses with respect to these missing totals. This case can be also handled easily within the MR-GRAS approach: technically for such aggregate sets $\{I, J\}$'s no aggregation constraint is imposed. To implement this case in our MR-GRAS code, in place of the corresponding missing aggregate total(s) write the number 1010101. We have chosen this rather ad hoc number because: (1) one cannot replace the unknown elements of $\mathbf{W}$ by a zero since the latter can be a valid constraint total as was the case in the previous example with exhaustive constraints, and (2) the chance that in real-life applications the number 1010101 will be indeed an aggregation value is quite low.[19]

As an example, assume that $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{W}$ remain the same is in Section 4.1, except now w_{12}, w_{21}, w_{22}, w_{23} and w_{32} are unknown. Then, to use our MR-GRAS function, the new national IOT totals should be defined as follows:

$$\mathbf{W} = \begin{bmatrix} 230 & 1010101 & 250 \\ 1010101 & 1010101 & 1010101 \\ 86 & 1010101 & 36 \end{bmatrix}.$$

In this case, the MR-GRAS procedure gives *exactly the same result* $\mathbf{X}$ that is presented in Table 2, but convergence is achieved in 114 iterations vs. 11 iterations when all aggregate values were available.[20] This is not at all surprising because $\mathbf{u}$ and $\mathbf{v}$ already determine the aggregate row sums and aggregate column sums of $\mathbf{W}$ (recall the constraints' mutual consistency requirements). Given the above configuration of known aggregate values, all the unknown elements in $\mathbf{W}$ are recovered accurately within the MR-GRAS procedure as its solution already implicitly reflects the constraints' consistency requirements.[21] So this simple example shows that *to have exhaustive aggregation constraints, it is not always necessary to have access to all the individual values of these constraints*: if the configuration of the known new aggregate values allows, all the missing w_{IJ}'s will be recovered accurately, provided that both $\mathbf{u}$ and $\mathbf{v}$ represent the "true" disaggregate constraints.

However, in this case the multipliers will not equal to their counterparts found earlier in Section 4.1. The reason is that within the MR-GRAS code, the regional effects t_{IJ}'s corresponding to the missing aggregation values are all set to unity so that all the entries within such aggregate sets are effectively bi-proportionally adjusted akin to the standard GRAS procedure. Therefore, if the focus is on MR-GRAS multipliers and there are missing aggregation values (potentially exhaustive or non-exhaustive), then we suggest to

[19] However, if the latter turns out to be the case, then simply replace this number in few (namely, three) places within the code by any other "magical number" of your choice that does not show up in your aggregation constraints.

[20] In terms of time difference, using a laptop with processor 2.7 GHz Intel Core i5, the *average elapsed time* over 1,000 MR-GRAS updating replications was 0.0549 seconds when updating with "non-exhaustive" aggregation constraints vs. 0.0061 seconds with exhaustive aggregation constraints.

[21] If this seems strange, here is how the missing values are recovered accurately in this particular case. If $w_{IJ} = 1010101$, then technically the MR-GRAS program does set the corresponding aggregation multiplier to one, i.e. $t_{IJ} = 1$ (thus no aggregation total is imposed for such pair of I and J). However, the constraints' identical aggregate row sums requirement, $\mathbf{W}\boldsymbol{\imath} = \mathbf{G}\mathbf{u} = [480, 328, 296]'$, allows computing w_{12} and w_{32} as residuals. Once these are known, the elements of the second row of $\mathbf{W}$ are recovered using the constraints' identical aggregate column sums requirement $\boldsymbol{\imath}'\mathbf{W} = \mathbf{v}'\mathbf{Q} = [439, 249, 416]$. Since these constraints' mutual consistency conditions hold for any MR-GRAS solution, the missing w_{IJ} in the $\mathbf{W}$ in question should be recovered accurately by construction, even without imposing the corresponding totals.

take the following three-step approach: (1) run the procedure with partially available aggregation values similar to the exercise performed above, (2) using $\mathbf{X}$ derived from the first step, recover all the missing aggregation values from $\mathbf{W} = \mathbf{GXQ}$, and (3) re-run the MR-GRAS with these exhaustive aggregation constraints. The result will be the same $\mathbf{X}$ as in step (1), but all the multipliers, including all t_{IJ}'s, will be obtained endogenously in full conformity with the MR-GRAS framework.

Consider an example of genuine non-exhaustive aggregation constraints. If all the entries of the first two columns of $\mathbf{W}$ are unknown, then it is no longer possible to recover these missing values such that they exactly coincide with their counterpart entries presented in Section 4.1. Then the solution is achieved in 17 iterations, and the resulting updated matrix is *different* from that in Table 2 (for space consideration this new matrix is not shown here). The MAPE and WAPE of the latter new matrix vs. that in Table 2 are found to equal, respectively, 3.68 and 2.19. The national IOT, recovered from the latter updated matrix $\mathbf{X}$, becomes:

$$\mathbf{GXQ} = \begin{bmatrix} 226.79 & 3.21 & 250 \\ 119.78 & 78.22 & 130 \\ 92.44 & 167.56 & 36 \end{bmatrix},$$

which confirms the fact that under the latter configuration of the missing w_{IJ}'s, it is impossible to accurately recover their "true" values as given in $\mathbf{W}$ in Section 4.1. One would expect that in real-world applications such cases are encountered most often.

Finally, if we set $w_{IJ} = 1010101$ for all I and all J, i.e. consider the case of no aggregation constraints, the results of MR-GRAS boils down to that of standard GRAS (hence our MATLAB code could be equally used for GRASing purposes; in this case the GRAS outcome was obtained in 9 iterations). The MAPE and WAPE between GRASed $\mathbf{X}$ and MR-GRASed $\mathbf{X}$ presented in Table 2 are found to equal to 4.87 and 3.17, respectively. Comparing the latter with the values of these indicators reported in the previous paragraph, we obtain the usual finding in the literature that incorporating additional accurate information on the new matrix – here its partial known aggregate values – improves the resulting estimates of the updated matrix.

4.3 Non-exhaustive row and/or column sums

In Section 2.1 we have touched upon the feasibility of MR-GRASing under the circumstance of missing *all* the disaggregate row and column totals, but known (possibly partial) aggregation values. Here we generalize this case, allowing for arbitrary missing values in the row and column totals of, respectively, $\mathbf{u}$ and $\mathbf{v}$. Decompose these constraints into $\mathbf{u} = \mathbf{u}_a + \mathbf{u}_n$ and $\mathbf{v} = \mathbf{v}_a + \mathbf{v}_n$, where subscripts a and n refer to available and non-available data; e.g. $\mathbf{u}_a$ includes all the known row sum totals with zeros placed for missing u_i's. Now keep exactly the same structure of zero and non-zero values in the corresponding original row and column totals of $\mathbf{u}^0 = \mathbf{u}_a^0 + \mathbf{u}_n^0$ and $\mathbf{v}^0 = \mathbf{v}_a^0 + \mathbf{v}_n^0$. Then, the benchmark matrix can be redefined as follows:

$$\overline{\mathbf{X}}^0 = \begin{bmatrix} \mathbf{X}^0 & -\mathbf{u}_n^0 \\ -(\mathbf{v}_n^0)' & \boldsymbol{\imath}'\mathbf{v}_n^0 \end{bmatrix}.$$

Thus, the new row- and column-sums vectors for MR-GRAS purposes should take the following forms:

$$\overline{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_a \\ 0 \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{v}} = \begin{bmatrix} \mathbf{v}_a \\ \boldsymbol{\imath}'\mathbf{u}_a - \boldsymbol{\imath}'\mathbf{v}_a \end{bmatrix},$$

where the choice of the last element of $\overline{\mathbf{v}}$ is dictated by our choice of the benchmark matrix $\overline{\mathbf{X}}^0$, i.e. the sum of its last column is equal to $-\boldsymbol{\imath}'\mathbf{u}_n^0 + \boldsymbol{\imath}'\mathbf{v}_n^0 = -(x^0 - \boldsymbol{\imath}'\mathbf{u}_a^0) + (x^0 - \boldsymbol{\imath}'\mathbf{v}_a^0) = \mathbf{v}_a^0 - \mathbf{u}_a^0$.

The expanded (by one column and one row) aggregator matrices $\overline{\mathbf{G}}$ and $\overline{\mathbf{Q}}$ have the same form as presented in Section 2.1, which together with the constraints' mutual consistency requirements imply the following structure of the *new* aggregation values:

$$\overline{\mathbf{W}} = \overline{\mathbf{G}}\,\overline{\mathbf{X}}\,\overline{\mathbf{Q}} = \begin{bmatrix} \mathbf{W} & -\mathbf{G}\mathbf{u}_n \\ -(\mathbf{v}_n)'\mathbf{Q} & \boldsymbol{\imath}'\mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{W} & \mathbf{G}\mathbf{u}_a - \mathbf{W}\boldsymbol{\imath} \\ (\mathbf{v}_a)'\mathbf{Q} - \boldsymbol{\imath}'\mathbf{W} & \boldsymbol{\imath}'\mathbf{W}\boldsymbol{\imath} - \boldsymbol{\imath}'\mathbf{v}_a \end{bmatrix}. \tag{15}$$

In implementing MR-GRAS, it is important to treat zeros in the last row and last column of $\overline{\mathbf{W}}$ in (15) as missing values (these correspond to available row and column sums).

Assume that the row and column totals for sectors 2 and 3 in both regions are unknown, while the values of the remaining elements in $\mathbf{u}$ and $\mathbf{v}$ and of the aggregation constraints $\mathbf{W}$ are exactly the same as given in Section 4.1. Then the redefined expanded constraints will take the form:

$$\overline{\mathbf{u}} = \begin{bmatrix} 160 \\ 0 \\ 0 \\ 320 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \overline{\mathbf{v}} = \begin{bmatrix} 197 \\ 0 \\ 0 \\ 242 \\ 0 \\ 0 \\ 41 \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{W}} = \begin{bmatrix} 230 & 0 & 250 & 1010101 \\ 123 & 75 & 130 & -328 \\ 86 & 174 & 36 & -296 \\ 1010101 & -249 & -416 & 665 \end{bmatrix}.$$

The expanded benchmark table takes the following form (compare it with the base-year IOT in Table 1):

$$\overline{\mathbf{X}}^0 = \begin{bmatrix} 63 & 9 & 14 & 9 & -18 & 75 & 0 \\ -14 & 53 & -10 & 66 & 69 & 66 & -230 \\ 16 & 56 & -21 & 9 & 93 & -25 & -128 \\ 53 & 16 & 74 & 72 & -1 & 80 & 0 \\ 4 & -48 & 14 & 64 & 51 & 99 & -184 \\ 61 & -1 & 84 & 6 & 16 & 27 & -193 \\ 0 & -85 & -155 & 0 & -210 & -322 & 772 \end{bmatrix}.$$

Then running MR-GRAS on $\overline{\mathbf{X}}^0$ to conform to the expanded constraints $\overline{\mathbf{u}}$, $\overline{\mathbf{v}}$ and $\overline{\mathbf{W}}$ results in $\overline{\mathbf{X}}$, whose last row and last column recover the missing disaggregate totals (taken in absolute value). This new matrix is presented in Table 3 below.

Table 3: The updated matrix, $\mathbf{X}$: The case of non-exhaustive disaggregate constraints

		Region A			Region B			Total sales of outputs
		sec1	sec2	sec3	sec1	sec2	sec3	
Region A	sec1	72.6	8.1	14.3	10.4	-21.5	76.1	160.0
	sec2	-14.0	42.4	-12.7	66.4	51.4	51.5	184.9
	sec3	14.3	61.9	-26.3	8.1	95.6	-31.5	122.1
Region B	sec1	62.1	14.6	76.9	84.9	-1.2	82.6	320.0
	sec2	4.1	-58.0	11.4	66.5	39.2	79.9	143.1
	sec3	57.8	-0.9	71.1	5.7	17.4	22.7	173.9
Total uses of inputs		197.0	68.1	134.6	242.0	180.9	281.4	1104.0

In this case, with our default threshold level, the MR-GRAS iterative procedure converged in 10 iterations. The MAPE and WAPE of the new IOT in Table 3 vs. the new IOT in Table 2 that is based on exhaustive constraints were found to be equal to 9.91 and 6.54, respectively.

4.4 Non-exhaustive disaggregate and aggregation constraints

In the previous two subsections we have covered the cases of non-exhaustive aggregation constraints and non-exhaustive row/column totals separately. It is also possible to MR-GRAS when *both* of these constraints include missing values. Consider the same case of unknown elements in the row and column totals as in Section 4.3, but additionally assume that aggregation values of the first two columns of $\mathbf{W}$ are missing (which is again an arbitrary choice). Therefore, for MR-GRAS purposes we use the same $\overline{\mathbf{X}}^0$, $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$ as in Section 4.3, but now the expanded aggregation constraints matrix, using (15), takes the following form:

$$\overline{\mathbf{W}} = \begin{bmatrix} 1010101 & 1010101 & 250 & 1010101 \\ 1010101 & 1010101 & 130 & 1010101 \\ 1010101 & 1010101 & 36 & 1010101 \\ 1010101 & 1010101 & -416 & 1010101 \end{bmatrix}.$$

In this case, the MR-GRAS convergence is obtained in 29 iterations (the average elapsed time over 1,000 runs of the updating procedure was only 0.0196 seconds). The obtained new IOT is shown in Table 4. Aggregation of the resulting $\overline{\mathbf{X}}$ gives

$$\overline{\mathbf{G}}\,\overline{\mathbf{X}}\,\overline{\mathbf{Q}} = \begin{bmatrix} 219.11 & 10.89 & 250 & 0 \\ 124.96 & 134.75 & 130 & -389.72 \\ 94.93 & 169.92 & 36 & -300.85 \\ 0 & -315.56 & -416 & 731.56 \end{bmatrix},$$

thus the known aggregation values (or national IOT entries) are satisfied perfectly.

Table 4: The new matrix, $\mathbf{X}$: The case of non-exhaustive disaggregate and non-exhaustive aggregation constraints

		Region A			Region B			Total sales of outputs
		sec1	sec2	sec3	sec1	sec2	sec3	
Region A	sec1	67.7	9.8	14.0	9.8	-16.2	74.9	160.0
	sec2	-13.9	54.1	-12.7	67.6	72.0	51.8	218.9
	sec3	16.1	57.0	-25.9	9.2	96.7	-30.9	122.2
Region B	sec1	59.6	18.2	77.6	82.0	-0.9	83.5	320.0
	sec2	4.1	-45.9	11.3	67.1	54.5	79.6	170.8
	sec3	63.3	-1.0	70.3	6.3	17.2	22.5	178.7
Total uses of inputs		197.0	92.3	134.7	242.0	223.3	281.3	1170.6

The MAPE and WAPE indicators of this new matrix vs. that obtained under the exhaustive constraints in Table 2 are, respectively, 15.73 and 11.74. These much larger differences – compared to all the cases discussed so far, including the GRAS outcome – is to be expected since the considered example includes many missing values for both aggregation constraints *and* disaggregate row and column sums constraints. Note also that the overall sum of the new IOT (i.e. 1170.6) now deviates from the new total of 1140 that was satisfied in all the previous settings due to the use of exhaustive aggregation or/and disaggregate constraints.

4.5 The case of weak sign-preservation

For completeness sake, let us finally consider an uninteresting case of weak sign preservation when positive elements are constrained to sum to zero. Let us focus on the case of zero aggregation constraint. Based on Table 1 one might want to impose $w_{13} = 0$ since the underlying sec3-to-sec1 transactions are all positive. In order to have constraints' mutual consistency, the corresponding totals in **u** and **v** also need to be adjusted. Hence, the constraints in Section 4.1 maybe changed into (the modified values are given in bold):

$$\mathbf{u} = \begin{bmatrix} 160 \\ 194 \\ \mathbf{102} \\ 320 \\ 134 \\ \mathbf{108} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{154} \\ 71 \\ 151 \\ \mathbf{199} \\ 178 \\ 265 \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} 230 & 0 & 250 \\ 123 & 75 & 130 \\ \mathbf{0} & 174 & 36 \end{bmatrix}.$$

Under this scenario, the MR-GRAS converges in 13 iterations, whose results are presented in Table 5 below. Not surprisingly, the positive elements constrained to sum to zero now all reduce to exactly zeros (these are highlighted in bold).

Table 5: The new matrix, **X**: The case of weak sign-preservation

		Region A			Region B			Total sales of outputs
		sec1	sec2	sec3	sec1	sec2	sec3	
Region A	sec1	82.3	8.0	15.0	7.6	-22.3	69.4	160.0
	sec2	-9.1	44.3	-10.9	65.0	52.3	52.3	194.0
	sec3	**0**	63.3	-23.9	**0**	95.4	-32.9	102.0
Region B	sec1	74.8	15.4	85.7	65.3	-1.1	80.0	320.0
	sec2	6.0	-59.2	12.5	61.1	37.5	76.1	134.0
	sec3	**0**	-0.9	72.6	**0**	16.1	20.2	108.0
Total uses of inputs		154.0	71.0	151.0	199.0	178.0	265.0	1018.0

The MAPE and WAPE measures of this new matrix vs. the matrix obtained under the strong sign-preservation assumption in Table 2 are found to be, respectively, 20.79 and 12.83. These are even larger than the respective differences found in the previous section with extensive missing constraints' values. The bottom line here is clear: if a zero aggregation constraint is introduced by mistake, the updated IOT/SUT/SAM might very well significantly deviate from (and change the structure of) the benchmark table. An extra caution is, therefore, necessary when introducing zero aggregation values within the MR-GRAS setting.

As mentioned earlier, within IOT/SUT/SAM setting there is no particular reason to constrain a set of old positive numbers to zero. However, if such cases arise, then one can simply (and logically) set the relevant elements to zero in the benchmark table. In the case considered here, one would have obtained exactly the same matrix as given in Table 5, if the constrained original transactions were all set to zero, i.e. $x^0_{31} = x^0_{34} = x^0_{61} = x^0_{64} = 0$. As such this case is of no practical importance in updating, whereas if one wants to constrain a set of positive numbers to zero, it is evident that the only solution is to nullify these entries.

5 Conclusion

In this paper we have presented the MR-GRAS method, which is an extension of the generalized RAS technique to a multi-regional (MR) setting, where "multi-regional" does not have to be understood only in its literal sense. The framework is applicable to updating/balancing/constructing regional, national, inter-regional and global input-output (IO) tables, supply and use tables (SUTs), social accounting matrices (SAMs), or generally any partitioned matrix that needs to conform the new row sums , column sums and non-overlapping aggregation constraints. The focus on only non-overlapping aggregation constraints is motivated by: (a) the aim of maintaining the simplicity and transparency properties of the GRAS technique in its "multi-regional" setting, and (b) the fact that in practice such settings are the cases practitioners come across most often. For example, within a MRIO setting the aggregation constraints will guarantee that the unknown inter- and intra-regional sectoral transactions are consistent with (i.e. add up to) the relevant exogenously specified aggregate (e.g. national accounts and aggregate trade) data. Moreover, updating regional, national, inter-regional/country or global SUTs very often can be easily formulated in a setting that is fully consistent with the MR-GRAS framework.

We derived the complete analytical solution of the method and proposed a simple iterative algorithm for its computation. Importantly, the MR-GRAS method also allows for *non-exhaustive* disaggregate (i.e. row- and/or colum-sums) and/or aggregation constraints. Having such flexibility is critical as in real life applications often not all the values of the three types of constraints are available, in which case the missing values are endogenously derived within the MR-GRAS updating procedure. We have elaborated in some detail on the main properties of the method, also to explain the popularity and attractiveness of the RAS-type balancing methods in practice. Further, we have discussed the issues of normalization and interpretation of MR-GRAS multipliers. From a wide range of possible MR-GRAS applications, we have examined a few updating settings, including national and global SUTs. Finally, in the last section a rather detailed guide on MR-GRAS implementation in practice is presented for IO settings with exhaustive and non-exhaustive constraints. This is demonstrated through a worked example, using our publicly available MR-GRAS code (see Appendix B) written in MATLAB programming language.

References

Almon, C.: 1968, 'Recent methodological advances in input-output in the United States and Canada'. Paper presented at the Forth International Conference on Input-Output Techniques in Geneva.

Bacharach, M.: 1970, *Biproportional Matrices and Input-Output Change*. Cambridge: Cambridge University Press.

Beutel, J.: 2002, 'The economic impact of objective 1 interventions for the period 2000-2006'. Report to the Directorate-General for Regional Policies, Konstanz.

Beutel, J.: 2008, 'An input-output system of economic accounts for the EU Member States'. Interim Report for Service Contract Number 150830-2007 FISC-D to European Commission, DirectoratefGeneral Joint Research Centre, Institute for Prospective Technological Studies.

Canning, P. and Z. Wang: 2005, 'A flexible mathematical programming model to estimate interregiional input-output accounts'. *Journal of Regional Science* **45**, 539–563.

Cardenete, M. A. and F. Sancho: 2004, 'Sensitivity of CGE simulation results to comptetting SAM updates'. *The Review of Regional Studies* **34**, 37–56.

de Mesnard, L.: 2002, 'Normalizing biproportional methods'. *Annals of Regional Science* **36**, 139–149.

de Mesnard, L.: 2004, 'On the idea of ex ante and ex post normalization of biproportional methods'. *Annals of Regional Science* **38**, 741–f749.

de Mesnard, L. and R. E. Miller: 2006, 'A note on added information in the RAS procedure: Reexamination of some evidence.'. *Journal of Regional Science* **46**, 517–f528.

Dietzenbacher, E. and R. E. Miller: 2009, 'RAS-ing the transactions or the coefficients: it makes no difference'. *Journal of Regional Science* **49**, 555–566.

Eurostat: 2008, *European Manual of Supply, Use and Input-Output Tables. Methodologies and Working Papers*. Luxembourg: Office for Official Publications of the European Communities.

Friedlander, D.: 1961, 'A technique for estimating a contingency table, given marginal totals and some supplemental data'. *Journal of the Royal Statitistical Society, Series A* **124**, 412–420.

Geschke, A., J. Ugon, M. Lenzen, K. Kanemoto, and D. D. Moran: 2019, 'Balancing and reconciling large multi-regional input-output databases using parallel optimisation and high-performance computing'. *Journal of Economic Structures* **8:2**.

Gilchrist, D. A. and L. V. St. Louis: 1999, 'Completing input-output tables using partial information, with an application to Canadian data'. *Economic Systems Research* **11**, 185–194.

Gilchrist, D. A. and L. V. St. Louis: 2004, 'An algorithm for the consistent inclusion of partial information in the revision of input-output tables'. *Economic Systems Research* **16**, 149–156.

Golan, A., G. Judge, and S. Robinson: 1994, 'Information from incomplete or partial multisectoral economic data'. *Review of Economics and Statistics* **76**, 541–549.

Golan, A. and S. J. Vogel: 2000, 'Estimation of non-stationary social accounting matrix coefficients with supply-side information'. *Economic Systems Research* **12**, 447–471.

Günlük-Şenesen, G. and J. M. Bates: 1988, 'Some experiments with methods of adjusting unbalanced data matrices'. *Journal of the Royal Statistical Society, Series A* **151**, 473–490.

Harrigan, F. and I. Buchanan: 1984, 'A quadratic programming approach to input-output estimation and simulation'. *Journal of Regional Science* **24**, 339–358.

Harthoorn, R. and J. van Dalen: 1987, 'On the adjustment of tables with Lagrange multipliers'. **NA-024**. Central Bureau of Statistics, The Netherlands, National Accounts Research Division.

Holý, V. and K. Šafr: 2017, 'The use of multidimensional RAS method in input-output matrix estimation'. published in `arXiv:1704.07814v1 [stat.AP]`.

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

Jackson, R. W. and A. T. Murray: 2004, 'Alternative input-output matrix updating formulations'. *Economic Systems Research* **16**, 135–148.

Junius, T. and J. Oosterhaven: 2003, 'The solution of updating or regionalizing a matrix with both positive and negative entries'. *Economic Systems Research* **15**, 87–96.

Kuroda, M.: 1988, 'A method of estimation for the updating transaction matrix in the input-output relationships'. In: K. Uno and S. Shishido (eds.): *Statistical Data Bank Systems. Socio-economic database and model building in Japan*. Amsterdam: North Holland, pp. 43–56.

Lahr, M. L. and L. de Mesnard: 2004, 'Biproportional techniques in input-output analysis: table updating and structural analysis'. *Economic Systems Research* **16**, 115–134.

Lenzen, M., B. Gallego, and R. Wood: 2009, 'Matrix balancing under conflicting information'. *Economic Systems Research* **21**, 23–44.

Lenzen, M., R. Wood, and B. Gallego: 2007, 'Some comments on the GRAS method'. *Economic Systems Research* **19**, 461–465.

Leontief, W. W.: 1941, *The Structure of American Economy, 1919-1929: An Empirical Application of Equilibrium Analysis*. Cambridge: Cambridge University Press.

Matuszewski, T., P. R. Pitts, and J. A. Sawyer: 1964, 'Linear programming estimates of changes in inputfoutput coefficients'. *Canadian Journal of Economics and Political Science* **30**, 203f–211.

McDougall, R. A.: 1999, 'Entropy theory and RAS are friends'. GTAP Working Paper no. 06.

Miller, R. E. and P. D. Blair: 2009, *Input-Output Analysis: Foundations and Extensions*. Cambridge: Cambridge University Press, 2nd edition.

Mínguez, R., J. Oosterhaven, and F. Escobedo: 2009, 'Cell-corrected RAS method (CRAS) for updating or regionalizing an input-output matrix'. *Journal of Regional Science* **49**, 329–348.

Motorin, V.: 2017, 'Enhancing the distance minimization methods of matrix updating within a homothetic paradigm'. *Economic Structures* **3:36**.

Oosterhaven, J.: 1984, 'A family of square and rectangular interregional input-output tables and models'. *Regional Science and Urban Economics* **14**, 565–582.

Oosterhaven, J.: 2005, 'GRAS versus minimizing absolute and squared differences: A comment'. *Economic Systems Research* **17**, 327–331.

Oosterhaven, J.: 2012, 'Adding supply-driven consumption makes the Ghosh model even more implausible'. *Economic Systems Research* **24**, 101–111.

Oosterhaven, J. and F. Escobedo-Cardeñoso: 2011, 'A new method to estimate input-output tables by means of structural lags, tested on Spanish regions'. *Papers in Regional Science* **90**(4), 829–844.

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

Rey Los Santos, L., K. Wojtowicz, M. Tamba, T. Vandyck, M. Weitzel, B. Saveyn, and U. Temursho: 2018, 'Global macroeconomic balances for mid-century climate analyses. Supplementary material to Global Energy and Climate Outlook 2018'. *JRC Technical Reports*. EUR 29490 EN, Publications Office of the European Union, Luxembourg.

Snower, D. J.: 1990, 'New methods of updating input-output matrices'. *Economic Systems Research* **2**, 27–38.

Stone, R.: 1961, *Input-Output and National Accounts*. Paris: Organization for European Economic Cooperation.

Strømman, A. H.: 2009, 'A multi-objective assessment of input-output matrix updating methods'. *Economic Systems Research* **21**, 81–88.

Temursho, U.: 2018, 'Entopy-based benchmarking methods'. *Statistica Neerlandica* **72**(4), 421–446.

Temursho, U.: 2019, 'On the EURO method'. *IOpedia Research Paper No. 3*. October 2019, www.IOpedia.eu.

Temurshoev, U., R. E. Miller, and M. C. Bouwmeester: 2013, 'A note on the GRAS method'. *Economic Systems Research* **25**(3), 361f–367.

Temurshoev, U. and M. P. Timmer: 2010, 'Joint estimation of supply and use tables'. *WIOD Working Paper Nr. 3*. http://www.wiod.org/publications/abstract?id=3.

Temurshoev, U. and M. P. Timmer: 2011, 'Joint estimation of supply and use tables'. *Papers in Regional Science* **90**(4), 863f–882.

Temurshoev, U., N. Yamano, and C. Webb: 2011, 'Projection of supply and use tables: methods and their empirical assessment'. *Economic Systems Research* **23**, 91–123.

Tilanus, C. B.: 1968, *Input-Output Experiments: the Netherlands, 1948-1961*. Rotterdam: Rotterdam University Press.

Toh, M.-H.: 1998, 'The RAS approach in updating input-output matrices: An instrumental variable interpretation and analysis of structural change'. *Environment and Planning A* **10**, 63–78.

United Nations: 2018, *Handbook on Supply, Use and Input-Output Tables with Extensions and Applications*. Studies in Methods, Series F No. 74, Rev.1, New York.

Valderas-Jaramillo, J. M. and J. M. Rueda-Cantuche: 2019, 'The multidimensional GRAS method: Applications for the projection of multiregional input-output frameworks and valuation matrices'. Paper presented at 27th International Input-Output Association Conference, Glasgow, Scotland.

Valderas-Jaramillo, J. M., J. M. Rueda-Cantuche, E. Olmedo, and J. Beutel: 2018, 'Projecting supply and use tables: new variants and fair comparisons'. *Economic Systems Research*. In press.

van der Linden, J. A. and E. Dietzenbacher: 2000, 'The determinants of structural change in the European Union: a new application of RAS'. *Environment and Planning A* **32**, 2205–2229.

van der Ploeg, F.: 1988, 'Balancing large systems of national accounts'. *Computer Science in Economics and Management* **1**, 31–39.

Zhao, X. and J. Squibb III: 2019, 'Modelling inter-regional trade flows: A new method based on Radiation model and multi-regional GRAS technique'. Paper presented at the 27-th International Input-Output Association Conference, Glasgow, Scotland.

A SUT-RAS as a particular case of GRAS

Without loss of generality, the integrated SUT matrix of the base table can be compactly written as follows:

$$\mathbf{X}^0 = \begin{bmatrix} -\overline{\mathbf{S}}_0 & \overline{\mathbf{U}}_0 \end{bmatrix}, \tag{A.1}$$

where $\overline{\mathbf{S}}_0$ and $\overline{\mathbf{U}}_0$ are the Supply and Use tables, whose configuration and contents can change depending on which components of SUTs need to be updated. For example, in case of national SUTs framework in basic prices (12) considered in Section 3 these would take the following form:

$$\overline{\mathbf{S}}_0 = \begin{bmatrix} \mathbf{S}_0^d & \mathbf{o} \\ \mathbf{O} & \mathbf{m}_0 \\ \mathbf{o}' & 0 \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{U}}_0 = \begin{bmatrix} \mathbf{U}_{b,0}^d & \mathbf{Y}_{b,0}^d \\ \mathbf{U}_{b,0}^m & \mathbf{Y}_{b,0}^m \\ \mathbf{t}_{u,0}' & \mathbf{t}_{y,0}' \end{bmatrix}.$$

The typical GRAS positive and negative decomposition of the base Supply and Use table can be written as $\overline{\mathbf{S}}_0 = \mathbf{P}_0^s - \mathbf{N}_0^s$ and $\overline{\mathbf{U}}_0 = \mathbf{P}_0^u - \mathbf{N}_0^u$. Negative elements in case of Supply table are possible if $\overline{\mathbf{S}}_0$ includes net taxes and/or trade and transport margins (see TT, Section 2.3), otherwise $\mathbf{N}_0^s = \mathbf{O}$ by construction. GRASing the integrated base SUT (A.1) to satisfy the new row and column constraints of, respectively, $\mathbf{u}$ and $\mathbf{v}$, results in the well-known GRAS solution:

$$\mathbf{X} = \hat{\mathbf{r}}\mathbf{P}^0\hat{\mathbf{s}} - \hat{\mathbf{r}}^{-1}\mathbf{N}^0\hat{\mathbf{s}}^{-1}, \tag{A.2}$$

where $\mathbf{X}^0 = \mathbf{P}^0 - \mathbf{N}^0$. Given that there is a minus sign in front of Supply table in (A.1), the matrices $\mathbf{P}^0$ and $\mathbf{N}^0$, consisting, respectively, of positive and negative elements of the base table $\mathbf{X}^0$, include the following SUTs components:

$$\mathbf{P}^0 = \begin{bmatrix} \mathbf{N}_0^s & \mathbf{P}_0^u \end{bmatrix} \quad \text{and} \quad \mathbf{N}^0 = \begin{bmatrix} \mathbf{P}_0^s & \mathbf{N}_0^u \end{bmatrix}.$$

Plugging the later expressions in (A.2), the updated matrix can be written as:

$$\mathbf{X} = \begin{bmatrix} -\overline{\mathbf{S}} & \overline{\mathbf{U}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{r}}\mathbf{N}_0^s\hat{\mathbf{s}}_v - \hat{\mathbf{r}}^{-1}\mathbf{P}_0^s\hat{\mathbf{s}}_v^{-1} & \hat{\mathbf{r}}\mathbf{P}_0^u\hat{\mathbf{s}}_u - \hat{\mathbf{r}}^{-1}\mathbf{N}_0^s\hat{\mathbf{s}}_u^{-1} \end{bmatrix},$$

where the column multiplier vector was separated into its Supply- and Use-related multipliers, i.e. $\mathrm{s} = [\mathrm{s}_v', \mathrm{s}_u']'$. Thus the updated Use and Supply matrices are derived as follows:

$$\overline{\mathbf{U}} = \hat{\mathbf{r}}\mathbf{P}_0^u\hat{\mathbf{s}}_u - \hat{\mathbf{r}}^{-1}\mathbf{N}_0^s\hat{\mathbf{s}}_u^{-1}, \tag{A.3}$$

$$\overline{\mathbf{S}} = \hat{\mathbf{r}}^{-1}\mathbf{P}_0^s\hat{\mathbf{s}}_v^{-1} - \hat{\mathbf{r}}\mathbf{N}_0^s\hat{\mathbf{s}}_v. \tag{A.4}$$

Equations (A.3) and (A.4) are exactly the update expressions of the SUT-RAS method as indicated in Theorem 2 in TT (p. 872). To see the full equivalence simply redefine $\hat{\mathbf{r}}_v \equiv \hat{\mathrm{s}}_v^{-1}$ and take the transpose of (A.4) to express the Supply table in terms of the extended Make matrix. Q.E.D.

B MR-GRAS code in MATLAB programming language

The following code is also available for download from IOpedia Research Paper Series section at www.IOpedia.eu.

```
function [X,r,s,T] = mrgras(X0,u,v,G,Q,W,eps)
% PURPOSE: estimate a new multi-regional (or any partitioned) matrix X as
% close as possible to a given matrix X0 subject to the row sums, column
% sums and non-overlapping aggregation constraints, using MR-GRAS approach
% ------------------------------------------------------------------------
% USAGE:
% Write X = mrgras(X0,u,v,G,Q,W) OR [X,r,s,T] = mrgras(X0,u,v,G,Q,W) with
% or without eps as the seventh optional input argument, where
%
% INPUT:
% -> X0 = benchmark (original) matrix, not necessarily square
% -> u = column vector of row totals (new row sums)
% -> v = column vector of column totals (new column sums)
% -> W = non-overlapping aggregation constraints matrix
% -> G & Q = the row and column aggregator matrices such that G*X0*Q = W;
%            non-overlapping aggergation necessarily implies that the
%            column sums of G  and the row sums of Q must be all unity
% -> eps = convergence tolerance level; if empty, the default threshold
%          level is 0.1e-5 (=0.000001)
% -> In case of missing w_IJ, set the corresponding missing number to
%    w_IJ = 1010101 (assuming that 1010101 is not among the w_IJ's)
%
% OUTPUT (input-output analysis-related interpretation):
% -> X = updated/balanced/adjusted/projected matrix
% -> r = substitution effects (row multipliers)
% -> s = fabrication effects (column multipliers)
% -> T = technology or regional effects (aggregation multipliers)
% ------------------------------------------------------------------------
% REFERENCES:
% Temursho, U., Oosterhaven, J. and M.A. Cardenete (2019), A multi-regional
% generalized RAS updating technique, IOpedia Research Paper No. 2,
% September 2019, www.IOpedia.eu
% ------------------------------------------------------------------------
% NOTE FROM THE AUTHOR: Using this program and publishing the results in
% the form of a report, working/discussion paper, journal article, etc.
% requires citation of the above reference paper.
% ------------------------------------------------------------------------
% Written by:   Umed Temursho, May 2019
%               E-mail: utemursho@gmail.com

[m,n] = size(X0);
[h,k] = size(W);
N = zeros(m,n);
N(X0<0) = -X0(X0<0);
N = sparse(N);      % could save memory with large-scale matrices
P = X0+N;
P = sparse(P);
%
r0 = ones(m,1);       % initial guess for r in step 0
T0 = ones(h,k);       % initial guess for T in step 0
Te = G'*T0*Q';        % T expanded to fit the dimention of X0
p_rt = (P.*Te)'*r0;
n_rt = (N.*invM(Te))'*invd(r0)*ones(m,1);
s1 = invd(2*p_rt)*(v+sqrt(v.^2+4*p_rt.*n_rt));    % first step s
ss = -invd(v)*n_rt;
s1(p_rt==0) = ss(p_rt==0);
```

```
%
p_st = (P.*Te)*s1;
n_st = (N.*invM(Te))*invd(s1)*ones(n,1);
r1 = invd(2*p_st)*(u+sqrt(u.^2+4*p_st.*n_st));     % first step r
rr = -invd(u)*n_st;
r1(p_st==0) = rr(p_st==0);
%
P_rs = G*(diag(r1)*P*diag(s1))*Q;
N_rs = G*(invd(r1)*N*invd(s1))*Q;
T1 = invM(2*P_rs).*(W+sqrt(W.^2+4*P_rs.*N_rs));    % first step T
TT = -invM(W).*N_rs;
T1(P_rs==0) = TT(P_rs==0);
T1(W==1010101) = 1;    % set t_IJ=1 for missing aggregation total w_IJ
Te = G'*T1*Q';
%
p_rt = (P.*Te)'*r1;
n_rt = (N.*invM(Te))'*invd(r1)*ones(m,1);
s2 = invd(2*p_rt)*(v+sqrt(v.^2+4*p_rt.*n_rt));    % second step s
ss = -invd(v)*n_rt;
s2(p_rt==0) = ss(p_rt==0);
%
p_st = (P.*Te)*s2;
n_st = (N.*invM(Te))*invd(s2)*ones(n,1);
r2 = invd(2*p_st)*(u+sqrt(u.^2+4*p_st.*n_st));     % second step r
rr = -invd(u)*n_st;
r2(p_st==0) = rr(p_st==0);
%
P_rs = G*(diag(r2)*P*diag(s2))*Q;
N_rs = G*(invd(r2)*N*invd(s2))*Q;
T2 = invM(2*P_rs).*(W+sqrt(W.^2+4*P_rs.*N_rs));    % second step T
TT = -invM(W).*N_rs;
T2(P_rs==0) = TT(P_rs==0);
T2(W==1010101) = 1;    % set t_IJ=1 for missing w_IJ
%
tmax = max(max(abs(T2-T1)));
dif = [s2-s1;r2-r1;tmax];
iter = 1              % first iteration
if  nargin < 7 || isempty(eps)
    eps = 0.1e-5;     % default tolerance level
end
M = max(abs(dif));
while (M > eps)
    s1 = s2;
    r1 = r2;
    T1 = T2;
    Te = G'*T1*Q';
    %
    p_rt = (P.*Te)'*r1;
    n_rt = (N.*invM(Te))'*invd(r1)*ones(m,1);
    s2 = invd(2*p_rt)*(v+sqrt(v.^2+4*p_rt.*n_rt));      % next step s
    ss = -invd(v)*n_rt;
    s2(p_rt==0) = ss(p_rt==0);
    %
    p_st = (P.*Te)*s2;
    n_st = (N.*invM(Te))*invd(s2)*ones(n,1);
    r2 = invd(2*p_st)*(u+sqrt(u.^2+4*p_st.*n_st));      % next step r
    rr = -invd(u)*n_st;
    r2(p_st==0) = rr(p_st==0);
    %
    P_rs = G*(diag(r2)*P*diag(s2))*Q;
    N_rs = G*(invd(r2)*N*invd(s2))*Q;
    T2 = invM(2*P_rs).*(W+sqrt(W.^2+4*P_rs.*N_rs));     % next step T
    TT = -invM(W).*N_rs;
    T2(P_rs==0) = TT(P_rs==0);
```

```
    T2(W==1010101) = 1;                  % set t_IJ=1 for missing w_IJ
    %
    tmax = max(max(abs(T2-T1)));
    dif = [s2-s1;r2-r1;tmax];
    iter = iter+1
    M = max(abs(dif));
end
s = s2;      % final step s
r = r2;      % final step r
T = T2;      % final step T
Te = G'*T*Q';
%
X = Te.*(diag(r)*P*diag(s))-invM(Te).*(invd(r)*N*invd(s));  % updated matrix
end

function invd = invd(x)      % auxiliary function used above
invd = 1./x;
invd(x==0) = 1;
invd = diag(invd);
end

function invM = invM(X)      % auxiliary function used above
invM = 1./X;
invM(X==0) = 1;
end
%--------------------------------------------------------------------------
% END OF THE CODE
%--------------------------------------------------------------------------
```

IOpedia Research Paper Series

www.iopedia.eu

IOpedia Research Paper Series

is an online publication series available at: www.iopedia.eu

Temursho U., Oosterhaven J. and M.A. Cardenete (2019), A multiregional generalized RAS updating technique, *IOpedia Research Paper No. 02*, September 2019, wwww.IOpedia.eu.