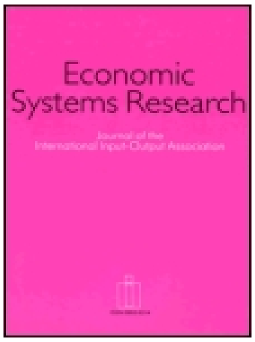
Economic Systems Research

ISSN: 0953-5314 (Print) 1469-5758 (Online) Journal homepage: www.tandfonline.com/journals/cesr20



The Solution of Updating or Regionalizing a Matrix with both Positive and Negative Entries

Theo Junius & Jan Oosterhaven

Economic Systems Research, Vol. 15, No. 1, 2003

Routledge
Taylor & Francis Group

The Solution of Updating or Regionalizing a Matrix with both Positive and Negative Entries

THEO JUNIUS & JAN OOSTERHAVEN

(Received February 2002; revised October 2002)

ABSTRACT *Normally, when updating or regionalizing input–output matrices with negative entries, the negative numbers are first brought outside the matrix, then the matrix is updated or regionalized, then the negative numbers are added back to the result. This is theoretically, and sometimes also empirically, a rather unsatisfactory procedure. This paper proposes a theoretically sound alternative for the presently used ad hoc procedure. Based on the first-order conditions of a restated information loss problem, we generalize the RAS-procedure using reciprocals of the exponential transformations of the related Lagrange multipliers. The diagonal matrices that update or regionalize a given matrix optimally are the solutions of a fixed-point problem. To derive a numerical solution, the paper presents the GRAS-algorithm, which is illustrated in terms of a simple updating example.*

KEYWORDS: *RAS-procedure; negative entries; fixed-point*

1. Introduction

The RAS-algorithm (developed for updating[1] input–output tables by Stone, 1961; Stone & Brown, 1962) iteratively adjusts an old matrix $\mathbf{A}$, with row sums $\mathbf{u}_0$ and column sums $\mathbf{u}_0$, to a 'new' matrix $\mathbf{X}$ that satisfies a 'new' set of given row sums $\mathbf{u}$ and column sums $\mathbf{v}$. With minimum loss of information the RAS-algorithm produces the new (target) matrix $\mathbf{X}$ with the required row and column sums such that:

$$\mathbf{X} = \hat{\mathbf{r}}\mathbf{A}\hat{\mathbf{s}}$$

in which $\hat{\mathbf{r}}$ and $\hat{\mathbf{s}}$ are diagonal matrices with positive entries on the main diagonal. The RAS-algorithm works, i.e. minimizes information loss, as long as the start matrix $\mathbf{A}$ contains only non-negative entries (see Bacharach, 1970; Lecomber, 1975; and Polenske, 1997, for overviews of RAS). Budavári (1981) discusses a generalization of RAS based on a general set of convex goal functions and on a

Theo Junius and Jan Oosterhaven, University of Groningen, Department of Economics, Postbus 800, 9700 AV Groningen, The Netherlands. E-mail: junius@eco.rug.nl and oosterhaven@eco.rug.nl. The authors thank Louis de Mesnard for his comments at the 48th North American Meetings of the RSAI in Charleston, USA, November, 2000. Furthermore comments by two anonymous referees are gratefully acknowledged.

ISSN 0953-5314 print; 1469-5758 online/03/010087-10 DOI: 10.1080/0953531032000056954

general set of linear constraints. This generalization, however, is still restricted to non-negative $\mathbf{X}$ and non-negative $\mathbf{A}$.

When negative entries are present two ad hoc approaches may be used. First, one may apply the RAS-algorithm to the matrix $\mathbf{A}$, inclusive of the negative entries. Such an approach, however, easily leads to a new matrix $\mathbf{X}$ with a structure that may strongly deviate from the structure of the old matrix $\mathbf{A}$. This is especially likely in the rows and columns of the relatively larger negative entries. In these rows and columns, the positive entries may have to be adjusted far more than the positive cells in the other rows and columns, as the negative entries have a negative contribution to the adjustment in each iterative round.

Second, in practice, therefore, the problem of dealing with negative entries is almost always 'solved' by treating the negative entries outside the RAS-procedure. In details this is done as follows.

(1) The matrix $\mathbf{A}$ is decomposed in two matrices, the matrix $\mathbf{P}$ with the non-negative entries of $\mathbf{A}$ and the matrix $\mathbf{N}$ with the absolute values of the negative entries of $\mathbf{A}$; thus $\mathbf{A} = \mathbf{P} - \mathbf{N}$.
(2) An adapted vector of row sums $\tilde{\mathbf{u}} = \mathbf{u} + \mathbf{N}\mathbf{i}$ and an adapted vector of column sums $\tilde{\mathbf{v}} = \mathbf{v} + \mathbf{i}\mathbf{N}$ is defined, where $\mathbf{i}$ is a summation vector with ones of appropriate length.
(3) The RAS-algorithm is then applied to the triple $\mathbf{P}$, $\tilde{\mathbf{u}}$, $\tilde{\mathbf{v}}$, which gives the matrix $\tilde{\mathbf{X}}$ as a result.
(4) The target matrix is obtained from $\mathbf{X} = \tilde{\mathbf{X}} - \mathbf{N}$.

The fact that the negative entries are *ignored* in this second procedure still is a disadvantage. The negative entries no longer have a negative contribution to the iterative adjustment procedure, but they do not have a positive contribution either. Hence, from a minimal information loss point of view, we have a suboptimal solution.

This paper develops a general mathematical device, such that one can apply an adaptation of the original RAS-algorithm to a start matrix $\mathbf{A}$, which consists of positive as well as negative entries, and positive as well as negative row and column totals. We call this adaption the GRAS-algorithm (Generalized RAS-algorithm). In the device we do not ignore the negative matrix-entries, but take account of the potential contribution of these entries to updating or regionalizing an input–output matrix. Basically, the device says that—when the target is the minimization of information loss—the objective function has to be written in terms of absolute values with respect to the negative matrix-entries. The generalization of the RAS-procedure can easily be combined with other generalizations, such as adding more constraints than just row and column totals (see, for example, Oosterhaven *et al.*, 1986; Snower, 1990).

This paper is organized as follows. In Section 2 we present the basic updating model. We show that under minimum information loss the negative entries have to be updated by means of the relevant reciprocals of the diagonal matrices $\hat{\mathbf{r}}$ and $\hat{\mathbf{s}}$. The model of Section 2 is investigated further in Section 3, in which the specific results have been shaped in terms of separate theorems, and we show that the original RAS-algorithm is a special case. In Section 4, we develop the GRAS-algorithm for numerically calculating the unknown 'adjustment' matrices $\hat{\mathbf{r}}$ and $\hat{\mathbf{s}}$ iteratively. In Section 5, a numerical example is used as an illustration of our approach. This paper ends with a number of conclusions in Section 6.

2. The Updating Problem

The updating, or regionalization, problem consists of three elements: the data, the problem to be solved and the programming model that specifies the problem mathematically.

2.1. The Data

The following data are given as follows.

- $\mathbf{A}$ is an $m \times n$ table (matrix) with an m-vector $\mathbf{u}_0 = \mathbf{Ai}$ of row sums and an n-vector $\mathbf{v}_0 = \mathbf{iA}$ of column sums.
- $\mathbf{A}$ consists of entries $a_{ij} \geqslant 0$ (non-negative flow variables) and/or entries $a_{ij} < 0$ (negative flow variables).
- $\mathbf{u} \neq \mathbf{u}_0$ is a given m-vector of 'new' row sums.
- $\mathbf{v} \neq \mathbf{v}_0$ is a given n-vector of 'new' column sums.
- $\mathbf{iu} = \mathbf{iv}$

Note that the row and column totals are not restricted to be non-negative.

2.2. The Problem

The problem is to find a 'new' $m \times n$ matrix $\mathbf{X}$: that *deviates least* from the given matrix $\mathbf{A}$ and satisfies $\mathbf{Xi} = \mathbf{u}$ and $\mathbf{iX} = \mathbf{v}$.

2.3. The Programming Model

The issue of *deviating least* is defined as the solution of the following *information loss* problem. Choose matrix entries x_{ij} such that:

$$x_{ij} = \arg\min \sum_i \sum_j |x_{ij}| \ln \frac{x_{ij}}{a_{ij}} \tag{1}$$

$$\text{s.t. } \forall i: \sum_j x_{ij} = u_i, \ \forall j: \sum_i x_{ij} = v_j \tag{2}$$

General discussions concerning the economic background of the information loss problem, applied to tables with only non-negative entries, can be found in, for example, Bacharach (1970), Miller & Blair (1985), Toh (1998), and Van der Linden & Dietzenbacher (2000). In the original RAS-problem, the requirements are added that zero entries keep their zero value and positive entries keep their positive signs. In our generalization, an analogous requirement holds for negative entries.[2] Mathematically, this is necessary since logarithms are only defined for positive arguments. The economic justification is that the economic information contained in the sign of each entry, whether it is negative, zero or positive, should be maintained whenever there is no additional information.

Observe that, from a mathematical point-of-view, this programming problem pertains to the minimization of the sum of a number of *strictly convex* functions restricted to a set of linear equality constraints. Hence, there exists only one unique solution to problem (1)–(2), provided that the equality constraints are mutually consistent.[3] This last provision implies that one of the $m \times n$ equality constraints, say the last one, is redundant.

3. The Solution to the Programming Model

To derive the solution of the programming model, it is convenient to rewrite the problem as follows. Define the variable z_{ij} as:

$$z_{ij} = \frac{x_{ij}}{a_{ij}} > 0 \text{ if } a_{ij} \neq 0 \text{ and } z_{ij} = 0 \text{ if } a_{ij} = 0$$

i.e.

$$x_{ij} = |a_{ij}|z_{ij}$$

Then problem (1)–(2) becomes: choose matrix entries $z_{ij} \geqslant 0$ such that

$$z_{ij} = \arg\min \sum_i \sum_j |a_{ij}| z_{ij} \ln z_{ij} \tag{3}$$

$$\text{s.t. } \forall i: \sum_j a_{ij} z_{ij} = u_i, \; \forall j: \sum_i a_{ij} z_{ij} = v_j \tag{4}$$

with the related Lagrange function:

$$L(Z, \lambda, \tau) = \sum_i \sum_j |a_{ij}| z_{ij} \ln z_{ij} + \sum_i \lambda_i \left[u_i - \sum_j a_{ij} z_{ij} \right] + \sum_j \tau_j \left[v_j - \sum_i a_{ij} z_{ij} \right]$$

where λ_i and τ_j are the Lagrange multipliers. This problem can be written as:

$$L(Z, \lambda, \tau) = \sum_{(i,j)\in P} a_{ij} z_{ij} \ln z_{ij} - \sum_{(i,j)\in N} a_{ij} z_{ij} \ln z_{ij} + \sum_i \lambda_i \left[u_i - \sum_j a_{ij} z_{ij} \right] + \sum_j \tau_j \left[v_j - \sum_i a_{ij} z_{ij} \right]$$

with P the set of pairs of indices (i,j) for which $a_{ij} \geqslant 0$ and N the set of pairs of indices (i,j) for which $a_{ij} < 0$. Then the following theorem can be derived.

Theorem 1
Let $Z = \{z_{ij}\}$, $\lambda = (\lambda_1, \ldots, \lambda_m)$, $\tau = (\tau_1, \ldots, \tau_n)$ denote the optimal solution. Then[4]

$$z_{ij} = r_i s_j e^{-1} \text{ if } a_{ij} \geqslant 0 \tag{5}$$

$$z_{ij} = r_i^{-1} s_j^{-1} e^{-1} \text{ if } a_{ij} < 0 \tag{6}$$

with:

$$r_i = e^{\lambda_i}, \quad s_j = e^{\tau_j}$$

Proof
Consider the optimality condition

$$\frac{\partial L}{\partial z_{ij}} = 0$$

For $a_{ij} \geqslant 0$ we have:

$$a_{ij} \ln z_{ij} + a_{ij} - \lambda_i a_{ij} - \tau_j a_{ij} = 0$$

or:

$$\ln z_{ij} = \lambda_i + \tau_j - 1 \Leftrightarrow z_{ij} = e^{\lambda_i} e^{\tau_j} e^{-1}$$

For $a_{ij} < 0$ we have:

$$-a_{ij} \ln z_{ij} - a_{ij} - \lambda_i a_{ij} - \tau_j a_{ij} = 0$$

or:

$$\ln z_{ij} = -\lambda_i - \tau_j - 1 \Leftrightarrow z_{ij} = \mathrm{e}^{-\lambda_i} \mathrm{e}^{-\tau_j} \mathrm{e}^{-1}$$

Equations (5) and (6) imply the following result.

Theorem 2
For the target matrix $\mathbf{X}$ we obtain:

$$\text{If } a_{ij} \geqslant 0 \text{ then } \mathbf{x}_{ij} = a_{ij} z_{ij} = r_i a_{ij} s_j / \mathrm{e} \geqslant 0 \tag{7}$$

$$\text{If } a_{ij} < 0 \text{ then } \mathbf{x}_{ij} = a_{ij} z_{ij} = r_i^{-1} a_{ij} s_j^{-1} / \mathrm{e} < 0 \tag{8}$$

With $\hat{\mathbf{r}}$ and $\hat{\mathbf{s}}$ are the diagonal matrices containing r_i, respectively s_j, solutions (7) and (8), can be expressed in matrix terms.

Theorem 3
The pair of diagonal matrices $(\hat{\mathbf{r}}, \hat{\mathbf{s}})$ is a solution of the system of non-linear equations:

$$(\hat{\mathbf{r}}\mathbf{P}\hat{\mathbf{s}} - \hat{\mathbf{r}}^{-1}\mathbf{N}\hat{\mathbf{s}}^{-1})\mathbf{i} = \mathbf{u}^\star \tag{9}$$

$$\mathbf{i}(\hat{\mathbf{r}}\mathbf{P}\hat{\mathbf{s}} - \hat{\mathbf{r}}^{-1}\mathbf{N}\hat{\mathbf{s}}^{-1}) = \mathbf{v}^\star \tag{10}$$

with:

$$p_{ij} = a_{ij} \text{ for } a_{ij} \geqslant 0, \quad p_{ij} = 0 \text{ elsewhere}$$

$$n_{ij} = -a_{ij} \text{ for } a_{ij} < 0, \quad n_{ij} = 0 \text{ elsewhere}$$

and the vectors $\mathbf{u}^\star$ and $\mathbf{v}^\star$ defined as:

$$\mathbf{u}^\star = \mathrm{e}\mathbf{u}, \quad \mathbf{v}^\star = \mathrm{e}\mathbf{v}$$

Furthermore, we have three important, additional, results.

Theorem 4
System (9)–(10) is *hyperbolic homogeneous* in terms of the array $(\hat{\mathbf{r}}, \hat{\mathbf{s}})$. With the property of hyperbolic homogeneity, we state that if the array $(\hat{\mathbf{r}}, \hat{\mathbf{s}})$ is a solution to (9)–(10), then for any scalar $k > 0$ the array $(k\hat{\mathbf{r}}, \hat{\mathbf{s}}/k)$ is also a solution to (9)–(10).

From the facts that (i) one of the equality constraints is redundant and (ii) the solution of system (9)–(10) has the property of hyperbolic homogeneity, we conclude that the following *normalization rule* holds.

Theorem 5
We may choose the numeraire

$$s_n = \mathrm{e}^{\tau_n} = 1 \Leftrightarrow \tau_n = 0 \tag{11}$$

That Stone's RAS-procedure is a special case of our more general approach is formalized in the next theorem.

Theorem 6

If we have $\forall(i,j)$: $n_{ij} = 0$ (i.e., no negative matrix entries enter), then one obtains the original RAS-procedure for updating a semi-positive matrix.

Proof

Inserting $n_{ij} = 0$ into equation (2) yields the minimum information loss problem pertaining to the Stone-approach (e.g. Bacharach, 1970).

4. The GRAS-Algorithm

The above problem, (9)–(10), can of course be solved by any non-linear equation solver, such as GAMS. It is, however, instructive and handy to derive an iterative solution that is comparable to the original RAS-algorithm. For that purpose, we present the GRAS-algorithm for updating a given matrix **A** with negative matrix entries.[5] It is a sequential algorithm of the form:

Step 1. Start from a given initial matrix $\hat{\mathbf{r}}(0)$.
Step 2. Use system (9) to calculate the matrix $\hat{\mathbf{s}}(1)$.
Step 3. 'Jump' to equation (10) to calculate the matrix $\hat{\mathbf{r}}(1)$ using the matrix $\hat{\mathbf{s}}(1)$ obtained in Step 2.
Step 4. The algorithm is described by:

$$\hat{\mathbf{r}}(0) \rightarrow \hat{\mathbf{s}}(1) \rightarrow \hat{\mathbf{r}}(1) \rightarrow \hat{\mathbf{s}}(2) \rightarrow \hat{\mathbf{r}}(2) \rightarrow \hat{\mathbf{s}}(3) \rightarrow \text{etc}$$

Step 5. It reaches its solution $(\hat{\mathbf{r}}, \hat{\mathbf{s}})$ if, for an arbitrary small $\varepsilon > 0$, we have:

$$\| (\hat{\mathbf{r}}\mathbf{P}\hat{\mathbf{s}})\mathbf{i} - (\hat{\mathbf{r}}^{-1}\mathbf{N}\hat{\mathbf{s}}^{-1})\mathbf{i} - \mathbf{u}^\star \| < \varepsilon \| \mathbf{u}^\star \| \tag{12}$$

$$\| \mathbf{i}(\hat{\mathbf{r}}\mathbf{P}\hat{\mathbf{s}}) - \mathbf{i}(\hat{\mathbf{r}}^{-1}\mathbf{N}\hat{\mathbf{s}}^{-1}) - \mathbf{v}^\star \| < \varepsilon \| \mathbf{v}^\star \| \tag{13}$$

After a number of iterations we have, see equations (9)–(10):

$$[\hat{\mathbf{r}}\mathbf{P}\hat{\mathbf{s}}_1]\mathbf{i} - [\hat{\mathbf{r}}^{-1}\mathbf{N}\hat{\mathbf{s}}_1^{-1}]\mathbf{i} = \mathbf{u}^\star \tag{14}$$

in which the matrix $\hat{\mathbf{s}}_1$ is equal to the matrix $\hat{\mathbf{s}}_i$ derived in the last iteration before the one considered here. After pre-multiplication with the matrix $\hat{\mathbf{r}}$, equation (14) results in:

$$[\hat{\mathbf{r}}^2\mathbf{P}\hat{\mathbf{s}}_1]\mathbf{i} - [\mathbf{N}\hat{\mathbf{s}}_1^{-1}]\mathbf{i} - \hat{\mathbf{r}}\mathbf{u}^\star = \mathbf{0} \tag{15}$$

The first row of this equation can be written as a *second-order equation* in the unknown r_1:

$$p_1(s)r_1^2 - u_1^\star r_1 - n_1(s) = 0 \tag{16}$$

$$\text{with } p_1(s) = \sum_j p_{1j}s_j \tag{17}$$

$$\text{and } n_1(s) = \sum_j (n_{1j}/s_j) \tag{18}$$

The positive root of equation (16) follows from the basic result of solving a second-order equation, i.e.

$$r_1 = \frac{u_1^\star + \sqrt{[(u_1^\star)^2 + 4p_1(s)n_1(s)]}}{2p_1(s)} \tag{19}$$

Observe that equation (16) is a second-order equation which, by definition, has two real roots, a negative and a positive one. In terms of absolute values, the positive root dominates the negative root. For our analysis, the negative root is, however, irrelevant because we must have $r_1 = \exp\{\lambda_1\} > 0$. In an analogous way, it is possible to obtain a mathematical specification of the other elements of the main diagonal of the matrix $\hat{\mathbf{r}}$. In general, we have the following result.

$$r_i(s) = \rho_i(s_1, \ldots, s_n) \tag{20}$$

$$s_j(r) = \sigma_j(r_1, \ldots, r_m) \tag{21}$$

where the functions ρ_i and σ_j are defined by:

$$\rho_i(s) = \frac{u_i^\star + [(u_i^\star)^2 + 4p_i(s)n_i(s)]^{1/2}}{2p_i(s)} \tag{22}$$

$$\sigma_j(r) = \frac{v_j^\star + [(v_j^\star)^2 + 4p_j^\star(r)n_j^\star(r)]^{1/2}}{2p_j^\star(r)} \tag{23}$$

with the functions $p_i(s)$, $n_i(s)$, $p_j^\star(r)$, $n_j^\star(s)$ defined in a similar way as in equations (17) and (18).

From a mathematical point-of-view system (22)–(23) pertains to the search for a fixed point $(\hat{\mathbf{r}}, \hat{\mathbf{s}})$ since in equilibrium we may write[6]

$$\hat{\mathbf{r}} = \hat{\sigma}(\hat{\mathbf{s}}) \tag{24}$$

$$\hat{\mathbf{s}} = \hat{\rho}(\hat{\mathbf{r}}) \tag{25}$$

Based on equations (20)–(23) we propose the following GRAS-algorithm, a *generalized RAS*-algorithm. Start from an arbitrary initial diagonal $\hat{\mathbf{r}}(0) > 0$ and calculate sequentially:

$$\hat{\mathbf{r}}(0) \to \hat{\mathbf{s}}(1) = \hat{\sigma}(\hat{\mathbf{r}}(0)) \to \hat{\mathbf{r}}(1) = \hat{\rho}(\hat{\mathbf{s}}(1)) \text{ etc}$$

in order to converge to the optimal values $\hat{\mathbf{r}}, \hat{\mathbf{s}}$ by using iteratively system (24)–(25).

We suggest that the initial vector $\mathbf{r}(0)$ is chosen such that:

$$\mathbf{r}(0) = \mathbf{i}$$

In this way, the first adjustment $\hat{\mathbf{s}}(1)$ applies to the columns of the input–output table. The reason for this choice is that, in this case, the equilibrium row multipliers $\hat{\mathbf{r}}$ may be interpreted economically as 'substitution effects' and the column multipliers $\hat{\mathbf{s}}$ as 'fabrication effects' (see Stone, 1961; Toh, 1998; and Van der Linden & Dietzenbacher, 2000).

5. A Numerical Comparison of GRAS with RAS

As an illustration, the following archetypal updating problem is considered. Table 1 is an initial input–output table in, say, billions of euros. The problem is to calculate the missing cells of the new input–output table (Table 2) such that the loss of information is minimal.

Table 1. Initial, old input–output table

	Goods	Services	Consumption	Net exports	Total output
Goods	7	3	5	−3	12
Services	2	9	8	1	20
Net taxes	−2	0	2	1	1
Total use	7	12	15	−1	33
Value added	5	8	0	0	13
Total input	12	20	15	−1	

Table 2. Row and column totals of the new, to be updated, input–output table

	Goods	Services	Consumption	Net exports	Total output
Goods					15
Services					26
Net taxes					−1
Total use	9	16	17	−2	40
Value added	6	10	0	0	16
Total input	15	26	17	−2	

Note that, besides the negative entries in the initial 3×4 upper-left sub-matrix, both the initial, and the new row and column sums have negative as well as positive entries, one of which changes sign. The numerical example thus contains all theoretical complications included in system (1)–(2) above. In fact, it simulates a growing economy, with growth rates for sectoral output equalling 25% and 30%, for sectoral total use equalling 28.6% and 33.3%, and for total consumption and total value added equalling 13.3% and 23%, respectively. The latter two increases are comparatively low, because the trade deficit is assumed to double, while net taxes are assumed to have turned in equally large net subsidies.

When the maximum allowable column-sum and row-sum error is set at 0.001%, applying our GRAS-algorithm (20)–(23) to this problem, gives the solution in Table 3.

With the same maximum allowable error, applying the standard RAS-algorithm

Table 3. New table derived with the GRAS algorithm

	Goods	Services	Consumption	Net exports	Total output
Goods	7.84	3.58	5.82	−2.24	15
Services	2.59	12.42	10.78	0.21	26
Net taxes	−1.43	0	0.40	0.03	−1
Total use	9	16	17	−2	40

to the matrix without the negatives and putting them back again, as explained in steps (1)–(4) of the introduction, gives the solution in Table 4.

Table 4. New table derived with the RAS algorithm, with exogenous negatives

	Goods	Services	Consumption	Net exports	Total output
Goods	8.38	3.73	5.89	−3	15
Services	2.62	12.27	10.34	0.77	26
Net taxes	−2	0	0.77	0.23	−1
Total use	9	16	17	−2	40

The two negative entries in Table 4 are equal to those of Table 1. Consequently, in order to satisfy the new row and column totals, the positive RAS entries in Table 4 deviate more from those in Table 1 than the corresponding GRAS entries in Table 3. This indicates that the negative entries in the adapted RAS-procedure do not contribute to the minimization of information loss.

This is also obvious in Table 5, which compares the performance characteristics of both approaches. The adapted RAS-procedure converges slightly slower than our GRAS-algorithm, but reaches the set limit for the maximum allowable column or row error in the same number of 12 iterations. At this same level of accuracy, however, our GRAS-algorithm shows a smaller loss of information (8.08) than the loss of 9.17 in the case of the adapted RAS-procedure.[7]

Table 5. Comparison of convergence speed and information loss of GRAS versus Adapted RAS

	GRAS	Adapted RAS
Iterations maximum row/column error <10%	2	3
Iterations maximum row/column error <1%	4	5
Iterations maximum row/column error <0.1%	8	8
Iterations maximum row/column error <0.001%	12	12
Information loss (maximum row/column error <0.001%)	8.08	9.17

6. Conclusions

In this paper, we generalize the RAS-procedure, yielding the GRAS-procedure, such that it is possible to update or regionalize a given matrix, which can be decomposed into a semi-positive matrix and a negative matrix. Our generalization has been found by:

- studying a generalization of the target function of Theil's minimum information loss problem with the negative entries in absolute terms;
- considering the *abc*-theorem pertaining to the positive root of the particular second-order equation $az^2 - bz - c = 0$, $a, b > 0$, yielding:

$$z = \frac{b + \sqrt{[b^2 + 4ac]}}{2a}$$

The existence, uniqueness and stability of the solution of the GRAS-algorithm are subjects that are beyond the scope of this paper. The fact that our analysis concerns the minimization of a sum of strictly convex functions restricted by a system of linear equality constraints, gives, in our view, sufficient mathematical structure to guarantee these three properties. As argued above, it basically concerns a non-linear fixed-point problem in terms of the unknown entries of diagonal matrices $\hat{\mathbf{r}}$ and $\hat{\mathbf{s}}$. In terms of a simple input–output table we showed that our GRAS-algorithm for updating and regionalizing input–output matrices works and produces a lower information loss than the standard ad hoc approach.

Notes

1. See McMenamin & Haring (1974) for a first application of regionalizing national input–output matrices, and Oosterhaven *et al.* (1986) for updating interregional input–output matrices.
2. This specification generalizes Theil's target function of a minimum information loss problem (Theil, 1967).
3. For the mathematical background of this type of non-linear programming problems see, for example, Takayama (1994).
4. The constant e is the exponential number $e \approx 2.718$.
5. The algorithm is tentative because we pay no attention to existence, uniqueness and stability of a solution.
6. See, for example, Takayama (1994) for a discussion regarding fixed-points.
7. When more simple examples with relatively larger negative entries are explored, larger differences in information loss occur between GRAS and RAS. With, for example, the matrix $\mathbf{A} = [2, -1, 3; 1, 2, -1]$, $\mathbf{u} = (6, 4)$ and $\mathbf{v} = (5, 2, 3)$, GRAS produces an information loss of 3.38 and the adapted RAS-procedure produces a loss of 4.07.

References

Bacharach, M. (1970) *Biproportional Matrices and Input–Output Change* (Cambridge, Cambridge University Press).

Budavári, P. (1981) Generalization of the 'RAS' method: linear restrictions with strictly convex distance function (duality theory and algorithm), *Proceedings of the Third Hungarian Conference on Input–Output Techniques*, Héviz.

Lecomber, J. R. C. (1975) A critique of methods of adjusting, updating and projecting matrices, in: R. I. G. Allen & W. F. Gossling (eds), *Estimating and Projecting Input–Output Coefficients* (London, Input–Output Publishing Company).

McMenamin, D. G. & Haring, J. E. (1974) An appraisal of nonsurvey techniques for estimating regional input–output models, *Journal of Regional Science*, 14, pp. 191–205.

Miller, R. E. & Blair, P. D. (1985) *Input–Output Analysis: Foundations and Extensions* (Englewood Cliffs, NJ, Prentice Hall).

Oosterhaven, J., Piek, G. & Stelder, D. (1986) Theory and practice of updating regional versus interregional interindustry tables, *Papers of the Regional Science Association*, 59, pp. 52–72.

Polenske, K. R. (1997) Current uses of the RAS technique: a critical review, in: A. Simonovits & A. E. Steenge (eds), *Prices, Growth and Cycles* (London, Macmillan).

Snower, D. J. (1990) New methods of updating input–output matrices, *Economic Systems Research*, 2, pp. 27–37.

Stone, R. (1961) *Input–Output and National Accounts* (Paris, Organization for European Economic Cooperation).

Stone, R. & Brown, A. (1962) *A Computable Model of Economic Growth* (London, Chapman and Hall).

Takayama, A. (1994) *Analytical Methods in Economics* (New York, Harvester Wheatsheaf).

Theil, H. (1967) *Economics and Information Theory* (Amsterdam, North-Holland).

Toh, M. H. (1998) The RAS approach in updating input–output matrices: an instrumental variable interpretation and analysis of structural changes, *Economic Systems Research*, 10, pp. 63–78.

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