ELSEVIER Computational Statistics & Data Analysis 28 (1998) 307–324

COMPUTATIONAL
STATISTICS
& DATA ANALYSIS

Three-way SIMPLIMAX for oblique rotation of the three-mode factor analysis core to simple structure

Henk A.L. Kiers*

Department of Psychology, University of Groningen, Grote Kruisstraat 2/1,
9712 TS Groningen, Netherlands

Received 1 July 1996; received in revised form 1 December 1997; accepted 2 April 1998

Abstract

SIMPLIMAX is proposed as a procedure for oblique rotation of a factor loadings matrix to simple structure. The distinguishing feature of this method is that it rotates the loading matrix so that after rotation the m smallest elements have a minimal sum of squares (where m is specified in advance). In the present paper, the SIMPLIMAX method is generalized to handle three-way arrays: Three-way SIMPLIMAX finds oblique simple structure rotations of the core matrix that results from a three-mode factor analysis. Specifically, three-way SIMPLIMAX minimizes the sum of the m smallest elements of the rotated core array. An algorithm for three-way SIMPLIMAX is presented, the performance of the algorithm is discussed, some applications are shown, and it is indicated how the method can be used to serve additional purposes. Also, it is shown how the method can be used for rotation of solutions of N-mode factor analysis, and for rotation over a subset of the modes of an N-mode core array. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Three-way data; Components analysis; Simple structure

1. Introduction

In three-mode factor analysis (Tucker, 1966; Kroonenberg and De Leeuw, 1980; Kroonenberg, 1994) a three-way data set is modelled by means of component matrices for each of the three modes, and by a so called three-way core array, relating the components of the three modes to each other. Mathematically, the three-mode factor

* Tel.: +31(0) 50 3636339; fax: +31(0) 50 3636304; e-mail: h.a.l.kiers@ppsw.rug.nl

PII S0167-9473(98)00041-3

analysis (3MFA) model can be described as follows. Let $\boldsymbol{X}$ denote an $I \times J \times K$ data array. Then, according to the 3MFA model, we have

$$\hat{x}_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} a_{ip} b_{jq} c_{kr} g_{pqr}, \tag{1}$$

where $\hat{x}_{ijk}$ denotes the estimate for the element (i,j,k) of $\boldsymbol{X}$; $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ (with elements a_{ip}, b_{jq} and c_{kr} respectively) are component matrices of orders $I \times P, J \times Q$, and $K \times R$, respectively, and $\boldsymbol{G}$ is a $P \times Q \times R$ three-way array denoted as the *core*, with elements g_{pqr}, $i = 1,\dots,I$, $j = 1,\dots,J$, $k = 1,\dots,K$, $p = 1,\dots,P$, $q = 1,\dots,Q$, and $r = 1,\dots,R$. The elements of $\boldsymbol{A}, \boldsymbol{B}$, and $\boldsymbol{C}$ can be considered component scores for "A-mode entries" (in $\boldsymbol{A}$), "B-mode entries" (in $\boldsymbol{B}$), and "C-mode entries" (in $\boldsymbol{C}$), respectively. The elements of the core indicate how the components from the different modes interact.

In 3MFA the model is (usually) fitted to the data by minimizing the sum of squared residuals, $\sum_i \sum_j \sum_k (x_{ijk} - \hat{x}_{ijk})^2$, over $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{G}$. Kroonenberg and De Leeuw (1980) have proposed an alternating least squares (ALS) algorithm for this minimization. The method does not have a unique solution. As in factor analysis of two-way data, the solution is unique up to a rotation of the component matrices: Tucker (1966) already described that postmultiplication of matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ by nonsingular matrices can always be compensated by applying the inverse of these matrices to the core array. Specifically, it can be verified that$\tilde{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{S}^{-1}$, $\tilde{\boldsymbol{B}} = \boldsymbol{B}\boldsymbol{T}^{-1}$, $\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{U}^{-1}$ and $\tilde{\boldsymbol{G}}$ with elements

$$\tilde{g}_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} s_{ip} t_{jq} u_{kr} g_{pqr}, \tag{2}$$

$i = 1,\dots,P$, $j = 1,\dots,Q$, $k = 1,\dots,R$, give the same estimates for $\hat{\boldsymbol{X}}$ as $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{G}$ do. Therefore, the solution obtained by the Kroonenberg and De Leeuw algorithm is just one out of infinitely many equally valid solutions. We may always transform this by nonsingular matrices $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$, as described above, without affecting the model fit.

The results from a 3MFA are rather difficult to interpret. This is because the interpretation does not only involve interpretation of the components for all three modes, but also of all interactions between these components (as given by the core). Each nonnegligible core element points to an important contribution to the data of a particular combination of components from the three modes. Hence, when all core elements are nonnegligible, interpretation consists of a burdensome enumeration of all those combinations, and hardly serves to give insight in the main information in the data. Therefore, it is desirable to have a "simple" core, that is, a core with only few large elements, and most elements close to zero. The present paper deals with a method that exploits the above mentioned transformational freedom of the 3MFA solution to obtain simple cores. Specifically, rotation matrices (which here denotes both orthogonal and oblique transformations) $\boldsymbol{S}$, $\boldsymbol{T}$ and/or $\boldsymbol{U}$ are found such that the core becomes simple. Because these rotations must be compensated by applying the (inverse) rotations to $\boldsymbol{A}$, $\boldsymbol{B}$ and/or $\boldsymbol{C}$, it is sometimes preferred to use only a subset of the matrices $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ to find a simple core: For instance, if matrix $\boldsymbol{A}$ is very easy to interpret, it may be preferable not to distort this aspect of the 3MFA solution, and one could attempt to simplify the core sufficiently by rotations involving $\boldsymbol{T}$ and $\boldsymbol{U}$ only. The method to be presented in the present paper allows for rotations by either the full set of rotation matrices or any subset thereof.

The idea to use the rotational indeterminacy to obtain a simpler core is not new. Several suggestions have been made to rotate the core to a simpler form, for instance, to a form in which the frontal core planes are as diagonal as possible (e.g., Cohen, 1974, 1975; MacCallum, 1976; Kroonenberg, 1983, Chapter 5), or in which the array is as close as possible to superdiagonality (Kiers, 1992). Rather than rotating the core to an a priori specified form, one may choose to transform the core to an unspecified simple structure, as was done, for instance, by Murakami (1983) who rotated the transposed supermatrix of frontal core planes to simple structure by means of varimax rotation. More recently, Kruskal (1988) proposed "tri-quartimax" rotation, which is a procedure for maximizing a combination of normalized quartimax functions applied to the supermatrices consisting of the frontal, lateral and horizontal planes, respectively, of the core. Recently, Kiers (1997) proposed a procedure which optimizes the "orthomax" criterion (see Crawford and Ferguson, 1970; Jennrich, 1970) applied to a similar set of supermatrices of core elements. Kruskal's and Kiers' procedures are the most general procedures available for oblique and orthogonal rotation, respectively, to an unspecified simple form. An obvious limitation of Kiers' method is that it does not allow for oblique rotations. Kruskal's method does allow for oblique rotations, but, unfortunately, no published details are available on Kruskal's procedure and its performance.

In the present paper, we propose a new procedure for oblique simple structure rotation of the core. This procedure is based on the SIMPLIMAX method (Kiers, 1994) for oblique simple structure rotation of a (two-way) loading matrix. Specifically, (two-way) SIMPLIMAX is a method for oblique rotation of a loading matrix such that in the resulting loading matrix the m (a number to be specified by the user) smallest loadings have a minimal sum of squares. It is not *a priori* known *which* loadings will be the smallest ones, hence the method does not only rotate such that m particular loadings have a minimal sum of squares, but simultaneously determines for *which* m loadings the smallest sum of squares can be found. Due to this search for the location of the smallest loadings, the method is very sensitive to finding suboptimal solutions, which, however is dealt with by using a large number of randomly started runs of the algorithm. Furthermore, by using different values for m, different SIMPLIMAX solutions can be compared, and the rotated loading matrix with the largest number of loadings that are sufficiently close to zero is taken as the solution with the best simple structure. It should be noted that here a compromise is to be struck between the choice for m, and the choice for what is sufficiently small. In the present paper, two-way SIMPLIMAX is generalized to rotate a three-way core such that the sum of m smallest squared elements is minimized. In this generalization, the core can be rotated in all three directions, hence the three-way

SIMPLIMAX method finds the three oblique rotations that jointly lead to the core with the minimal sum of smallest squared core elements.

After presenting an algorithm for three-way SIMPLIMAX, we will discuss the performance of the algorithm. Next, the method will be illustrated on artificially and empirically obtained core arrays. The method will be generalized a bit further by describing how to handle rotation of N-mode factor analysis solutions, and how to handle rotation of subsets of the modes of the N-mode core array. Finally, it will be discussed how the method can be put to use for other purposes, especially in the context of constrained three-mode factor analysis (Kiers, 1992; Kiers, et al. 1997).

2. Three-way SIMPLIMAX

The SIMPLIMAX method (Kiers, 1994) minimizes the sum of squares of the smallest m rotated loadings. This is achieved, somewhat indirectly, as follows: SIMPLIMAX finds an oblique rotation matrix $\boldsymbol{T}_s$ and a target matrix $\boldsymbol{H}$ with m zero elements, such that $\Lambda\boldsymbol{T}_s$ (where Λ denotes a loading matrix) and $\boldsymbol{H}$ are optimally similar. Specifically, SIMPLIMAX minimizes

$$\sigma(\boldsymbol{T}_s,\boldsymbol{H}) = \|\Lambda\boldsymbol{T}_s - \boldsymbol{H}\|^2, \tag{3}$$

over $\boldsymbol{T}_s$ subject to $\mathrm{Diag}(\boldsymbol{T}_s^{-1}\boldsymbol{T}_s^{-1'}) = \boldsymbol{I}$, *and* over all matrices $\boldsymbol{H}$ that have m zero elements. The matrix $\boldsymbol{H}$ that minimizes σ is always such that the *zero* elements correspond to the m smallest elements of $\Lambda\boldsymbol{T}_s$, thus, in this way SIMPLIMAX determines *which* are the m loadings for which the sum of squares is to be minimal. Furthermore, the *nonzero* elements of $\boldsymbol{H}$ *equal* the corresponding elements of $\Lambda\boldsymbol{T}_s$, and hence all contribute 0 to the function σ. As a consequence, the function $\sigma(\boldsymbol{T}_s,\boldsymbol{H})$ reduces to the sum of squares of the m smallest rotated loadings, as desired. Therefore, minimizing $\sigma(\boldsymbol{T}_s,\boldsymbol{H})$ is indeed equivalent to minimizing the sum of squares of the m smallest loadings.

For minimizing the function $\sigma(\boldsymbol{T}_s,\boldsymbol{H})$, an iterative algorithm proposed by Kiers (1994) is used, which consists of alternating (until convergence) between, on the one hand, updating $\boldsymbol{T}_s$ and the nonzero elements of $\boldsymbol{H}$, while keeping the *zero values* in $\boldsymbol{H}$ fixed (using Browne's (1972) algorithm for oblique rotation to a partly specified target), and, on the other hand, updating all elements of $\boldsymbol{H}$, while keeping $\boldsymbol{T}_s$ fixed (which simply amounts to setting $\boldsymbol{H} = \Lambda\boldsymbol{T}_s$ and next setting the smallest m elements of $\boldsymbol{H}$ to zero). The latter step serves to readjust the target to the location of the m smallest loadings. In both steps $\sigma(\boldsymbol{T}_s,\boldsymbol{H})$ decreases, or at least does not increase, hence, because the function is bounded below by zero, the function value will eventually stabilize, although not necessarily to the global minimum. In fact, the algorithm is rather prone to hitting suboptimal solutions. Therefore, several restarts must be used to increase the chance of attaining the global minimum.

The simplicity of the SIMPLIMAX solution can be described in terms of the number (m) of small elements, and in terms of how small these small values actually are (expressed by σ, which gives the sum of squares of the smallest m elements). Because there is no rationale for choosing m in advance, it is advised to try a number of different values for m, and choose that solution that gives the best compromise between parsimony in terms of the number of allegedly small elements (m) and σ, which indicates how small the smallest m elements actually are.

To adapt the SIMPLIMAX procedure to rotation of the *three-way core array* of a 3MFA solution, we have to note that in 3MFA the rotational freedom pertains to three directions (rather than the single one in the two-way case). Specifically, the transformed core array ($\tilde{\boldsymbol{G}}$) involves matrices $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$, see Eq. (2). Applying the SIMPLIMAX criterion to the core, we seek to minimize

$$\sigma(\boldsymbol{S},\boldsymbol{T},\boldsymbol{U},\boldsymbol{H}) = \sum_{i=1}^{P}\sum_{j=1}^{Q}\sum_{k=1}^{R}(\tilde{g}_{ijk} - h_{ijk})^2, \tag{4}$$

where $\boldsymbol{H}$ is a $P \times Q \times R$ target array with m zero elements (at unspecified positions). In two-way SIMPLIMAX, the oblique rotation matrix ($\boldsymbol{T}_{\mathrm{s}}$) is constrained such that $\mathrm{Diag}(\boldsymbol{T}_{\mathrm{s}}^{-1}\boldsymbol{T}_{\mathrm{s}}^{-1'}) = \boldsymbol{I}$. In this way the component scores which, before rotation, are uncorrelated and standardized, will, after oblique rotation, still be standardized (although no longer uncorrelated). In the present situation we have three component matrices, $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$. In a 3MFA these matrices are usually columnwise orthonormal (and otherwise can be taken columnwise orthonormal without loss of fit). To ensure that the obliquely rotated versions of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, matrices$\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{C}}$ have unit sums of squares columnwise (like $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$), it is proposed here to constrain $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ such that Diag $((\boldsymbol{S}')^{-1}\boldsymbol{S}^{-1}) = \boldsymbol{I}$, Diag $((\boldsymbol{T}')^{-1}\boldsymbol{T}^{-1}) = \boldsymbol{I}$ and Diag $((\boldsymbol{U}')^{-1}\boldsymbol{U}^{-1}) = \boldsymbol{I}$. To verify this normalization we may check that Diag $(\tilde{\boldsymbol{A}}'\tilde{\boldsymbol{A}}) =$ Diag $((\boldsymbol{S}')^{-1}\boldsymbol{A}'\boldsymbol{A}\boldsymbol{S}^{-1}) =$ Diag $((\boldsymbol{S}')^{-1}\boldsymbol{S}^{-1}) = \boldsymbol{I}$, etc. Thus, three-way SIMPLIMAX is defined as the method that minimizes $\sigma(\boldsymbol{S},\boldsymbol{T},\boldsymbol{U},\boldsymbol{H})$ over $\boldsymbol{S},\boldsymbol{T},\boldsymbol{U}$ and a target $\boldsymbol{H}$ with m zeros, subject to the above chosen constraints on $\boldsymbol{S},\boldsymbol{T}$ and $\boldsymbol{U}$.

To minimize (4) subject to the constraints, we alternately update $\boldsymbol{S}$ and $\boldsymbol{H}$, $\boldsymbol{T}$ and $\boldsymbol{H}$, and $\boldsymbol{U}$ and $\boldsymbol{H}$, by using the two-way SIMPLIMAX method, as described below. The procedure is started with randomly or rationally chosen initial matrices $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ that satisfy the constraints, and continues until the function value stabilizes. To update $\boldsymbol{S}$ and $\boldsymbol{H}$, given the other matrices ($\boldsymbol{T}$ and $\boldsymbol{U}$), we have to minimize (or at least decrease) the function

$$\begin{aligned}\tilde{\sigma}(\boldsymbol{S},\boldsymbol{H}|\mathbb{T},\mathbb{U}) &= \sum_{i=1}^{P}\sum_{j=1}^{Q}\sum_{k=1}^{R}(\tilde{g}_{ijk} - h_{ijk})^2\\ &= \|\boldsymbol{S}\boldsymbol{G}_{\mathrm{F}}(\mathbb{U}'\otimes\mathbb{T}') - \boldsymbol{H}_{\mathrm{F}}\|^2,\end{aligned} \tag{5}$$

where $\boldsymbol{G}_{\mathrm{F}}$ and $\boldsymbol{H}_{\mathrm{F}}$ denote the supermatrices with frontal planes of the respective three-way arrays next to each other, $\mathbb{T}$ and $\mathbb{U}$ denote the current (fixed) versions of $\boldsymbol{T}$ and $\boldsymbol{U}$, and $\otimes$ denotes the Kronecker product. It is worth noting that $\boldsymbol{G}_{\mathrm{F}}(\mathbb{U}'\otimes\mathbb{T}')$ contains the elements of the core after rotation by $\mathbb{T}$ and $\mathbb{U}$ in the appropriate directions. By transposing both sides in Eq. (5), we find

$$\tilde{\sigma}(\boldsymbol{S},\boldsymbol{H}|\mathbb{T},\mathbb{U}) = \|(\mathbb{U}\otimes\mathbb{T})\boldsymbol{G}_{\mathrm{F}}'\boldsymbol{S} - \boldsymbol{H}_{\mathrm{F}}'\|^2, \tag{6}$$

which can be recognized as the SIMPLIMAX function (3) with Λ replaced by $(\mathbb{U} \otimes \mathbb{T})G'_F$, T_s replaced by S', and, in line with this replacement, the constraint Diag $(T_s^{-1}T_s^{-1'}) = I$ replaced by Diag $((S')^{-1}S^{-1}) = I$, and the target matrix H replaced by H'_F. Because the constraints on S' and H'_F in Eq. (6) match those imposed in the two-way SIMPLIMAX method on T_s and H in Eq. (3), the minimum of Eq. (6) is found by applying the SIMPLIMAX procedure to $(\mathbb{U} \otimes \mathbb{T})G'_F$. It is worth noting that by applying SIMPLIMAX to $(\mathbb{U} \otimes \mathbb{T})G'_F$ we do not only find the best oblique rotation matrix S (given U and T), but also find *which* are the smallest core elements after rotation by this S (as indicated by the zeros in the matrix H found).

Upon permuting the three-way array, the updates for T and H, and for U and H can be found completely analogously. Thus, by repeatedly applying two-way SIMPLIMAX to a supermatrix with partly rotated core elements, we minimize $\sigma(S,T,U,H)$ over a subset of the parameters, while keeping the other parameters fixed. Hence, the above iterative procedure monotonically decreases (or at least not increases) the three-way SIMPLIMAX function. Rather than applying the full two-way SIMPLIMAX procedure (which is iterative itself), one may use only a few cycles (called "inner iterations") for updating the rotation and target for each mode. Because each update in two-way SIMPLIMAX decreases (or at least not increases) the two-way SIMPLIMAX function at hand, it does the same for the full three-way SIMPLIMAX function, provided that each two-way SIMPLIMAX updating is started with the current values of the matrix to be updated; this updating may be supplemented with a number of two-way SIMPLIMAX runs from different starting positions, and, in case it leads to a better solution, this better solution is used instead of the original update. Because it is thus guaranteed that each step of the procedure decreases the function value, and because the function value is bounded below (by zero), the function will converge to a stable value, although not necessarily the global minimum. To increase the chance of finding the global minimum, it is recommended to run the algorithm from several (randomly chosen) starts for S, T and U.

The algorithm for one three-way SIMPLIMAX run can schematically be summarized as follows:

Step 1a. Choose n_{in} (number of SIMPLIMAX iterations to be used within each two-way SIMPLIMAX updating), κ (number of additional randomly started two-way SIMPLIMAX runs) and ε (convergence criterion).

Step 1b. Initialize S, T and U, and choose m (the number of zeros in the target core).

Step 1c. Set H equal to the current rotated core, and set the m smallest elements of H equal to 0.

Step 1d. Compute the function value.

Step 2. Set $\sigma^{\text{old}} = \sigma$.

Step 3a. Update S and H from the current position by n_{in} two-way SIMPLIMAX cycles.

Step 3b. (Optional) Compute κ alternative updates for S by means of n_{in} two-way SIMPLIMAX cycles each, started randomly; use the best from these runs and the run in Step 3a, to update S and H.

Step 4a. Update $\boldsymbol{T}$ and $\boldsymbol{H}$ from the current position by n_{in} two-way SIMPLIMAX cycles.

Step 4b. (Optional) Compute κ alternative updates for $\boldsymbol{T}$ by means of n_{in} two-way SIMPLIMAX cycles each, started randomly; use the best from these runs, and the run in Step 4a, to update $\boldsymbol{T}$ and $\boldsymbol{H}$.

Step 5a. Update $\boldsymbol{U}$ and $\boldsymbol{H}$ from the current position by n_{in} two-way SIMPLIMAX cycles.

Step 5b. (Optional) Compute κ alternative updates for $\boldsymbol{U}$ by means of n_{in} two-way SIMPLIMAX cycles each, started randomly; use the best from these runs, and the run in Step 5a, to update $\boldsymbol{U}$ and $\boldsymbol{H}$.

Step 6. Compute σ. If $(\sigma^{\text{old}} - \sigma) > \sigma^{\text{old}} * \varepsilon$ and the number of iterations is still lower than a preset maximum, then return to Step 2; else consider the algorithm converged.

To increase the chance that the global minimum is attained, this complete procedure is repeated from a number of different random starting configurations. To get some insight into how many random starts suffice, the performance of the algorithm, and its sensitivity to hitting suboptimal solutions are considered in the next section.

3. Performance of the Three-way SIMPLIMAX Algorithm

As the two-way SIMPLIMAX algorithm was rather prone to hitting suboptimal solutions, we may expect the present procedure to suffer even more from this problem. The reason is that the proposed algorithm combines and iteratively applies three two-way SIMPLIMAX procedures, each of which is prone to hitting suboptimal solutions. As soon as one of the three parameter matrices (say $\boldsymbol{S}$) is associated with nonoptimal positions of the nonzero values in the target core, the other parameter matrices ($\boldsymbol{T}$ and $\boldsymbol{U}$) will be adjusted towards a target core with zero elements at mostly the same positions, and thus the positions of the zero values in the target core will usually not be adjusted anymore. Since any of the three updating procedures can thus cause the algorithm to lead to a suboptimal solution, the method can be expected to hit suboptimal solutions very often.

To get some insight into the extent of the problem, we tested the algorithm (programmed in PCMATLAB and PASCAL, available from the author) on 100 core arrays. These core arrays were produced by 3MFA applied to data that satisfied a particular 3MFA model exactly (in 50 cases), or approximately (in 50 other cases). The main purpose of these analyses was to gain insight in the number of random starts needed to guarantee that the global minimum is attained. In addition, we hoped to gain insight in the efficiency of the algorithm.

The 100 data sets were constructed as follows. For the construction of each data set a $(P \times Q \times R)$ core array $\boldsymbol{H}_0$ with a particular simple structure (which varied over five conditions, as specified below) was first chosen. This matrix was multiplied by matrices $\boldsymbol{A}(50 \times P)$, $\boldsymbol{B}(10 \times Q)$ and $\boldsymbol{C}(10 \times R)$, with random elements; for $\boldsymbol{A}$, which is here chosen to simulate the mode of the observation units, the elements were drawn from the standard normal distribution; for $\boldsymbol{B}$ and $\boldsymbol{C}$ (modes representing, for instance, fixed variables and measurement instances) the elements were drawn from the uniform [-0.5,0.5] distribution. The multiplication of these matrices with $\boldsymbol{H}_0$ results in the 50 × 10 × 10 array $\boldsymbol{X}$. For each condition ten such data sets were constructed, leading to a total of 50 three-way data sets. These data sets represent the perfect data, because a 3MFA in (P,Q,R) dimensions will yield a perfect fit, and by means of oblique rotations of the core, the core can be simplified back into its original form. Because error free data do not occur in practice, a set of more realistic data was constructed as well: Fifty data sets were constructed by adding a random three-way array $\boldsymbol{N}$ to each of the former data sets. The elements of this array were drawn from a standard normal distribution, and normalized such that their sum of squares equalled 0.25 times the sum of squares of the three-way array to which they were added.

As mentioned above, five different types of core arrays $\boldsymbol{H}_0$ were employed. The frontal planes of these are described below, where each × indicates a random element drawn from the uniform [0.5,1] distribution:

Case 1 (3 × 3 × 3 with 9 nonzero elements):

$$\boldsymbol{H}_1=\begin{pmatrix}\times&0&0\\0&0&\times\\0&\times&0\end{pmatrix},\quad \boldsymbol{H}_2=\begin{pmatrix}0&0&\times\\0&\times&0\\\times&0&0\end{pmatrix},\quad \boldsymbol{H}_3=\begin{pmatrix}0&\times&0\\\times&0&0\\0&0&\times\end{pmatrix}.$$

Case 2 (3 × 3 × 3 with 6 nonzero elements):

$$\boldsymbol{H}_1=\begin{pmatrix}\times&0&0\\0&0&\times\\0&0&0\end{pmatrix},\quad \boldsymbol{H}_2=\begin{pmatrix}0&0&0\\0&\times&0\\\times&0&0\end{pmatrix},\quad \boldsymbol{H}_3=\begin{pmatrix}0&\times&0\\0&0&0\\0&0&\times\end{pmatrix}.$$

Case 3 (3 × 3 × 3 with 4 nonzero elements):

$$\boldsymbol{H}_1=\begin{pmatrix}\times&0&0\\0&0&\times\\0&0&0\end{pmatrix},\quad \boldsymbol{H}_2=\begin{pmatrix}0&0&0\\0&\times&0\\0&0&0\end{pmatrix},\quad \boldsymbol{H}_3=\begin{pmatrix}0&0&0\\0&0&0\\0&0&\times\end{pmatrix}.$$

Case 4 (4 × 3 × 2 with 6 nonzero elements):

$$\boldsymbol{H}_1=\begin{pmatrix}\times&0&0\\0&\times&0\\0&0&\times\\0&0&0\end{pmatrix},\quad \boldsymbol{H}_2=\begin{pmatrix}0&0&0\\\times&0&0\\0&\times&0\\0&0&\times\end{pmatrix}.$$

Case 5 (4 × 3 × 2 with 4 nonzero elements):

$$\boldsymbol{H}_1=\begin{pmatrix}\times&0&0\\0&\times&0\\0&0&0\\0&0&0\end{pmatrix},\quad \boldsymbol{H}_2=\begin{pmatrix}0&0&0\\0&0&0\\0&\times&0\\0&0&\times\end{pmatrix}.$$

Each of the 50 data sets was first analyzed by 3MFA, employing the least squares algorithm proposed by Kroonenberg et al. (1989), taking the best of three runs (one of which was started rationally, the other two randomly). The resulting cores were analyzed by three-way SIMPLIMAX. The number m employed in the SIMPLIMAX analyses was always taken exactly equal to the number of zeros used in the underlying cores. In this way, we knew in advance that SIMPLIMAX of the cores *based on the perfect data* could lead to a core with m elements exactly equal to zero. Thus, for these data we know that the global minimum of the function value is 0. For imperfect data, we chose the same values for m as for the perfect data, expecting that we would be able to get m values relatively close to zero; for these data, the global minimum is not known.

The main interest of the present analyses was to inspect the sensitivity of the three-way SIMPLIMAX algorithm to suboptimal solutions. First, we did some informal experimentation to study the effect of different choices for n_{in} (the number of iterations in each two-way SIMPLIMAX run) and for κ (the optional number of additional randomly started two-way SIMPLIMAX runs). It turned out that taking $n_{\text{in}} = 5$ works considerably better (in terms of avoiding suboptimal solutions) than taking $n_{\text{in}} = 1$, but that taking $n_{\text{in}} = 10$ or larger is hardly worth the increase in time consumption. Furthermore, it was found that using $\kappa = 50$ additional randomly started runs improved the algorithm's ability to find the global minimum considerably; it was observed, however, that after the second major iteration (i.e., after the second time that the Steps 3 – 6 were completed), these additional runs hardly affected the solution. Moreover, it was observed that the positions of the zero values in the target core seemed to change only in the first few iterations; after that, the updatings of $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ only served to further decrease the values of the smallest core elements, rather than changing their locations. So we decided to use $\kappa = 50$ only in the first two major iterations, and $\kappa = 0$ afterwards, and we took $n_{\text{in}} = 5$. To avoid very long iterative processes, we always stopped a run after 1000 iterations. Unfortunately, even after these first improvements, a single run of the algorithm still has a high probability of landing in a suboptimal solution,[1] as can be deduced from Table 1. In the Case 2 conditions this probability is close to 90%, and it is also very high in the Case 1 conditions. Hence, in these conditions we certainly need a large number of random restarts. In fact, for one data set, the global minimum was found in only one out of the 200 runs. Hence, it seems that we actually need at least 200 randomly started runs. Fortunately, as will be seen later, a single run of the algorithm does not take much computation time, even when many iterations are needed.

Upon checking each individual analysis of the 50 perfect data sets, we found that a value below 0.0001 was always reached. (For these data, the global minimum is exactly 0, but this value can only be expected to be approximated, the accuracy of which depends on the number of iterations used). In fact, in most cases the obtained value was much smaller than 0.0001, but even if it was not, and even if the method actually attained a (very small) suboptimal function value, the difference from the

[1] A solution was considered a suboptimum for the perfect data, if the function value was more than 0.0001 higher than the global minimum, or, in case of imperfect data, than the best observed value.

Table 1
Average number of locally optimal solutions (out of 200 runs)

	Case 1	Case 2	Case 3	Case 4	Case 5
Perfect data	147	184	63	118	37
Imperfect data	155	179	79	113	150

global minimum is negligible for all practical purposes. Hence, it can be concluded that the global minimum was attained sufficiently closely in all 50 cases.

Having seen that the three-way SIMPLIMAX method indeed needs large numbers of randomly started runs, one may wonder if the method is still practically feasible. The PASCAL program written for this method, implemented on a pc with a Pentium (100 MHz) processor, a set of 200 runs used computation times in the range of 1600–2075 s for Case 1, 289–1114 s for Case 2, 238–562 s for Case 3, 454–1044 s for Case 4, and 264–1492 s for Case 5. This shows that it is indeed rather time consuming to perform one three-way SIMPLIMAX analysis with 200 runs, but not unfeasible. In practice, one will usually perform several three-way SIMPLIMAX analyses, with different numbers of nonzero elements, so a full analysis of one core array will require considerable computational effort, and any procedure to accelerate this process (e.g., based on a procedure for avoiding suboptimal solutions) is most welcome. Nevertheless, even the present procedure is still feasible within an acceptable period of time. Moreover, with the availability of considerably faster machines, computation times can be reduced to several minutes.

4. Illustrative analyses

To see how the three-way SIMPLIMAX method works in practical applications, we used it for simple structure rotation of two core arrays, one resulting from an artificial data set, and one resulting from an empirical data set. The first core array results from one of the imperfect Case 5 data sets. This (randomly picked) $4 \times 3 \times 2$ array resulted from a data set based on a core with 20 zero elements. We applied three-way SIMPLIMAX to this core, using different values for m. Specifically, we did SIMPLIMAX analyses using $m = 19$ through $m = 22$ zeros. In fact, we also did a few three-way SIMPLIMAX runs with $m = 18$, but soon found that the function value was exactly zero. For each complete SIMPLIMAX analysis, we used the same options as in the study on the performance of the algorithm (i.e., 200 random starts, at most 5 inner iterations, 50 additional random starts for each two-way SIMPLIMAX update in the first two main cycles, and a maximum of 1000 main iterations per run). The results for $m = 18$ through 22 are reported in Table 2. It can be seen that for $m = 18$ through 20, the function value is very small, but that from $m = 21$ on, the value is nonnegligible. Here, the relation between the number of zeros (m) and the sum of squares (σ) indicates that $m = 20$ is the highest value that still gives relatively small smallest elements. The giant step from $m = 20$ to $m = 21$ (representing a multiplication by more than 1000 of the average of the

Table 2
Three-way SIMPLIMAX applied to a selected artificial core array: function values for different values of m

m	σ
18	0.000
19	0.000
20	0.002
21	3.994
22	13.267

squared smallest elements) suggests that $m = 21$ is based on too high a value for m. Hence, for this data array one would choose the $m = 20$ solution as the most useful one, which corresponds exactly to the number of zeros used in constructing this array.

The second application of three-way SIMPLIMAX is meant to simplify a $3 \times 3 \times 2$ core reported by Kroonenberg (1994, p. 90). In this study, 82 subjects were measured on 5 variables (pertaining to performance and drunkenness) at 8 occasions (at which different doses of alcohol had been administered to them); the reported core array resulted from a 3MFA applied to these data, followed by varimax of $\boldsymbol{B}$ (the variables loading matrix) and a special rotation of $\boldsymbol{A}$ (the subject components matrix). For more details, see Kroonenberg (p.85ff). The present $3 \times 3 \times 2$ core array has been rotated here by means of three-way SIMPLIMAX. We used the values $m = 12$ through $m = 16$. For $m = 12$ and $m = 13$, three-way SIMPLIMAX found a solution in which the smallest m elements were zero, up to the accuracy implied by the convergence criterion used. In fact, it can be proven that any $3 \times 3 \times 2$ core can be transformed into a core with as many as twelve or thirteen exactly zero elements (Ten Berge, 1996), which explains what we found here. Hence, the only nontrivial applications of SIMPLIMAX are those with $m = 14$, $m = 15$ and $m = 16$. Again, in each complete SIMPLIMAX analysis, we used the same options as before (200 random starts, at most 5 inner iterations, 50 additional random starts for each two-way SIMPLIMAX update in the first two main cycles, and a maximum of 1000 main iterations per run). The rotated cores, as well as the values of the function σ (the sums of smallest squared core elements) are given in Table 3, with the (18-m) highest elements in bold face. It can be seen that the core rotated towards 14 zeros indeed gives only four important core elements, the others being about ten times as small or smaller. Even in the core rotated to 15 zeros, the high values are at least eight times as large as the small values. However, in the core rotated to 16 zeros, the smallest elements were no longer negligible compared to the high values. It can hence be concluded that this $3 \times 3 \times 2$ core can be simplified tremendously, and, in fact, the main relations between components can be described in 3 or 4 terms. This also illustrates that a solution that can be simplified, without loss of fit, by fixing a certain number of elements to zero, can, still without loss of fit, be further simplified by making a number of elements very close to zero, and hence negligible.

To study the solution with 15 near zero core elements more closely, it is interesting to see what transformations are used to find this core, and what matrices$\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{C}}$

Table 3
Three-way SIMPLIMAX applied to the core reported by Kroonenberg (1994)

Original core (Frontal planes next to each other)					
24	−26	−5	−2	1	1
18	11	9	−3	−3	0
−2	−12	15	−1	−6	7
Core rotated to 13 zeros				$\sigma = 0.000$	
35.4	0	0	**3.1**	0	0
0	**24.0**	0	0	0	0
0	0	**18.9**	0	0	**12.4**
Core rotated to 14 zeros				$\sigma = 2.958$	
35.5	0.0	0.0	0.9	0.3	0.0
0.0	**23.5**	0.0	0.2	−1.4	0.0
0.0	0.0	**18.8**	0.0	0.0	**11.3**
Core rotated to 15 zeros				$\sigma = 10.967$	
33.9	0.0	−0.0	1.9	−0.6	0.0
0.0	**25.7**	−0.1	−0.5	−2.5	0.0
0.0	0.0	0.4	−0.0	0.0	**21.6**
Core rotated to 16 zeros				$\sigma = 276.355$	
36.6	0.0	−0.7	−0.2	0.7	−1.4
0.0	**25.8**	1.4	0.7	2.7	−0.4
0.4	0.8	−9.3	2.2	−2.2	12.9

correspond to the thus rotated core. The inverses of the matrices $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ are the oblique rotations which rotate the matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, to$\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{C}}$ respectively. These rotation matrices, as well as the resulting inner products between columns of the matrices$\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{C}}$ are given in Table 4. Because Kroonenberg (1994) has only reported the elements of $\boldsymbol{B}$ (repeated in Table 4), we could only compute $\tilde{\boldsymbol{B}}$, and not $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{C}}$.

It can be seen from Table 4 that there has been only little transformation in the A-mode, but the loadings for the B-mode have been affected considerably, and so have those for the C-mode. Specifically, the loadings in $\boldsymbol{B}$ were very simple before our SIMPLIMAX rotation, whereas the rotated loadings indicate contrasts that are more difficult to interpret. Apparently, the enormous gain in simplicity in the core is, in this case, offset by a loss in simplicity of the B-mode loadings. One could say that the complicated (contrastlike) relations in the core have now been moved to contrasts in the matrix $\boldsymbol{B}$. The advantage of using the SIMPLIMAX rotated core is that we no longer need to bother about the complicated problem of taking into account many triple product terms (each involving the contribution of one core element). We have not totally eliminated these intricate relations (which we could not even hope to, since they are in the data), but have moved them to a position where they may be easier to handle: They are now localized in the matrix $\boldsymbol{B}$, and interpreting these

Table 4
Rotation matrices inner product matrices, and the rotated version of $\boldsymbol{B}$ as a result of three-way SIMPLIMAX applied to the Kroonenberg (1994) core

S^{-1}			$\tilde{A}'\tilde{A}$		
1.00	−0.01	0.09	1.00		
0.11	1.00	0.15	0.10	1.00	
−0.01	0.02	0.98	0.10	0.17	1.00
T^{-1}			$\tilde{B}'\tilde{B}$		
0.70	0.68	−0.11	1.00		
−0.69	0.66	−0.64	−0.03	1.00	
−0.17	0.31	0.76	0.24	−0.26	1.00
U^{-1}			$\tilde{C}'\tilde{C}$		
1.00	0.90		1.00		
−0.10	0.44		0.85	1.00	

B-mode loadings	Before rotation			After rotation		
Auditory reaction time	0.63	0.02	−0.08	0.44	0.42	−0.14
Visual reaction time	0.57	−0.01	0.05	0.40	0.40	−0.02
Complex reaction time	0.53	−0.01	0.02	0.37	0.36	−0.04
Arithmetic computation	0.00	1.00	0.00	−0.69	0.66	−0.64
Self-rated drunkenness	0.02	0.00	0.99	−0.16	0.32	0.75

contrasts is a matter of interpreting the elements of $\boldsymbol{B}$ only, without bothering about complicated relations with the different dimensions of $\boldsymbol{A}$ and $\boldsymbol{C}$.

In the present empirical example, one may maintain that the simple structure that had been obtained for $\boldsymbol{B}$ is so useful that we do not wish to affect this. In that case, we might still try to simplify the solution by means of oblique rotations of the A- and C-mode only. In Section 6, it is discussed how the three-way SIMPLIMAX method can be adjusted for this and similar purposes. At the end of Section 6, this procedure will be used for the reanalysis of the present empirical core matrix.

5. N-way SIMPLIMAX

The basic principle of three-way SIMPLIMAX is easily generalized to four- and higher way core arrays. The only modification is to add extra steps for updating transformation matrices for the fourth and higher modes.

For the general N-way SIMPLIMAX, we may expect even stronger sensitivity to hitting suboptimal solutions. One way to avoid this problem is to restrict the number of modes of which the components will be rotated to three, or even less. This possibility is discussed in the next section.

6. Rotation in a subset of the modes of the N-mode core array

As mentioned above, in N-way SIMPLIMAX it is probably wise to rotate the core over a subset of all permissible directions only. In fact, even in three-way

Table 5
Results of three-way SIMPLIMAX rotation over the A- and C-mode, applied to the core reported by Kroonenberg (1994)

Rotated core (frontal planes next to each other)					
30.2	−1.1	2.9	−0.2	−1.7	−1.6
−0.1	**26.5**	**9.7**	−2.2	0.6	0.2
0.0	−0.4	0.6	−2.3	**-13.8**	**16.3**
	S^{-1}			$\tilde{A}'\tilde{A}$	
0.76	−0.83	0.13	1.00		
0.65	0.55	0.17	−0.27	1.00	
0.00	0.01	0.98	0.21	0.00	1.00
U^{-1}			$\tilde{C}'\tilde{C}$		
0.99	0.89		1.00		
−0.11	0.44		0.84	1.00	

SIMPLIMAX it may be wise to rotate the core in only one or two directions. This can be accomplished simply by iteratively updating only a subset of the transformation matrices, while keeping the other transformation matrices fixed to the identity matrix. In the extreme case where rotation is applied in only one direction, the method reduces to the two-way SIMPLIMAX method.

An example of a case where it may be desirable to rotate in only two directions is the rotation of the empirical 3 × 3 × 2 core array in the previous section. Because $\boldsymbol{B}$ has a very clear simple structure, it may be preferable to keep $\boldsymbol{B}$ fixed, and to rotate the core to simplicity by rotations of the A- and C-mode only. We applied the thus modified three-way SIMPLIMAX procedure for $m = 12$ through $m = 15$. In the present version of SIMPLIMAX with rotation in only two directions, using the $\kappa = 50$ random restarts in the initial phases of the algorithm now caused the method to land in a suboptimal solution for $m = 13$ even when using 200 random starts (as was found upon comparison with the results from 200 runs with $\kappa = 0$; in fact, it turned out that all 200 runs led to the same suboptimal solution). Therefore, for this case we did not use random restarts for two-way SIMPLIMAX in the initial phases; otherwise, we used the same options for the algorithm as before. We now found σ=7.9 (m=12), σ=25.9 (m=13), σ=73.0 (m=14) and σ=119.2 (m=15). It turned out that only for $m = 12$ and $m = 13$ the smallest core elements were considerably smaller than the largest core elements. Moreover, there is a large jump in the size of the function values when going from $m = 13$ to $m = 14$. For these reasons, the $m = 13$ solution was considered the most attractive one available for this core, by means of oblique rotation of the A- and C-mode only. The resulting core, as well as the transformation matrices and inner product matrices are reported in Table 5. It can be seen that this solution is in part similar to the one found with $m = 15$ using rotation in all modes (see Table 3), but that the present solution differs from the latter in that it has two sizeable interaction terms, which apparently suffice to capture most of the three-way interactions in the present data. It can be concluded that, having thus reduced the core to 5 large elements and 13 small elements, we have succeeded in considerably simplifying the original core, without sacrificing the main simplicity found in the original solution (in terms of $\boldsymbol{B}$).

7. Using three-way SIMPLIMAX for other purposes

An alternative approach to finding 3MFA solutions with simple cores is to constrain certain elements of the core to be zero (see Kiers, 1992; Kiers et al. 1997). An advantage of this approach over three-way SIMPLIMAX is that it guarantees certain elements in the core to be exactly zero, whereas with SIMPLIMAX it may happen that purported small elements are by no means close to zero (e.g., see the last panel in Table 3). One problem with this constrained 3MFA approach, however, is that we may well know *how simple* we would like the core to be (e.g., we could like a $3 \times 3 \times 3$ core to have 21 zeros), but it is much more difficult to choose the *form* of this simple core (*which* 21 elements should be constrained to 0). With three-way SIMPLIMAX we do not constrain the core to have exact zeros, but we do transform the core such that it has m values that are approximately zero, the positions of which are found by the method itself. Now these positions can be used in a subsequent constrained 3MFA where the approximately zero values are constrained to be exactly zero. In this way, we have a procedure for both constraining core elements to zero and finding which elements to constrain to zero.

Another situation where three-way SIMPLIMAX can be useful is the following. From a theoretical point of view, it is interesting to study how many elements of a core can be made exactly zero by means of oblique rotations. One way to study this is to impose zero constraints, and to verify if the corresponding 3MFA model fits the data perfectly. However, in doing so we do not only have to test every *number* of zero core elements, but also every *configuration* of zeros in the core. Since, usually, very many different configurations are possible, it is practically impossible to test if indeed a certain number of zeros is the maximal possible number of zeros to be obtained by oblique rotations.

With three-way SIMPLIMAX, we have an alternative possibility of testing how many elements of a core can be made exactly zero: By testing different values for m, and inspecting whether or not these lead to m elements exactly equal to zero, we can assess what is the largest number of zeros that can be found by oblique rotations of the core. In this procedure *we do not need to specify the configuration* of the zero values in advance. The method finds the form of the configuration itself, and we no longer have to test all conceivable configurations. Hence, if SIMPLIMAX indicates that a certain number of zeros (m_{m}) is the maximal possible number of zeros to be obtained by oblique rotations, the only remaining uncertainty is that the result is based on a suboptimal solution found when using $m_{\mathrm{m}} + 1$ zeros, but this possibility can be ruled out by using a sufficient number of random starts for this analysis.

8. Discussion

In the present paper it has been described how SIMPLIMAX can be generalized so as to rotate a core array obliquely in as many directions as desired. It has also been seen that the method works well in that the method indeed finds simple cores when these can be found by oblique rotations. However, the method requires a large number of randomly started runs to find the globally optimal solution. Therefore, alternative procedures for avoiding suboptimal solutions are most welcome. Nevertheless, three-way SIMPLIMAX leads us to a new horizon of possibilities: Rotation of a core to simplicity in three directions was hitherto only possible by means of orthogonal rotations; lacking the details of Kruskal's approach for oblique simplicity rotation of the core, the present approach is the only detailed procedure for oblique rotation of a core to simplicity in three (or, if desired, fewer) directions.

The three-way SIMPLIMAX method is not only useful as a method for oblique rotation to a simple core, but can also be used to reduce the number of nonzero core elements, as mentioned in the last section. In fact, by means of three-way SIMPLIMAX, we are able to study empirically how many elements of a three-way (core) array can be made zero by means of nonsingular transformations. For instance, it has been found consistently that we can (obliquely) rotate a $3 \times 3 \times 3$ array such that we end up with at least as many as 18 zeros. We now have the means to study empirically how many zero elements can be found in an arbitrary $P \times Q \times R$ core array. With the help of these empirical results, we can try to prove theoretically how many zeros can really be obtained by oblique rotations. This is of theoretical importance, because it provides insight into the attainable amount of simplicity of a core array. It is important from a practical point of view because it sets a lower bound to the amount of parsimony to be obtained for a particular core array. For instance, any reported array that has fewer zero elements than the number of zeros that can always be obtained by SIMPLIMAX, can be discarded as nonparsimonious if parsimony of the core is a desideratum. Furthermore, knowing that, for instance, a $5 \times 3 \times 2$ core can always be transformed so as to have 24 zeros (Murakami et al., 1998), implies that any attempt to simplify this array by three-way SIMPLIMAX should start with at least 24 zeros. For these reasons, a further study into the attainable simplicity of three-way arrays by means of nonsingular transformations is called for, and can be supported by empirical results on the basis of three-way SIMPLIMAX.

A further use of three-way SIMPLIMAX is in the study of uniqueness of parsimonious 3MFA models. Again, proving uniqueness requires a theoretical approach, but can very well be guided by empirical findings. Three-way SIMPLIMAX can be of use here, because it indicates, for instance, that a three-way array can be transformed to have a particular number of zeros *in many essentially different ways*, with configurations of zeros differing by more than permutations. Such discoveries are greatly facilitated by a method that finds the configuration of zeros *itself*.

The present paper has focussed on oblique rotations of the core, thus allowing for optimal flexibility in rotation of the core. However, in certain situations, it may be desirable to restrict oneself to orthogonal rotations of the core. One reason could be that orthogonal rotations do not affect the property that the square of each core element gives the (independent) contribution of the associated interaction to the total fit of the model. Another reason could be that it is sometimes important to use orthogonal (and hence independent) dimensions of concepts, subjects, or whatever the modes pertain to. In such cases, one may resort to three-way orthomax methods (Kiers, 1997). Alternatively, one may use an orthogonal variant of three-way SIMPLIMAX, which is readily constructed. The only modification is to replace each two-way SIMPLIMAX step for updating a rotation matrix, by an *orthogonal* two-way SIMPLIMAX step (as described by Kiers, 1994, p.578). In this way, it is even possible to use orthogonal rotations in some modes and oblique in others.

The three-way SIMPLIMAX rotation method proposed aims at simplifying the core. As has been mentioned, for the interpretation of a 3MFA solution it may be desirable that (some of) the component matrices are simple as well. Therefore, in SIMPLIMAX it is possible to use only a subset of the rotation matrices so that one can keep some of the component matrices fixed to a simple solution (which itself can have been obtained by simple structure rotation of that matrix). This has indeed been done in the second analysis of our illustrative 3MFA solution, and it turned out that SIMPLIMAX rotation of the core by means of only two matrices indeed still simplified the core considerably. However, it is conceivable that for certain 3MFA solutions, by fixing one or two of the rotation matrices insufficient freedom is left to simplify the core. In such cases, we would need an alternative approach: It would be desirable to optimize a criterion that *combines* simplicity of the core and simplicity of the component matrices. Specifically, it would be useful to have a method that finds those rotation matrices $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ that optimize the (weighted) "average" of the simplicity of the core and the component matrices. In further research, it will be studied how such a criterion should be defined, and how it can be optimized.

Acknowledgements

The author is obliged to Jos ten Berge for stimulating this research, and for comments on earlier versions of this paper.

References

Browne, M.W., 1972. Oblique rotation to a partially specified target. British J. Math. Statist. Psychol. 25, 207-212.

Cohen, H.S, 1974. Three-mode rotation to approximate INDSCAL structure (TRIAS). Paper presented at the Psychometric Society Meeting, Palo Alto.

Cohen, H.S., 1975. Further thoughts on three-mode rotation to INDSCAL structure, with jackknifed confidence regions for points. Paper presented at U.S.–Japan seminar on Theory, Methods and Applications of Multidimensional Scaling and Related Techniques, La Jolla.

Crawford, C.B., Ferguson, G.A., 1970. A general rotation criterion and its use in orthogonal rotation. Psychometrika 35, 321–332.

Jennrich, R.I., 1970. Orthogonal rotation algorithms. Psychometrika 35, 229–235.

Kiers, H.A.L., 1992. TUCKALS core rotations and constrained TUCKALS modelling. Statistica Applicata 4, 659–667.

Kiers, H.A.L., 1994. SIMPLIMAX: Oblique rotation to an optimal target with simple structure. Psychometrika 59, 567–579.

Kiers, H.A.L., 1997. Three-mode Orthomax rotation. Psychometrika 62, 579–598.

Kiers, H.A.L., Ten Berge, J.M.F., Rocci, R., 1997. Uniqueness of three-mode factor models with sparse cores: The $3 \times 3 \times 3$ case. Psychometrika 62, 349–374.

Kroonenberg, P.M., 1983. Three-Mode Principal Component Analysis: Theory and Applications. DSWO press, Leiden.

Kroonenberg, P.M., 1994. The TUCKALS line: a suite of programs for three-way data analysis. Comput. Statist. Data Anal. 18, 73–96.

Kroonenberg, P.M., De Leeuw, J., 1980. Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika 45, 69–97.

Kroonenberg, P.M., Ten Berge, J.M.F., Brouwer, P., Kiers, H.A.L., 1989. Gram-Schmidt versus Bauer-Rutishauser in alternating least-squares algorithms for three-mode component analysis. Comput. Statist. Quart. 5, 81–87.

Kruskal, J.B., 1988. Simple structure for three-way data: a new method intermediate between 3-mode factor analysis and PARAFAC-CANDECOMP. Paper presented at the 53rd Annual Meeting of the Psychometric Society, Los Angeles.

MacCallum, R.C., 1976. Transformations of a three-mode multidimensional scaling solution to INDSCAL form. Psychometrika 41, 385–400.

Murakami, T., 1983. Quasi three-mode principal component analysis - A method for assessing factor change. Behaviormetrika 14, 27–48.

Murakami, T., Ten Berge, J.M.F., Kiers, H.A.L., 1998. A case of extreme simplicity of the core matrix in three-mode principal components analysis. Psychometrika 63.

Ten Berge, J.M.F., 1996. Some general results for Murakami zeros in $p \times p \times 2$ arrays ($p \leq 5$). Unpublished note.

Tucker, L.R., 1966. Some mathematical notes on three-mode factor analysis. Psychometrika 31, 279–311.