KUNST

MINISSA

Michigan Israel Nijmegen Integrated Smallest Space Analysis

Nol Bendermacher
Maria Thissen,

RTOG-FSW
Radboud University Nijmegen

April 2012

Content

1 MINISSA: PROGRAM DESCRIPTION

If one is given a map of the Netherlands (Holland), it is a simple matter to construct a table of distances between the most important cities. But is it possible to do the reverse? Can one reconstruct the map from a table of distances? If you acquainted with linear algebra, you will answer yes. However, can it also be done if the distances are more or less inaccurate? Figure 1 gives a map of the Netherlands with the location of sixteen cities*.

Figure 1: A map of the Netherlands with 16 cities.

Figure 2 shows a table of railroad distances between these sixteen cities. Figure 3 shows the reconstruction of the map by MINISSA from the railroad distances. The map is reconstructed pretty well except for a rather heavy distortion around the lake *IJsselmeer* in the northern part, because there are no railroad connections over the water, not even over the dike *Afsluitdijk* between cities 3 (*Den Helder*) and 1 (*Leeuwarden*). Note also that the orientation may be different and that the map may be reflected.

---

* The cities are: Leeuwarden (1), Groningen (2), Den Helder (3), Zwolle (4), Amsterdam (5), Deventer (6), Enschede (7), Den Haag (8), Utrecht (9), Rotterdam (10), Nijmegen (11), Middelburg (12), Tilburg (13), Eindhoven (14), Venlo (15) and Maastricht (16).

MINISSA bases its solution merely on the order of the given 'distances'. If, for example, we would replace the railroad distances by their squares, the solution would still be the same! Therefore, the input to the program is not called distances, but **dissimilarities** or, their counterparts, **similarities**. The general term for both dissimilarities and dissimilarities is **proximities**.

MINISSA can also handle other metrics than the usual Euclidean.

The first version of MINISSA was developed by J.C. Lingoes and E.E.Ch.I. Roskam in 1968.

```
  0
 54   0
287 298   0
 95 106 192   0
206 217  81 111   0
126 137 184  31 103   0
162 173 245  67 164  61   0
242 253 121 147  63 139 200   0
182 193 120  87  39  79 140  60   0
238 249 144 143  86 135 196  23  56   0
190 201 197  95 116  64 101 137  77 133   0
365 376 271 270 213 243 280 150 183 127 179   0
252 263 190 157 109 129 166  94  70  71  65 114   0
262 273 200 167 119 139 176 131  80 108  75 151  37   0
251 262 252 156 171 125 162 183 132 160  61 203  89  52   0
320 331 298 225 217 194 231 229 178 206 130 249 135  98  69   0
```

Figure 2: Railroad distances between 16 cities.

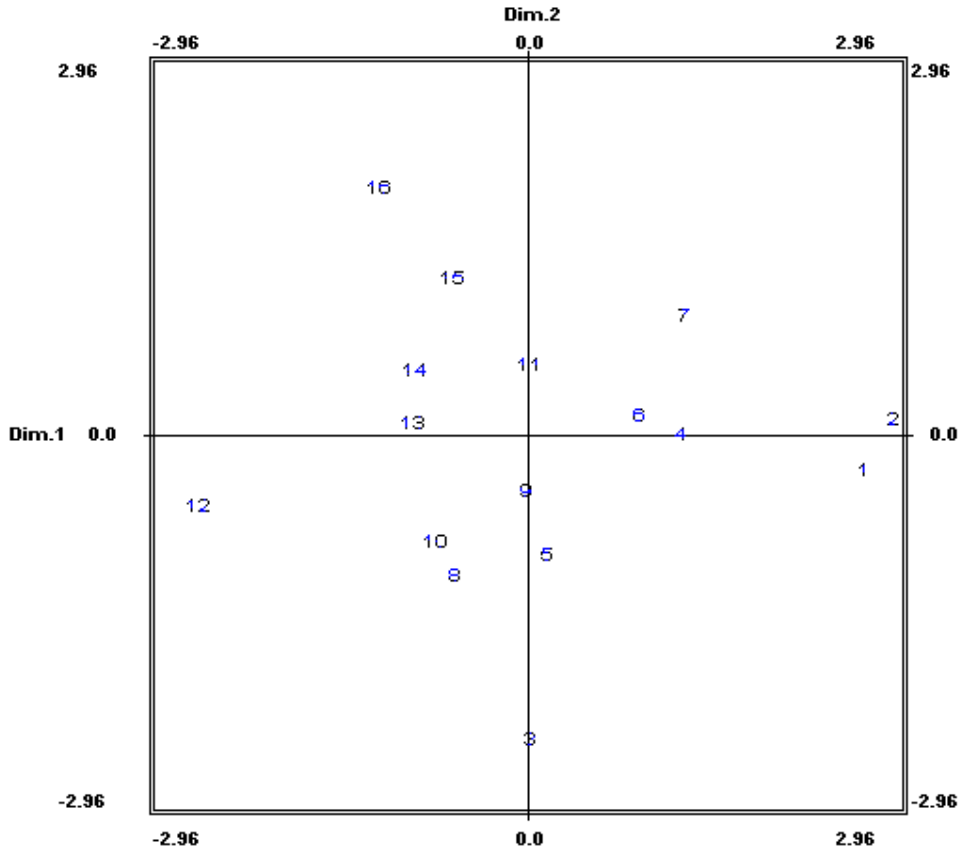

Figure 3: A two dimensional map with 16 cities constructed from railroad distances.

A proximity is a measure of similarity or dissimilarity between two objects. The objects are usually called **stimuli**. The rows and columns of the matrix refer to the same objects and the lower left part must contain the same values as the upper right part: cell (i,j) must contain the same value as cell (j,i)*. The values on the diagonal are not used. In the resulting map, the *solution*, each object will be represented by a point in the solution space. In MINISSA, the objects are called stimuli or variables.

A MINISSA analysis may be useful if there are no predefined characteristics from which one can derive the (dis)similarities between objects, but if one wants to find out what characteristics are important for the perceived (dis)similarity of the objects. In such a situation, the researcher tries to interpret the dimensions of the MINISSA solution or the clusters of stimuli.

MINISSA uses only the order in the proximities, not their actual values. And if it cannot find a perfect solution (one that preserves the order of the distances), it will try to compute the best solution it can find. The program is not restricted to two-dimensional space, but can use up to nine dimensions.

---

* If a cell (i,j) differs from its counterpart (j,i) MINISSA will replace them both by their average.

Now it will be clear that nobody will use such a program to produce a geometrical map of a country. However, there are other types of data for which a geometrical representation may yield useful insights.

Extra (1978) analyzed two groups of children learning Dutch: the L1-group consisting of 15 Dutch children (average age 4.4) and the L2-group consisting of 15 Turkish children (average age 14.4). From a number of tasks each child received a score on each of ten language characteristics indicating the percentage of times its response was correct. Similarities between children were then computed as rank correlations based on these scores.

Extra measured the consistency within and between groups by averaging these rank correlations. The internal consistency of group L1 was 0.85, the internal consistency of group L2 0.71 and the consistency between groups was 0.66.

In order to find out why the consistency of group L2 was hardly larger than that between groups, MINISSA was used to make a two-dimensional map. The result is shown in figure 4. Members of L2 are marked with a star. From the picture, it shows that two L2-children (2 and 14) take extreme positions that explain the low internal consistency of their group.

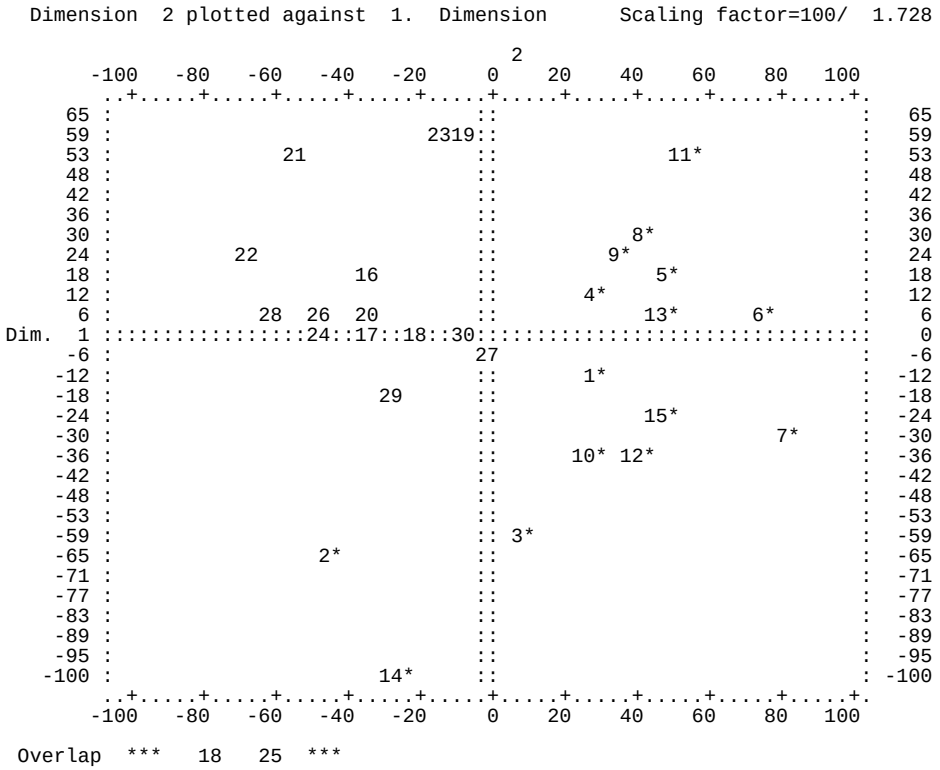

Figure 4: A two-dimensional map of Extra's language acquisition data (1-15 Turkish, 16-30 Dutch).

The input to MINISSA may come from different sources:

- One may ask several subjects to cluster the stimuli in categories so that stimuli in the same category are more similar to each other than to those in other categories. The similarity of two stimuli is then derived from the number of subjects that put them in the same category.

- In another method each stimulus pair is presented to a number of subjects and their similarity is derived from the number of subjects that judges them as identical or similar.
- Proximity between persons or groups may be derived from the amount of interaction between them: joint activities, traffic and so on.
- Proximity data can also be derived from non-symmetric matrices like a matrix of cases by variables. Any type of correlation or association measure may be interpreted as a matrix of similarities.

The results may be interpreted by giving some meaning to the dimensions of the solution space (or to rotations of the dimensions), but it is also possible to look for characteristic features of clusters of stimuli in the space.

MINISSA represents the stimuli in some kind of space, where the number of dimensions as well as the definition of distance can be chosen by the user. Since one may not know what the appropriate number of dimensions should be, the program offers the opportunity to specify a range of dimensionalities, for instance 1 to 3. MINISSA will first search for a solution in the largest number of dimensions (for instance 3) and then reduce the number of dimensions step by step until the solution becomes too bad to go on.

There is also some freedom in the choice of the distance formula to be used. The 'normal' or Euclidean formula for the distance between two points `p` and `q` in `k`-dimensional space is:

$$d_{pq} = \left(\sum_{i=1}^{k} (p_i - q_i)^2\right)^{\frac{1}{2}}$$

But there is a more general formula where the constant 2 is replaced by a so-called Minkowski-parameter `m`:

$$d_{pq} = \left(\sum_{i=1}^{k} |p_i - q_i|^m\right)^{\frac{1}{m}}$$

In MINISSA one can choose any Minkowski-parameter between 0 and 15 as long as it is positive. The parameter can also be infinite. Even non-integer values are allowed. It is however not advised to choose a value less than one, since then the theorem of the triangle inequality does not hold anymore:

$$(d_{xy} + d_{yz} \geq d_{xz})$$

Three of these distance parameters are of special interest:

city block metric: If the Minkowski-parameter is 1 the distance between two points is just the sum of the differences between their corresponding coordinates. The metric owes its name to the fact that it is the distance one has to walk if one is forced to follow a rectangular street pattern in a city (and the streets are parallel to the coordinate system).

Euclidean metric: This is the 'normal' metric with Minkowski-parameter 2. The coordinate differences are weighted according to their size, so the largest differences have more impact than the smaller ones.

dominance metric: As the Minkowski-parameter increases, the distance between two points will come closer and closer to the largest coordinate difference, thereby outclassing all other coordinate axes. Therefore the metric with an infinite Minkowski-parameter is called a dominance metric.

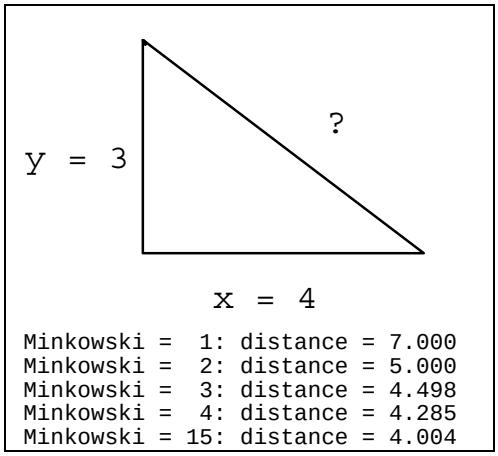

Figure 5: Distance dependent from Minkowski parameter.

Figure 5 shows two points with coordinate differences 4 and 3. It gives also the distances corresponding to several choices of the Minkowski-parameter.

MINISSA does not require that the stimuli can be represented perfectly in the given space. If it does not succeed completely, it tries to find a solution that is as good as possible. Therefore, it uses a loss- or stress function that indicates how well the order in the distances between the points corresponds to (the order in) the proximities. It starts from a provisional map and then moves the stimuli in small steps trying to minimize the stress function.

A more detailed description of the method is given in the following section.

2 MINISSA IN DETAIL

MINISSA operates on a symmetric matrix $\mathbf{X}_{nn}$ of proximities between stimuli $\mathbf{S}_1$ through $\mathbf{S}_n$. The program associates each stimulus $\mathbf{S}_i$ with a point $\mathbf{P}_i$ in r-dimensional space. The set of points (**P**) is called the *configuration*. From the configuration, a matrix **D** of distances between points can be derived. The purpose of MINISSA is to adjust the configuration **P** step by step such that at the end the order in the distances **D** corresponds (as well as possible) to the order in the data **X**.

With similarity data this means: $\mathbf{X}_{ij} \geq \mathbf{X}_{kl} \longleftrightarrow \mathbf{D}_{ij} \leq \mathbf{D}_{kl}$

and with dissimilarities it means: $\mathbf{X}_{ij} \geq \mathbf{X}_{kl} \longleftrightarrow \mathbf{D}_{ij} \geq \mathbf{D}_{kl}$

2.1 Fitting values and stress

In order to measure the quality of a solution, intermediate values are computed between the distances of the points (**D**) and the data (**X**). These intermediate values stay as close to the distances as possible, but they are not allowed to violate the order in the proximities. These intermediate values are called *disparities* or *fitting values*. Figure 6 shows a plot of the rank order of the dissimilarities (the x-axis) and the distances as well as the fitting values (both on the y-axis) in a hypothetical configuration. The distances are represented by the symbol | ; the fitting values are shown as dashes (-).. In a perfect solution, the distances must increase from left to right or at least stay on the same level. A plot like that in figure 6 is called a Shepard* plot.

The quality of a solution is measured from the differences between the distances and the fitting values according to the formula:

$$S = \sqrt{\frac{\sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(D_{ij} - F_{ij}\right)^2}{\sum_{i=2}^{n}\sum_{j=1}^{i-1}D_{ij}^2}}$$

where D is the matrix of distances and F is the matrix of corresponding fitting values. S is called the *stress*. For a perfect solution its value will be zero (and the plot will have asterisks only).

* After its inventor R.N. Shepard

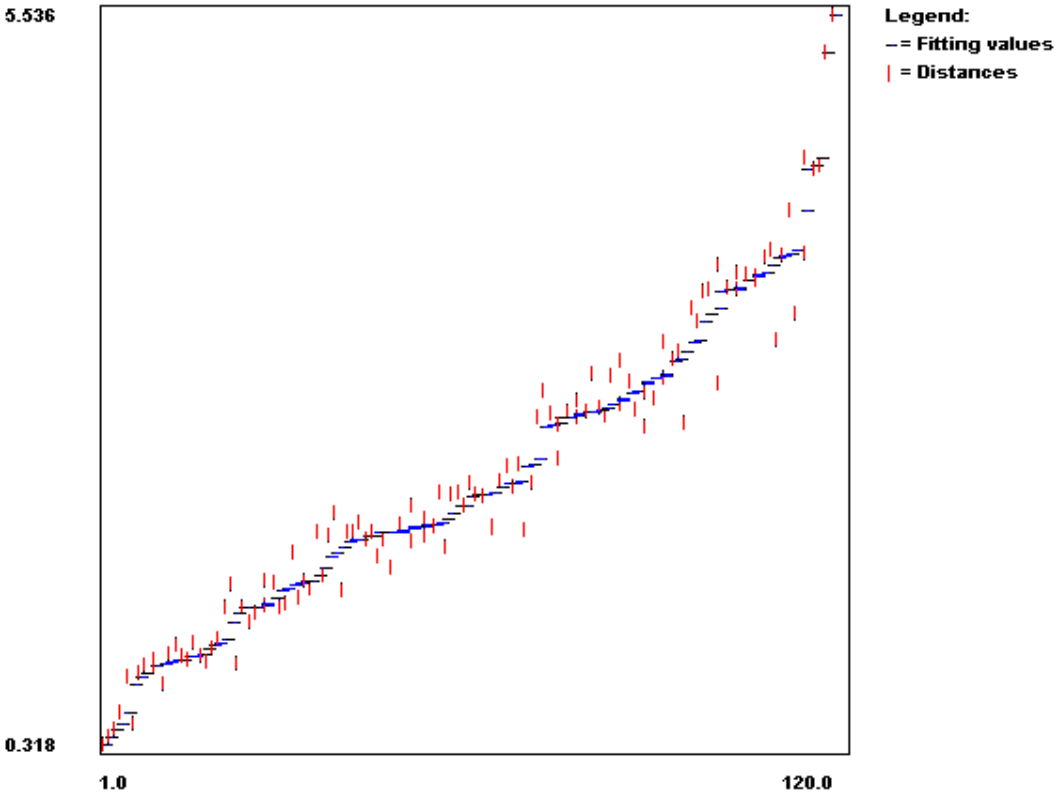

Figure 6: a Shepard plot (on the solution of figure 3).

There are two possible definitions of the fitting values:

d^ or DHat: the values are chosen such that the enumerator of the stress formula is minimal. They are called monotone regression values. These are the fitting values that are used in the final stage of the minimization process, since d^ is used in the function the program finally wants to minimize. The computation of d^ is described in detail in section 10.

d* or DStar: the values are obtained by sorting the distances and the data and matching them according to these order. They are called rank image values. These fitting values are used at the start of the minimization process.

Figure 7 shows the both types of fitting values with the corresponding dissimilarities and distances in the analysis of a small matrix, somewhere in the middle of the process.

pair		dissim.	distance	d^	d*
5	4	-89	0.6378	0.6378	0.6378
4	3	-86	0.7423	0.7311	0.7039
5	1	-82	0.7156	0.7311	0.7156
6	5	-82	0.7625	0.7311	0.7423
6	1	-80	0.7039	0.7311	0.7625
4	1	-79	1.1723	1.1723	1.1723
6	4	-69	1.3997	1.3902	1.3711
3	2	-64	1.4000	1.3902	1.3997
5	3	-63	1.3711	1.3902	1.4000
3	1	-58	1.7759	1.7627	1.7495
4	2	-54	1.7495	1.7627	1.7759
2	1	-50	2.0087	2.0601	2.0087
5	2	-50	2.1114	2.0601	2.1114
6	3	-39	2.1255	2.1255	2.1255
6	2	-80	2.6560	2.6560	2.6560

Figure 7: Dissimilarities, distances and fitting values in the analysis of a 6 by 6 matrix.

2.2 The treatment of ties

Ties in the data (groups of equal proximities) can be handled in two ways: they can be represented by different distances in the solution, such that the goodness of fit is optimized (the *primary approach*) or they can be represented by equal distances (the *secondary approach*). If the primary approach is used, we say that *ties may be broken*. The secondary approach will in general lead to a higher stress, since differences between distances for tied data will contribute to the stress.

If the primary approach is used, tied data are assumed to have the same order as their distances, which of course changes at every iteration. Consequently, there are never violations of monotonicity within a tie when this approach is used. In the program the primary approach is realized by sorting tied elements according to the order in their distances before the computation of the fitting values.

The secondary approach tries to give tied data equal distances. Therefore, fitting values within a tie are replaced by their average value.

2.3 Versions of stress

MINISSA will adjust the configuration step by step in order to decrease the value of S. The success of the algorithm can be expressed by several versions of the stress function. MINISSA computes the following results:

1. Stress DHat: the stress S with $\hat{d}$ as the fitting values.
2. Raw stress DHat: the numerator of S (above) with $\hat{d}$ as the fitting values.
3. Raw stress DStar: the numerator of S with d^* as the fitting values.
4. The coefficient of alienation K: It is very closely related to the total stress S based on d^*:

$$K = \sqrt{1 - \frac{\sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(D_{ij}F_{ij}\right)^2}{\sum_{i=2}^{n}\sum_{j=1}^{i-1}D_{ij}^2 \sum_{i=2}^{n}\sum_{j=1}^{i-1}F_{ij}^2}}$$

Typically the coefficient of alienation is about 1.4 times S (based on d^*).

2.4 The main algorithm

The main algorithm performs the following steps:

1. The data are replaced by rank numbers (and reversed in case of similarity data).
2. A trial configuration **P** is constructed (see 2.6) and the distances **D** are computed from it. The formula for the fitting values is set to d* since it works faster and has less chance to lead to a local minimum or a trivial solution (see 2.5).
3. (*Phase 2*:) The configuration is normalized such that the average of the coordinates is zero for each of the dimensions and the sum of squares of all coordinates is equal to the number of stimuli. The distances **D** are computed from the normalized configuration **P**. From the distances the fitting values d^* and the stress are computed.

4. (*Phase 1*:) The configuration is adjusted by moving each point a small step such that the stress probably decreases (using partial derivatives of the stress with respect to the configuration **P**). This adjustment is performed in a number of steps (max. 5 if d^* is used and only 1 if $d^\wedge$ is used). This iterative minimization of the stress is performed by use of the steepest descent method as described by Kruskal (1964a).

   If d^* is used for the fitting values, only the numerator of the stress formula is minimized in order to gain time (the method is called *soft squeeze*). If $d^\wedge$ is used the whole stress formula is minimized (the *hard squeeze* method).

5. Now a decision has to be made whether the process must go on and how. The decision depends on a number of criteria:
   - There is a minimum and a maximum number of iterations.
   - If the stress does not decrease any more, the formula for the fitting values is changed to $d^\wedge$; if that was already done before, the process must stop (continue with step 6).
   - After the change of stress formula, the process continues from the best configuration hitherto (possibly not the last one). Even if the stress increases, the process continues a few steps (never more than 4) to see if the increase is just temporary.
     If the process continues, the step size (see step 4) will be adjusted and the algorithm loops back to step 3.

6. The final configuration is normalized and if the metric is Euclidean, the configuration is rotated to its principal axis orientation. As a result, there will be no correlation between the positions on the different dimensions.

There is no objective rule to determine what is 'high' stress and what is not. As a rule of thumb, one may use the following hints for the primary approach:

$S < 0.01$ is excellent
$S < 0.05$ is good
$S < 0.10$ is fair
$S < 0.15$ is moderate
$S \geq 0.15$ is poor

When ties are not allowed to be split up (the secondary approach), the stress will normally be higher than when splitting is allowed. In addition, a solution in fewer dimensions will in general have a higher stress. If the solution in more dimensions has the higher stress, that solution must have been caught in a local minimum or the process may have stopped before convergence was reached (one might consider to increase the maximum number of iterations).

2.5 Trivial solutions

The requirement that the distances have the same order as the dissimilarities can be met very easily by making them all equal. The easiest way to do so is to give all stimuli the same position, but the stress formula counteracts a movement towards such a solution by its denominator. Nevertheless, it can happen that at least groups of points fall together or come very close to each other. In addition, MINISSA may construct groups of points with almost equal distances to each other. A configuration that suffers from such a disease is called a *trivial solution*. The Shepard plot for a trivial solution has a characteristic appearance, like a staircase consisting of a few large steps. There is no general way to prevent the program from trivialization. One may try another trial configuration or start from more dimensions.

2.6 The trial configuration

The user can define a trial configuration, but the program can also generate one automatically. Most times the generated configuration is quite good.

It may be helpful to start in a large number of dimensions and then to use the coordinates on the first dimensions of the solution as the start for the analysis in fewer dimensions. MINISSA offers the possibility to perform analyses in higher dimensionality just to generate a good trial configuration. If the Euclidean metric is used, the program will perform a principal axis rotation on its intermediate solutions and just thereafter drop its last dimension.

A start from higher dimensionality reduces also the chances of landing in a local minimum. Figure 8 shows how the stress changes if a range of dimensions is used in the analysis of the railroad data. The first row shows the stress history when the number of dimensions is reduced gradually from nine to two dimensions. The second row describes the stress when the algorithm starts from eight dimensions and so on. Clearly the solution that immediately starts in two dimensions is caught in a local minimum.

Number of dimensions:

9	8	7	6	5	4	3	2
0.001	0.002	0.006	0.004	0.005	0.010	0.028	0.057
	0.003	0.003	0.004	0.005	0.009	0.026	0.057
		0.003	0.004	0.005	0.009	0.025	0.057
			0.003	0.004	0.009	0.024	0.057
				0.006	0.010	0.024	0.057
					0.010	0.024	0.057
						0.023	0.057
							0.163

Figure 8: Final stress for several ranges of dimensions (on the data of figure 2).

If the trial configuration is generated automatically by the program, a quite sophisticated method is used. In a copy of the n x n dissimilarities matrix (call the copy C) the dissimilarities are replaced by their ranks: the smallest gets a 1 and the largest n(n-1)/2. These numbers are entered in the lower triangle but also in the upper part. Each off diagonal cell is divided by the highest rank: n(n-1)/2. Then each diagonal cell gets the sum of its row plus 1 and each off-diagonal cell is replaced by 1 minus its current value. Now the resulting matrix contains a kind of similarities, with the highest values on the diagonal. Then a principal components analysis is performed on this matrix C. The first component is discarded since all its elements will have the same value ($\sqrt{n}$). The second and following components will form the columns of the trial configuration!

It can be shown that this configuration maximizes the correlation between the distances and the ranks of the dissimilarities.

If for some reason the trial configuration is incomplete (some cells may be left empty or too few columns may be given), the program will try to fill the gaps:

1) It will generate its own trial configuration.
2) If a coordinate value is missing in the given configuration, MINISSA will look for the two stimuli that are closest to that with the missing value (according to its own configuration, see 1) and for which the corresponding coordinate values are known.
3) If two such stimuli are found it will estimate the missing value by interpolation between the two known coordinates. Otherwise, it will fill the gap by a small random value.

If a column of the trial configuration contains the same value in all its entries, the entries are also treated as missing values and filled with small random numbers.

3 Files

There are six file types that may be important to this program:

data files:
The files that contain the proximity data to be analyzed. One program run can analyze several files of this type, each containing a proximity matrix. The chosen options and data definitions will apply to all these files.

trial configuration file:
If the user specifies his own trial configuration, he can do so in three ways:

1. By specifying the trial configuration after the proximity matrix in the first (or only) data file to be analyzed. If there are several data files, this may a bit tricky, since the order in which the files are analyzed may be different for different operating systems.
2. By specifying the trial configuration in a separate file.
3. By specifying the trial configuration interactively during the first program phase, in which the options are set. If the settings are saved in a settings file (see the next paragraph), the trial configuration will be saved with them.

Even if there are several data files, there can be only one trial configuration.

settings files:
These files are used to save the options as they are specified by a user. A settings file contains all information about the analyses to be performed, including the description of the proximities and, possibly, the trial configuration, but not the proximities themselves. Many settings files may be available. By default, their extension is '`.setmin`'.

listing files:
A listing file contains the main results of an analysis in a nice layout for human readers. There will be one listing file for each data file. The name will borrow its first part from the data file and end on '`.LST`' (or on '`1.LST`', '`2.LST`', ... and so on). You can inspect it by any editor, but in order to have an orderly layout you must view it in a small non-proportional font like `Courier New 9`. Listing files are stored in the folder in which the corresponding input files are found.

raw output files:
Depending on the chosen options, the program may produce one file with the final configuration for each input file. These raw files are meant to be input to other programs, for instance as trial configurations for other MINISSA-runs. Their names will take their first part from the data files and end with '`.OUT`' (or '`1.OUT`', '`2.OUT`', ... and so on).

plot files:
MINISSA may produce two types of plots:

1. plots of the **final configuration**, that is the positions of the stimuli in their space. If the final solution contains more than two dimensions, the program will produce a separate bitmap file for the projections on the two-dimensional subspaces: a file for dimensions 1 and 2, one for dimensions 1 and 3 and so on. The names of these bitmap files will take their first part from the data files and, at least on Windows machines, they will end with '`P.BMP`', '`P1.BMP`', '`P2.BMP`' and so on (or '`2P.BMP`', '`2P1.BMP`', '`2P2.BMP`' and so on, if there are already files with the first mentioned names).
2. Shepard plots, one for each data file. See 2.1 for an explanation. The names of these files will take their first part from the data files and, on Windows machines, they will end with '`S.BMP`', '`S1.BMP`', '`S2.BMP`' and so on.

4 Installing MINISSA on Windows

The installation of the program is very simple:

1. Copy the file *Minissa.exe* to any place on your hard disk. Optionally you may make shortcuts on the task bar and/or on the desktop.

2. After the first time you have used the program, double click on a listing file. Windows will ask you to select the program to be used when opening the file. Then select a simple text editor like Notepad or WordPad.

3. After the first time you have saved the program settings, double click on the settings file. Windows will ask you to select the program to be used when opening the file. Select the program *Minissa.exe* or any shortcut to it.

That is all: from now on, you can start the program by double clicking the exe-file, one of its shortcuts or a settings file and you can open a listing file by double clicking it.

5 Running MINISSA

To run MINISSA you must double click on its executable file (for Windows: *Minissa.exe*) or, if you have used the program before, on one of its settings files (for instance *Current.setmin*). The first thing you will see then, is the main window, as shown in figure 9.

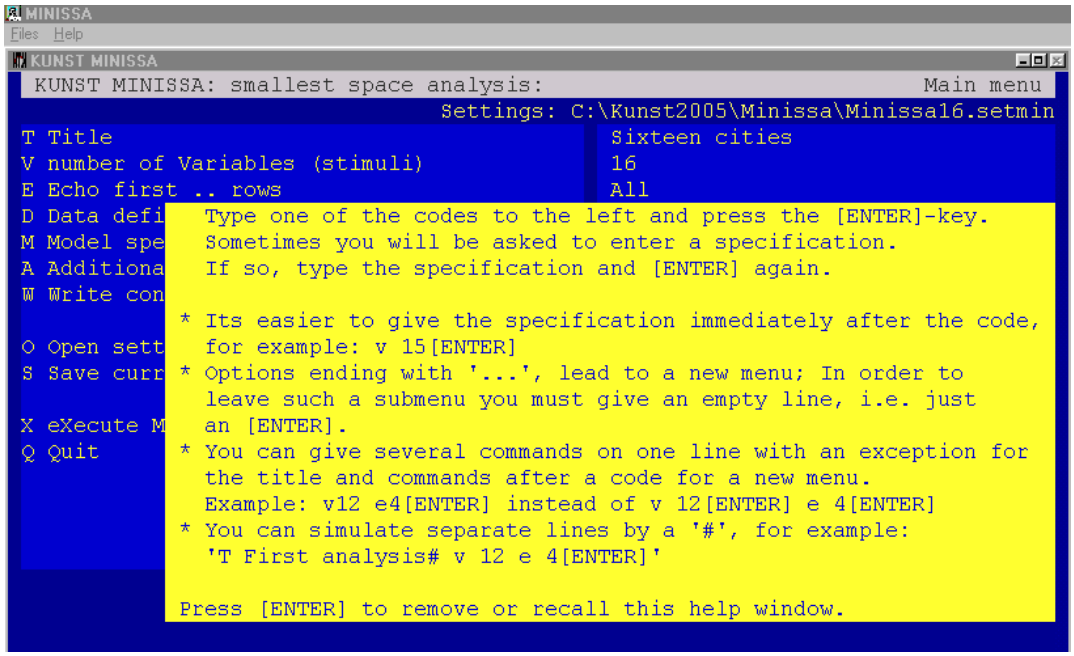

Figure 9: The main window.

On the screen the light part with the text '`Type one ...`' is a *yellow* text window. Yellow windows contain hints and explanations. If you have read the text in it (or do not need it), you can press the Enter-key and the yellow window will disappear (see figure 10).

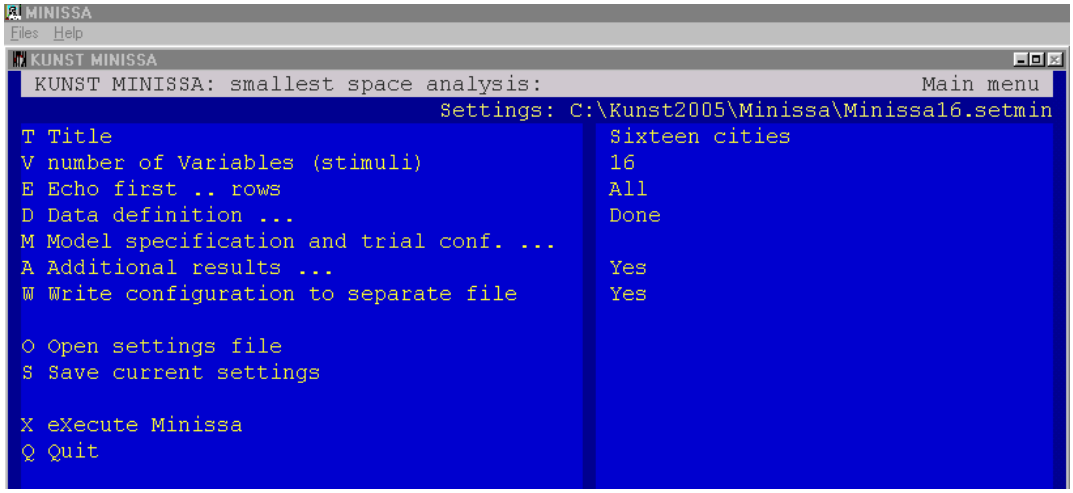

Figure 10: The main window without the help information.

Now you can see the entire main window. The left part is the main *menu*. It consists of a list of options, each preceded by a one-character code. To select an option, you must type the code, followed by the information you want to give and then the Enter-key. From now on, we will indicate the Enter-key as: Enter. You may for instance type: `T Analyzing railroad distances`Enter to define the title that will appear as a header in the listing files.

If you do not know what the meaning of an option is, you may just enter its code and Enter. There will appear a question on the screen and, if helpful, a yellow window with additional information. Press Enter to remove the yellow window and then give the specification belonging to the code, or ignore the yellow window and give the specification at once. **Do not repeat the one-character code itself!**

The right part of the window gives a short review of the options, as they are currently set. In the example of figure 10, you can see the following:

- The title is '`Sixteen cities`'.
- There are 16 stimuli.
- All input rows will be echoed in the listing file.
- The required data definitions are specified.
- Some additional results will be shown in the listing file.
- The final configuration(s) will be written to raw output file(s)

If the text of a menu item ends with three dots, like '`D Data definition ...`' in figure 10, the option leads to a submenu, i.e. a new menu with new options. To return from such a submenu you must simply press Enter without any text. If the screen shows a yellow help-window, you will have to type two Enters: one to remove the yellow window and one to leave the submenu.

6 Menu options

6.1 The main menu

The *main* menu (see figure 10) contains the following options:

6.1.1 T Title

This option allows you to specify a header to be used in the listing files.

6.1.2 V Number of variables (stimuli)

By this option you must specify the number of stimuli. There is a system dependent maximum to this number. For the current windows version that maximum is 500.

6.1.3 E Echo first .. rows

By typing `e ##Enter` you specify that the first ## rows of each input data matrix and of the trial configuration, if read from a file, must be shown in the listing file. This may help you to check if the input specifications and the inputs match correctly. The default value for this option is 2.

6.1.4 D Data definition ...

The option `d` leads to the *data definition* menu, where you must define the main input containing the proximities. This menu will be discussed further in 6.2.

6.1.5 M Model specifications and trial conf.

The option M leads to the menu with model specifications. It will be treated in 6.3.

6.1.6 A Additional results ...

This option leads to a new menu that allows you to order additional information in the listing file. It will be treated in 6.4.

6.1.7 W Write configuration to separate file ...

By this option you can specify that the final configuration must be stored in a “raw” text file. It will be discussed in 7.2.

6.1.8 O Open settings file

If you ever have saved the options for MINISSA or if you received a settings file from someone else, you can retrieve the options from the settings file. If you type `OEnter`, a file-selector box will appear on the screen that allows you to select the settings file. Figure 11 shows such a file selector box.

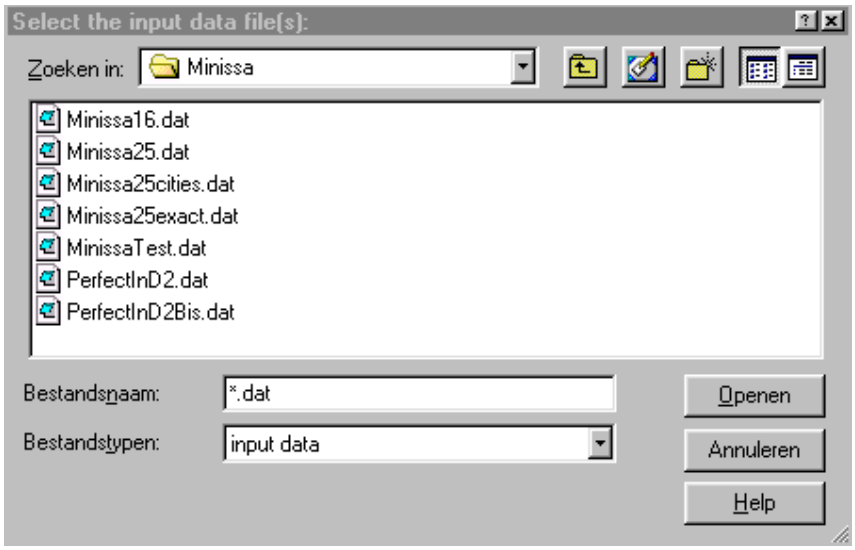

Figure 11: Retrieving information from a settings file.

By default settings files from MINISSA have the extension '`.setmin`'. It may be handy to save your current settings before collecting new ones. The program may remind you of that. After you have collected information from a settings file, its name will be visible on the upper right part of the main window.

6.1.9 S Save current settings

If you want to save the options and specifications that you have made so far, you can enter `S`Enter. If you do so a file-selector box will appear that allows you to specify the place and the name of the file to which the settings must be written (see figure 12).

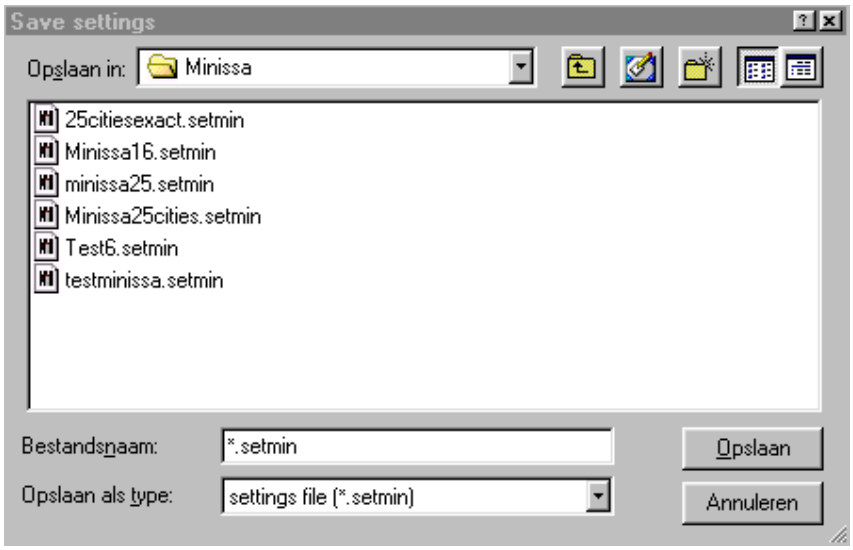

Figure 12: Saving information to a settings file.

6.1.10 X eXecute Minissa

If you have specified all options, you can type `X`Enter to start the computations. The program will check if all obliged options are specified and if there are no inconsistencies. If everything is right, the computations will start. If the program has been correctly installed and runs without problems, it will, when it is finished, automatically open the last (or only) listing file it has produced. If it fails to do so, you can open it yourself by any text editor like WordPad, Notepad or Word. In order to have a nicely outlined text, you must select a small non-proportional font, for instance `Courier New 9`.

6.1.11 Q Quit

The option `Q` is a kind of emergency exit. If you choose it, MINISSA will halt without performing any calculations and without producing any output files.

6.2 Definition of the data

From the *main* menu, you may type the option `D`Enter to enter the *data definitions* menu. After you have filled out the options in this menu you must press Enter to return to the *main* menu. If there is a yellow window on the screen, you must type two Enters: one to remove the yellow window and one to return to the *main* menu.

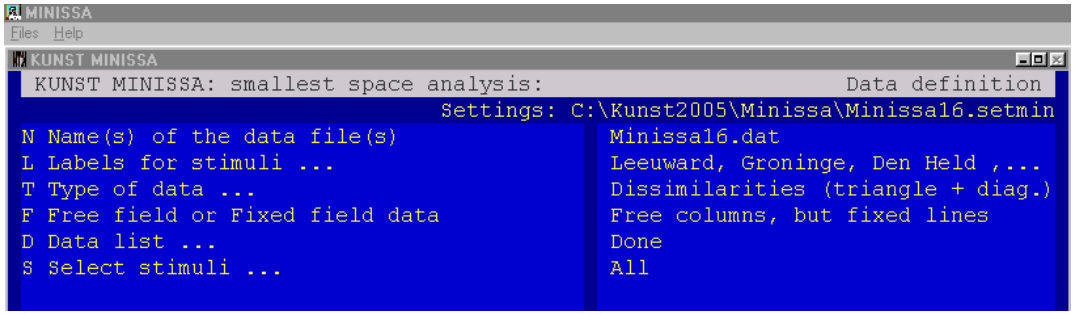

Figure 13: The *data definition* menu.

In the *data definition* menu (see figure 13) the following options can be chosen:

6.2.1 N Name(s) of the data file(s)

If you choose this option, a file-selector box will appear that enables you to select one or more data files. Figure 14 contains an example.

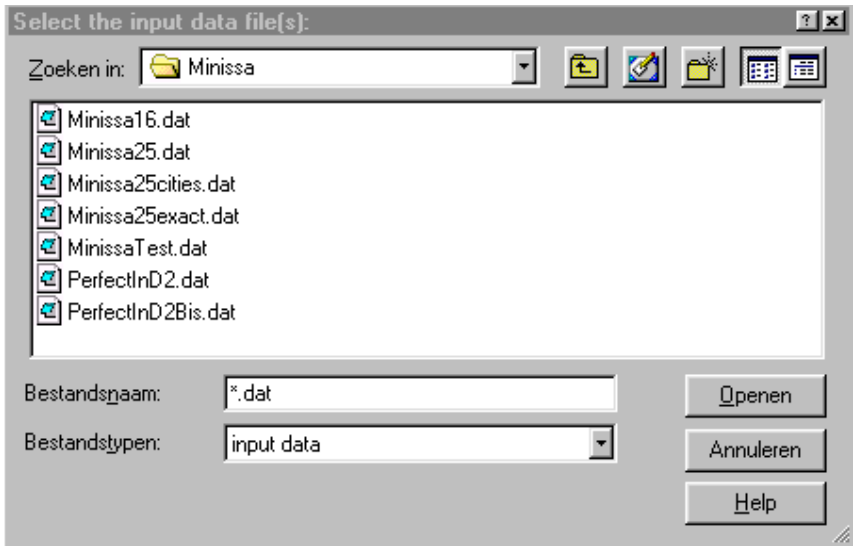

Figure 14: Choosing one or more input data files.

Each of the files may contain one matrix with proximities. One, the first to be analyzed, may contain a trial configuration after the proximity matrix. Note that all proximity matrices must have the same type, size and format, and that they will be analyzed according to the same options.

6.2.2 L Labels for stimuli ...

The option **L** can be used to specify names for the stimuli. If you type `L`Enter
a new window appears, like the one in figure 15.

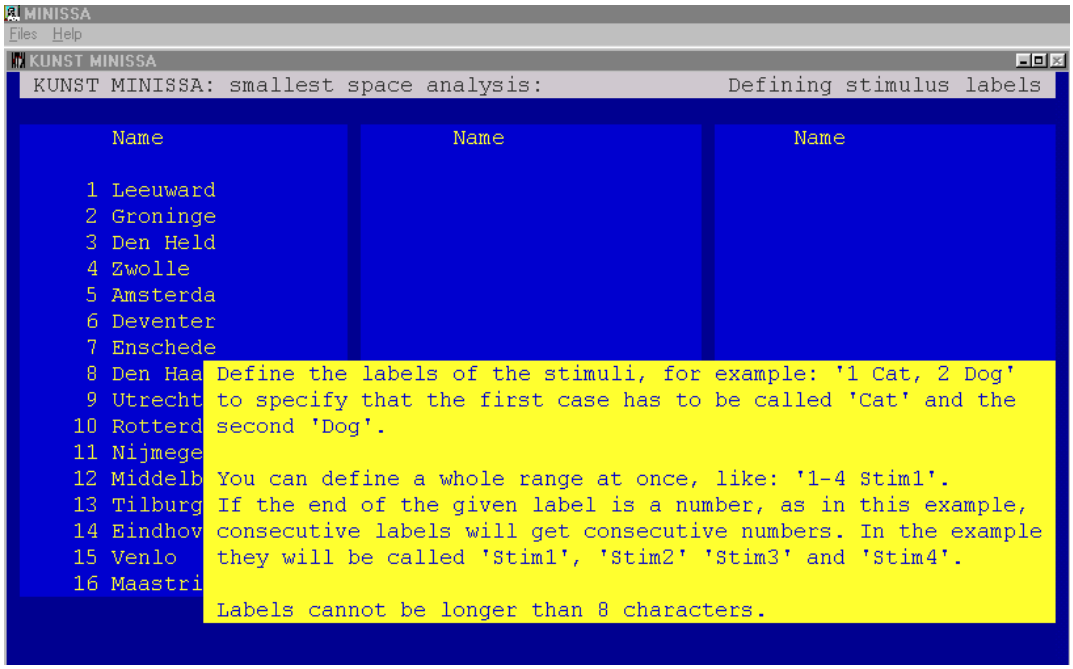

Figure 15: Specifying names for the stimuli.

Now you can type the sequence number of a stimulus and its name. The name will be truncated to its first eight characters. If the name contains spaces or commas or if it starts with a dash (a minus sign) you must put it between quotes. If such a quoted name contains a quote itself, that quote must be replaced by two adjacent quotes.
You may also specify a range of sequence numbers followed by a name: all stimuli in the range will receive the same name, unless the name ends with a number; in that case they will receive increasing numbers. Leave this window by an empty line (just `Enter`).

Examples: `1 Leeuwarden 2 Groningen 3 'Den Helder'Enter`
`1-16 City1Enter`

In the second example the stimuli receive the names `City1`, `City2`, `City3`, ... and so on.

6.2.3 T Type of data ...

If you have chosen the option **T** in the *data definition* menu, the *type of data* menu appears (see figure 16).

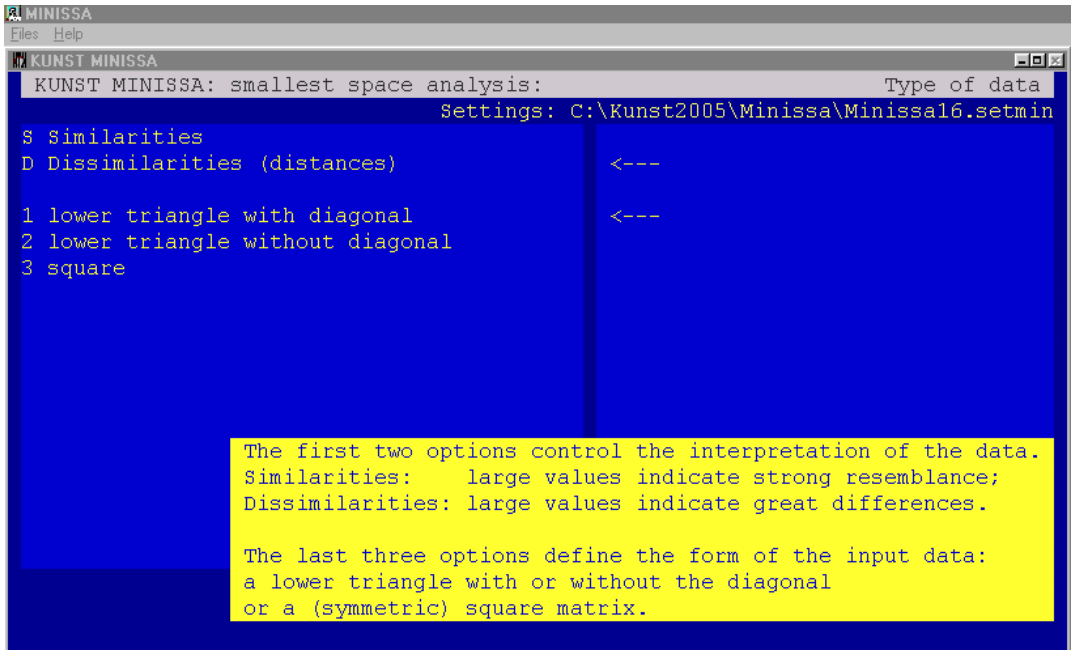

Figure 16: The *type of data* menu.

The *type of data* menu contains two groups of options. Within each group the options exclude each other. The first group contains the options **S** and **D**. They define if the input proximities are to be interpreted as **s**imilarities or as **d**issimilarities. If the data are similarities (so S is chosen), MINISSA will change the sign of all input proximities and thereby transform them into dissimilarities.

With the second group of options, you can describe the general form of the input matrix. Since the proximity matrix is supposed to be symmetric and the diagonal elements contain no useful information, MINISSA needs only half of the matrix without the diagonal. Figure 17 shows the different forms for a small data matrix of 4 stimuli.

```
 0                13                0 13 72 24
13  0             72 75            13  0 75 26
72 75  0          24 26 48         72 75  0 48
24 26 48  0                        24 26 48  0

lower with      lower without   square
diagonal        diagonal
```

Figure 17: The three possible forms of the proximity matrix.

If a square matrix is given, MINISSA will take its information from both parts and take for each pair of stimuli i and j the mean of the two proximities d_{ij} and d_{ji}.

6.2.4 Free field or Fixed field data

The positions of the stimuli within a row of the proximity matrix can be organized in three different ways.

Free field format with fixed line numbers:

For each proximity within the row, the exact line number is specified. The values within the line are not bound to exact positions, but they must be separated by spaces, tabs or commas. Each line must contain enough values to fill all proximities on that line.
The list of stimuli must be specified such that the line numbers are in ascending order.
If not all values are on a single line, you must use a *data list* to specify the lines (see 6.2.5.1).

Completely free format:

With completely free format neither line numbers nor positions within lines are specified. For each row of the matrix, the program will read just the number of values it needs. These values must be separated by spaces, commas, tabs or line ends. The number of lines per row may vary. The only restriction is, that each row must start from a new line. In order to choose this format you must activate the *datalist* and set the line number of the first stimulus to zero. See 6.2.5.2.

Fixed format:

For each proximity within a row the exact line number and position is specified. If a fixed format is chosen, the positions of the proximities must be specified in a *data list*,. See 6.2.5.3.

In general, the use of free format with fixed lines is highly recommended since it is less sensitive to irregularities in the data and errors in the specifications. Moreover, if each row takes only one line you do not need to fill out the data list.

However, in two situations a fixed format is unavoidable:

- If the data contain values that should be skipped.
- If not all values in the data are separated by spaces, commas or tabs.

6.2.5 D Data list for the proximity matrix

A data list may be needed to specify the input positions of the proximities in each row of their matrix. It can also be used to define names for the stimuli or to delete some stimuli from the list. The exact appearance of the data list depends on the format type (see 6.2.4).

6.2.5.1 Free format with fixed lines

Figure 18 shows an example of a data list for a free format with fixed lines.

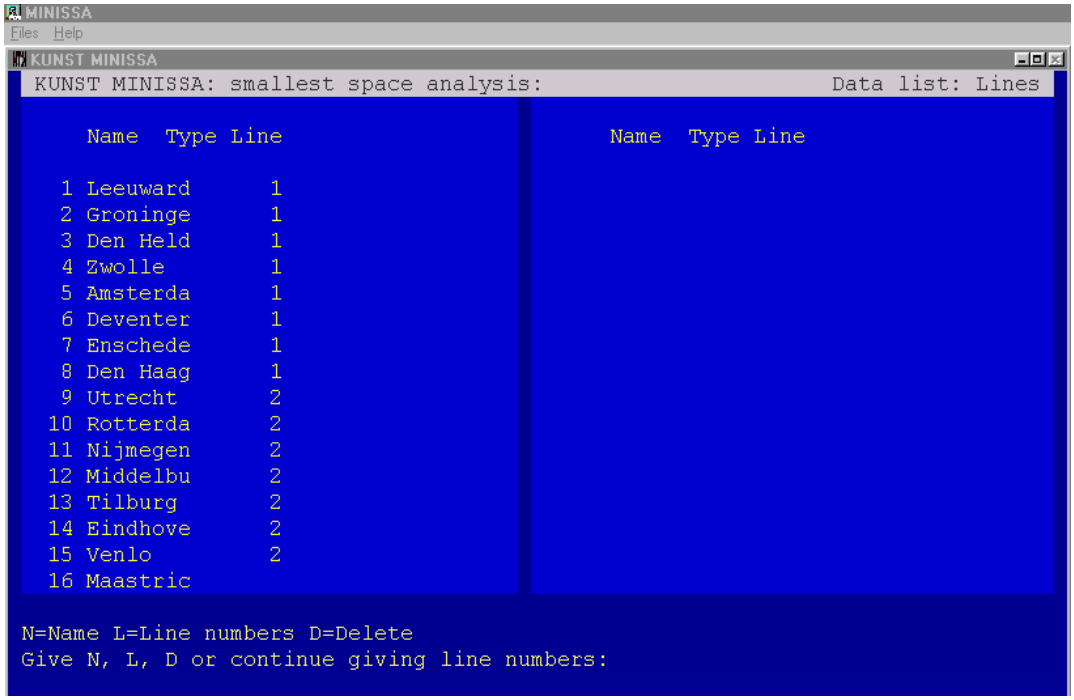

Figure 18: Example of a data list for free format with fixed lines.

If a free format with fixed line numbers is used and there is more than one line per row, you must specify the line numbers. To specify these line numbers you must type the code '**L**' and then the sequence number of the variable and its line number. You may also type a range of sequence numbers.

Example: `L 9-14 2Enter`

This specification changes the line numbers for the stimuli 9 through 14 to 2. If there are more line numbers to be set, the code (`L`) needs not to be repeated. The line numbers must be in ascending order. Consequently, in the example, the program will automatically change the line numbers of stimuli 15 and 16 to 2, if they were still 1.

If the input matrix has a triangular form, these specifications must describe the last row, since that contains the largest number of values.

If the input matrix is triangular, and the format specifies more than one line per row, the program assumes that for a row only those lines are present, that contain values for the lower triangle.

As an example:
Suppose there are 16 stimuli, the form of the matrix is a triangle without diagonal and line numbers are defined as in figure 18. Then the input file must look like figure 19. Note that no line number is specified for stimulus 16; since the input data do not contain diagonal values, the longest row contains only proximities for the first 15 stimuli!

```
 54
287 298
 95 106 192
206 217  81 111
126 137 184  31 103
162 173 245  67 164  61
242 253 121 147  63 139 200
182 193 120  87  39  79 140  60
238 249 144 143  86 135 196  23
 56
190 201 197  95 116  64 101 137
 77 133
365 376 271 270 213 243 280 150
183 127 179
252 263 190 157 109 129 166  94
 70  71  65 114
262 273 200 167 119 139 176 131
 80 108  75 151  37
251 262 252 156 171 125 162 183
132 160  61 203  89  52
320 331 298 225 217 194 231 229
178 206 130 249 135  98  69
```

Figure 19: A proximities file with two lines per row.

For square matrices, the *total* number of lines per row will automatically be adjusted to the last specified line number, but if the actual number of lines is larger than that, you can specify it explicitly by the code **T**:

Example: `T 4Enter`

This specification informs the program that each row will occupy 4 lines.
This option will not be available if the input matrix has a triangular form.

From the data list, you can also define names for the stimuli. These are the same names that can de specified by the option `L` in the *data* menu (see 6.2.2).

In order to specify the name of a stimulus you must type the code **N**, followed by the sequence number of the stimulus and its name. The name will be truncated to its first eight characters. If a name contains spaces or commas or if it starts with a dash (a minus sign) you must put it between quotes. If such a quoted name contains a quote itself, that quote must be replaced by two adjacent quotes.
You may also specify a range of sequence numbers: all stimuli in the range will receive the same value, unless the name ends with a number; in that case they will receive increasing numbers.

Examples:
```
N 1 Leeuwarden 2 Groningen 3 'Den Helder'Enter
  4-16 City4Enter
```

As you can see from the examples the code character `N` may be omitted the second and following times. In the first example, the name for the third stimulus is too long, so it will be abbreviated to `Den Held`.

The code **D** can be used to delete stimuli from the list. Type the character `D`, followed by a stimulus number or a range of numbers.

Examples:
```
D 1Enter
D 1-3Enter
```

The number of stimuli (in the *main* menu) and any selection will be adjusted automatically.

Note that this option does not modify the input data but their description.

6.2.5.2 Completely free format

To specify a completely free format you must activate the datalist and set the line number for the first stimulus to zero. For details see 6.2.5.1.

6.2.5.3 Fixed format

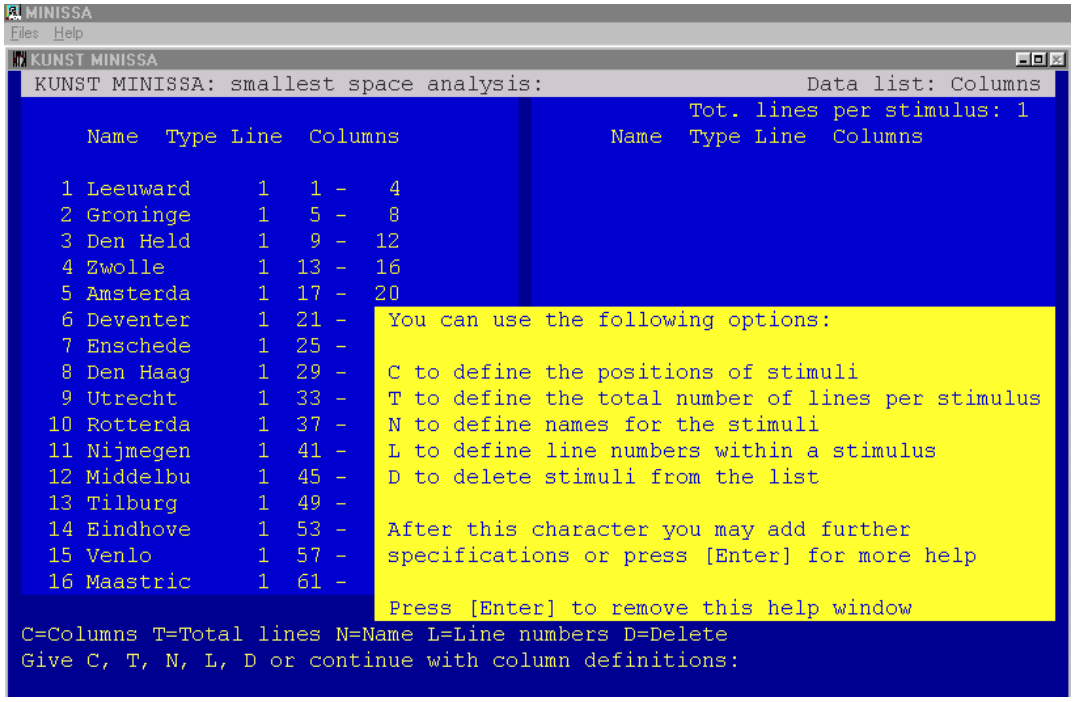

Figure 20: Example of a data list for fixed format.

Figure 20 shows an example of a data list for a fixed format.

With this format type, the exact line number and position are specified for each proximity in a row.

For each stimulus the exact line number and position are specified. According to figure 20, the proximity for the first stimulus (called `Leeuward`) must be taken from positions 1-4 in the first line. The second value (`Groninge`) from positions 5-8 in the first line and so on.

To specify the **positions** for the stimuli you must use the code **`C`**. The use will become clear from the following list of possibilities:

- One stimulus in one column: specify just the stimulus number and the position:
  `C 10 19`Enter
  {stimulus 10 is in position 19}.

- One stimulus in more positions: specify the stimulus number and the range of positions:
  `C 16 40-45Enter`
  {stimulus 16 is in positions 40-45}.
- More stimuli in consecutive columns: specify the range of stimulus numbers and the first position:
  `C 1-10 10Enter`
  {stimuli 1 to 10 are in positions 10-19, each occupying one position}.
- More stimuli in consecutive columns with more than one position per stimulus: specify the range of stimulus numbers and the range of positions. The number of positions must be a multiple of the number of stimuli in the command:
  `C 1-16 1-64Enter`
  {stimuli 1 to 16 are in positions 1-64, each occupying four positions: see figure 20}.

After you have specified the positions for a stimulus or a range of stimuli, you may continue defining positions without repeating the character `C` itself.

To specify line numbers you must type the code **`L`** and then the sequence number of the variable and its line number. You may also type a range of consecutive numbers.
Examples: `L 8-14 2Enter`
`L 15 3Enter`
These specifications change the line numbers for the stimuli 8 through 14 to 2 and the line numbers for stimulus 15 to 3. The code (`L`) needs not to be repeated. The line numbers must be in ascending order. Consequently, the program will automatically change the line number of stimulus 16 to 3.

After you have specified the line numbers for a stimulus or a range of stimuli, you may continue defining line numbers without repeating the character `L` itself.

The total number of lines for the last row will automatically be adjusted to the last specified line number, but if the actual number of lines is larger than that, you can specify it explicitly by the code **`T`**:

Example: `T 4Enter`

This specification informs the program that each row will occupy 4 lines. This option is only available if the input data for a square matrix.

In order to specify the name of a stimulus you must type the code **`N`**, followed by the sequence number of the stimulus and its name. The name will be truncated to its first eight characters. If a name contains spaces or commas or if it starts with a dash (a minus sign) you must put it between quotes. If such a quoted name contains a quote itself, that quote must be replaced by two adjacent quotes.
You may also specify a range of sequence numbers: all stimuli in the range will receive the same value, unless the name ends with a number; in that case they will receive increasing numbers.

Examples: `N 1 Leeuwarden 2 Groningen 3 'Den Helder'Enter`
`4-16 City4Enter`

As you can see from the examples the code character `N` may be omitted the second and following times. In the example, the name for the third stimulus is too long, so it will be abbreviated to `Den Held`. The stimuli after the third will be named `City4`, `City5` and so on.

The option **`D`** can be used to delete stimuli from the list. Type the character `D`, followed by a stimulus number or a range of numbers.

Examples: `D 1`Enter
`D 1-3`Enter

The number of stimuli (in the *main* menu) and any selection will be adjusted automatically.

Note that this option does not modify the input data but their description.

6.2.6 S Select stimuli

In MINISSA, you can select stimuli from the input matrix. Only the selected stimuli will take part in the analysis. The option allows you to skip some rows and columns of the proximity matrix without adjusting the input matrix. Moreover, since names and positions of stimuli apply to the whole matrix, the sequence numbers and names will not change by the selection and stay comparable between analyses with different selections. If the trial configuration comes from a file, it still must contain values for the deselected stimuli. The stimuli that are dropped by selection will also stay in the raw output file: their (unknown) coordinates are replaced by the value 99.0.

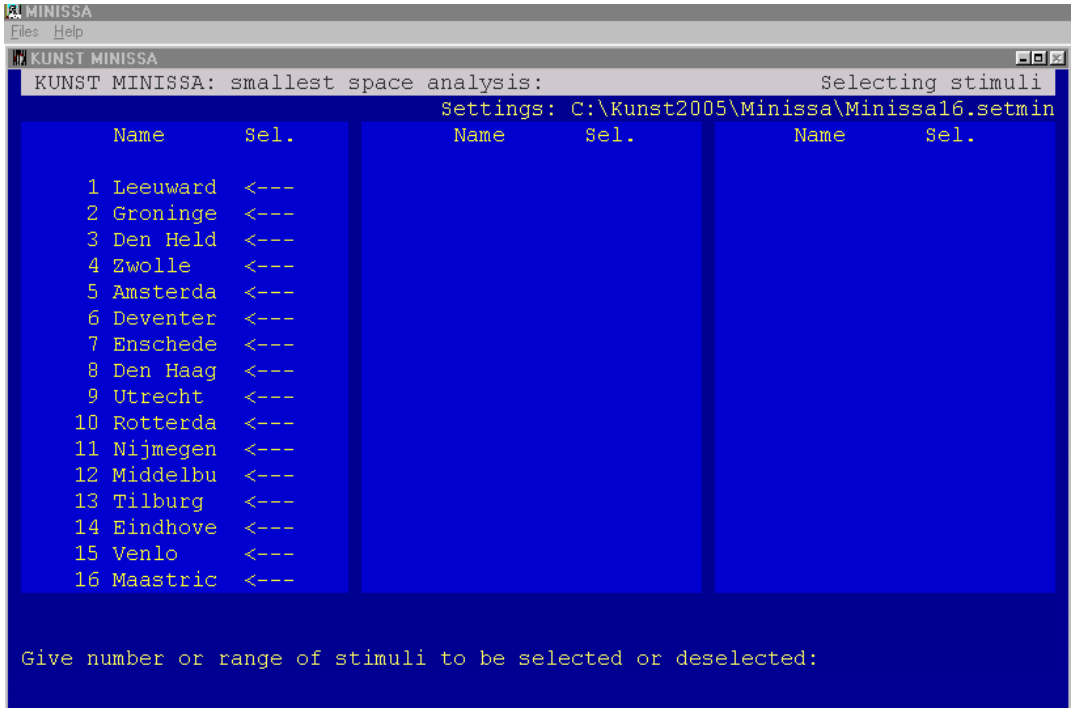

Figure 21: Selecting stimuli.

If in the *data definition* menu the option **S** is chosen, the screen will show a list of all stimuli, as in figure 21. By giving a stimulus number or a range of numbers, one can de-select or select stimuli. Stimuli that are to be included in the analysis are marked by an arrow (<--), stimuli that are not to be included are not marked. If one specifies a range of stimulus numbers, like 5-7, not all the stimuli in the range will switch: the first is switched, the others copy its new status: if the first stimulus of the range had an arrow, all stimuli in the range will end without arrows; if the first had no arrow, all stimuli in the range will get one.

6.3 Model specifications and trial configuration

If in the *main* menu the option **M** is chosen, the *model* menu shows up. It allows you to set some parameters that influence the computations to be performed. Figure 22 contains an example.

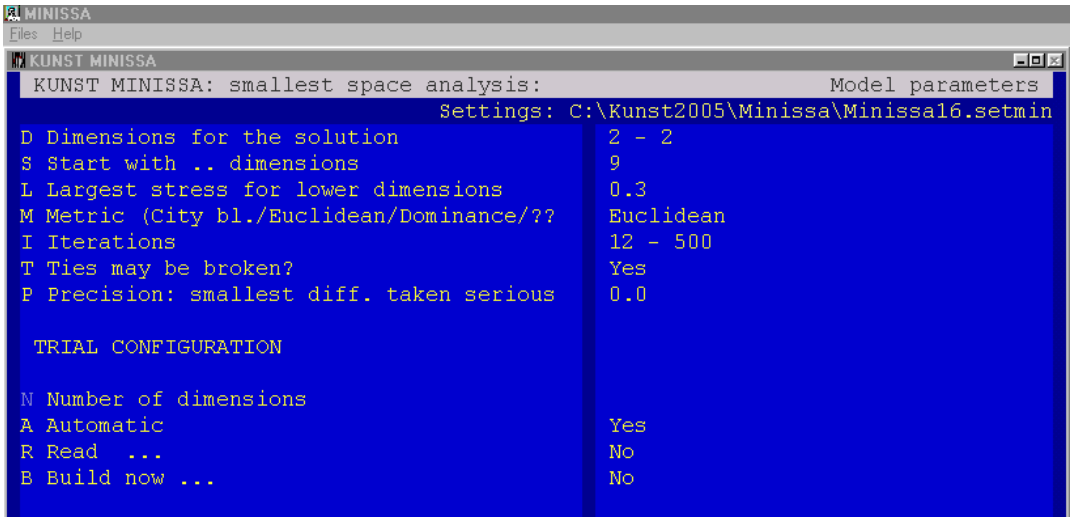

Figure 22: The *model* menu.

With the option **D** one defines the minimum and maximum number of dimensions in which solutions have to be sought. The number of dimensions must be in the range 1 through 9. MINISSA will start the analysis with the maximum number of dimensions. When the solution is found, it will drop one dimension and perform a new analysis, starting with the positions on the other dimensions as its trial configuration. The reduction of dimensions will however be interrupted if the stress of the last found solution is too high. The criterion for this decision can be set by the user (the option L).

If the Euclidean metric is applied, the solutions will be rotated to their principal axes, so that a minimum of information is lost when the last dimension is dropped.

Even if one expects a solution in a few dimensions, it may be helpful to start with more, because that may lead to a better trial configuration for the lower dimensional solutions! In that case, one must use the option **S** to specify the number of dimensions from which MINISSA must start. The program will then perform analyses in higher dimensions with a minimum of output in the listing file and only start reporting results extensively when the maximum of the actual range (according to option D) is reached.

This process of stepping down stops if the minimum number of dimensions is reached but also if the stress of the solution becomes too large. It will never stop for this reason when it is still preparing the trial configuration, that is when the number of dimensions is greater than the maximum according to option `D`. What is 'too large' is defined by the option **L**: if the stress exceeds the value given there, no solutions in fewer dimensions will be looked for.

The option **M** offers the opportunity to define the metric (the distance formula) to be used. The given value will act as the Minkowski-parameter (see 1). One may give any positive number less than or equal to 15, but strange things may happen with a parameter below 1. The Minkowski parameter is 2 by default, thereby defining the ordinary Euclidean metric. Instead of a 1 one may choose `C` in order to choose the city block metric, instead of a 2 `E` may be typed to define the Euclidean metric and with `D` one asks for the dominance metric.

MINISSA stops its iteration process if the configuration does not improve any more (see 2.4). But the user may put some restrictions on this mechanism by defining a minimum and a maximum number of iterations to be performed. By default the minimum is 12 (don't give up too soon) and the maximum is 200. One may change the values by the option **I**. Because the process consists of two phases, one using D* and one using D^ as fitting values, the iterations are evenly distributed among them: for both phases the minimum and the maximum are set to half the limits that are given here.

The option **T** determines whether the program must try to represent equal dissimilarities in the data by equal distances or not. If ties are allowed to be broken, the stress may be somewhat less than if they are not.

With the option **P,** one can define when two input proximities are to be considered to be tied and when not. Values are tied if their difference is smaller than the value given by the option **P**. The default value for this criterion is 0.

Their may be some ambiguity in this definition of ties: if the proximities are ordered there may be a chain of small differences, each smaller than the criterion. In that case, the program scans the proximities from small to large. The first proximity starts a tie. All consequent values that differ less than the criterion from the starting proximity are added to the tie. The first proximity that differs too much from the starting element will act as the start of a new tie and so on.

With the option **N** one may define how many dimensions the trial configuration contains. It has no meaning as long as the trial configuration must be constructed automatically. However, the trial configuration can also be taken from a file or defined within the menu system itself. Reasons to do so may be:

- You may be disappointed by the solution from a previous MINISSA run and want to guide the program in a certain direction.
- There may be theoretical grounds to expect some type of configuration to be the best.
- If several data matrices are analyzed you may want to use the same trial configuration for all of them.
- The final solution from one analysis may be an interesting starting configuration for the next, for instance when the two analyses pertain to two comparable groups of respondents (or to the same group under different conditions).

In such cases the trial configuration or at least a part of it must be given by the user and the option N must be used to define the number of dimensions in it.

The option **R** is used to indicate that the trial configuration must be read from a file. The option leads to new submenu (see 6.3.1). If the trial configuration and the proximities have to be read from the same file, the trial configuration must be placed after the proximities.

With the option **B** one can define the trial configuration immediately. The corresponding menu is explained in 6.3.3.

The options A, R and B are mutually exclusive. If one of them is chosen, the others are cleared and the only way to clear one is to choose another.

When R or B is chosen, the resulting trial configuration may contain too few dimensions. It may even contain 'holes'. MINISSA will fill these missing values as well as possible, but the resulting configuration may be far from optimal.

6.3.1 Reading the trial configuration

If you choose the option R in the *model* menu, a submenu emerges, as shown in figure 23.

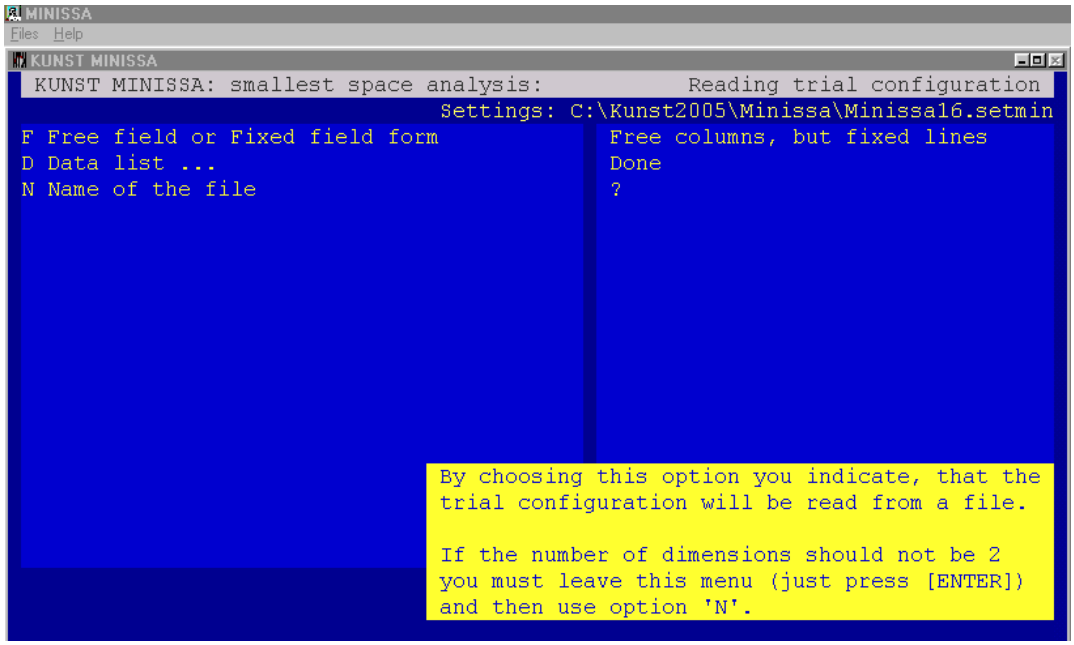

Figure 23: The menu for reading the trial configuration.

In this menu the option **N** can be used to give the name of the file that contains the trial configuration. If you choose this option, a file-selector box will appear like that in figure 24.

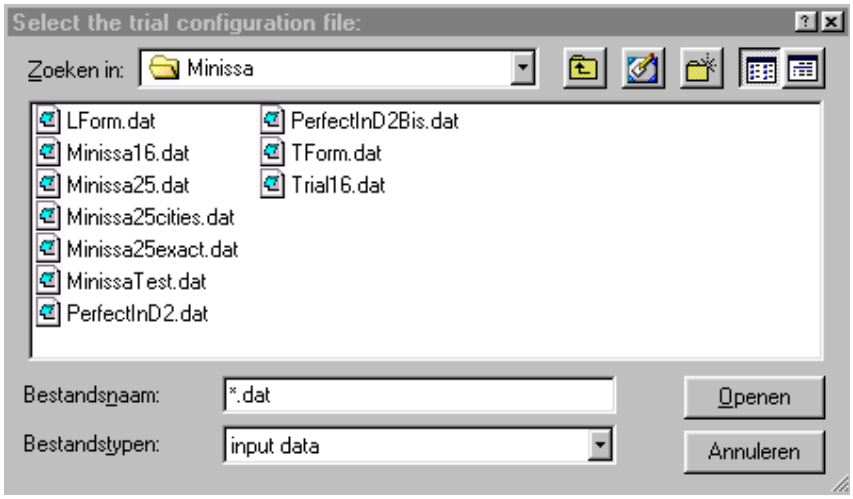

Figure 24: Specifying the file with the trial configuration.

The option **F** defines the format type of the trial configuration file. The positions of the dimension scores within a row of the configuration can be organized in three different ways.

Free format with fixed lines:

For each dimension within the row, the exact line number is specified. The coordinates within the line are not bound to exact positions, but they must be separated by spaces, tabs or commas. Each line must contain enough values to fill all dimensions for that line.

If a row of the configuration contains more than one line, the line numbers must be in ascending order. To define the line numbers you must use a *data list* (see 6.3.2.1).

Completely free format:

With a completely free format neither line numbers nor positions within lines are specified. For each row of the trial configuration, the program will read just the number of coordinates it needs. These values must be separated by spaces, commas, tabs or line ends. The number of lines per row may vary. The only restriction is, that each row must start from a new line. In order to choose this format you must activate the *datalist* and set the line number of the first stimulus to zero. See 6.3.2.2.

Fixed format:

For each coordinate value within a row the exact line number and position is specified. If this format is chosen, the positions of the coordinate values must be specified in a *data list*,. See 6.3.2.3.

In general, the use of free format with fixed lines is highly recommended since it is less sensitive to irregularities in the data and errors in the specifications. Moreover, if each row takes only one line you do not need to fill out the data list.

However, in two situations a fixed format is unavoidable:

- If the file contains values that should be skipped.
- If not all values in any row are separated by spaces, commas or tabs.

The option **N** is needed to specify the file that contains the trial configuration. If you choose this option, a file-selector box will appear that enables you to select the file. Figure 25 gives an example. If the trial configuration and the proximities have to be read from the same file, the trial configuration must be placed after the proximities.

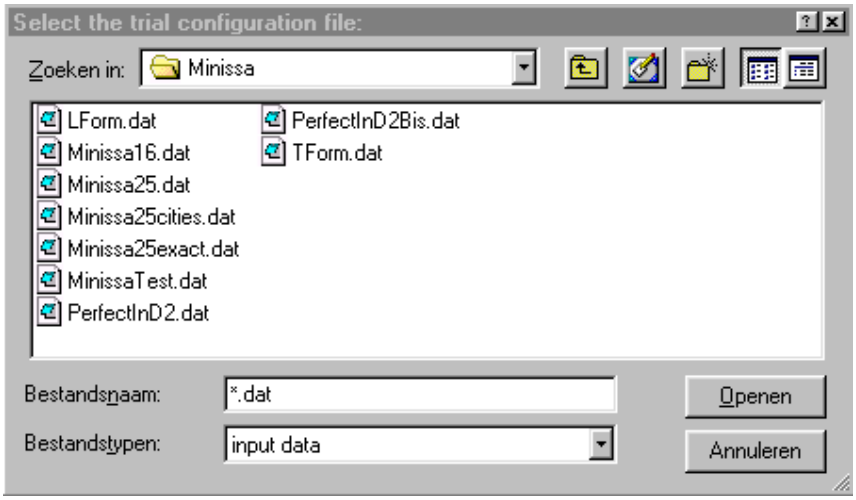

Figure 25: Choosing a file for the trial configuration.

6.3.2 D Data list for the trial configuration

A data list may be needed to specify the input positions of the coordinates in each row of the trial configuration. The data list can also be used to define names for the dimensions or to delete some dimensions from the list. The exact appearance of the data list depends on the format type (see 6.3.1).

6.3.2.1 Free format with fixed lines

Figure 26 shows an example of a data list for a free format with fixed lines.

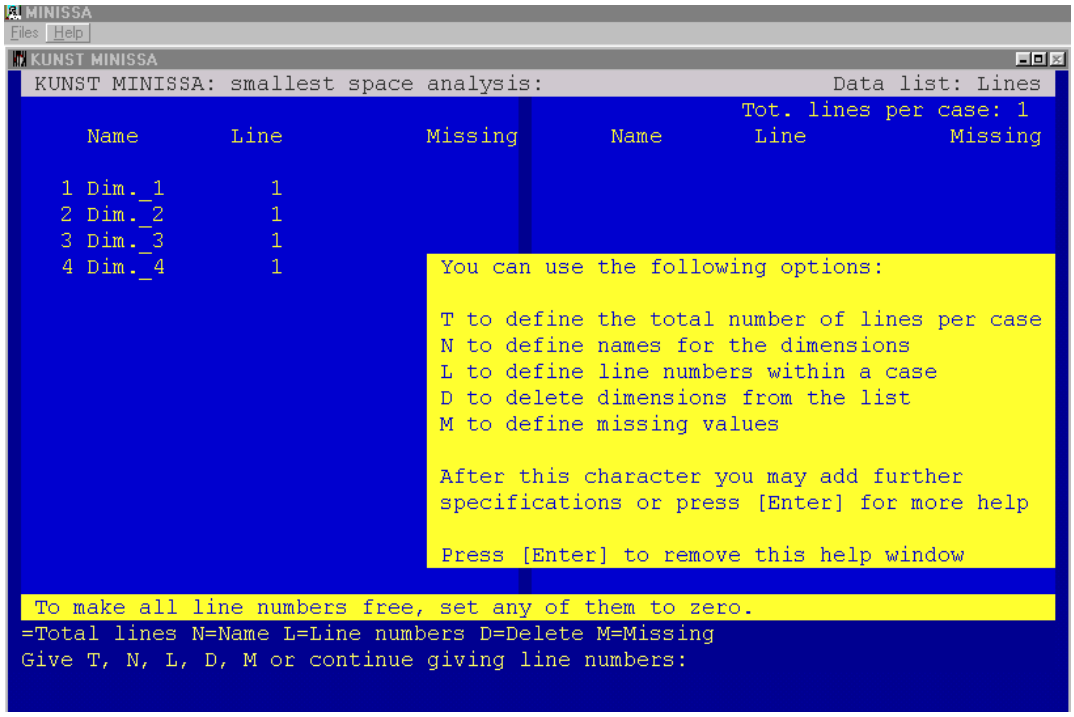

Figure 26: Example of a data list for a trial configuration in free format with fixed lines.

If a free format with line numbers is used and there is more than one line per row, you must specify the line numbers. Type the code '**L**' and then the sequence number of the variable and its line number. You may also type a range of sequence numbers.

Examples: `L 2 2`Enter
`L 3-4 2`Enter

These specifications change the line number for the second through the fourth dimension to 2. The code (`L`) needs not to be repeated for consecutive specifications of this type. The line numbers must be in ascending order. Consequently, in the first example, the program will automatically change already all line numbers after the second to 2, making the second command superfluous.

The total number of lines for each row will automatically be adjusted to the last specified line number, but if the actual number of lines is larger than that, you can specify it explicitly by the code **`T`**:

Example: `T 4`Enter

This specification informs the program that each row will occupy 4 lines, even if all relevant numbers are in the first three.

From the data list, you can also define names for the dimensions. In order to specify such a name you must type the code **N** followed by the sequence number of the dimension and its name. The name will be truncated to its first eight characters. If a name contains spaces or commas or if it starts with a dash (minus sign) you must put it between quotes. If such a quoted name contains a quote itself, that quote must be replaced by two adjacent quotes.
You may also specify a range of sequence numbers: all dimensions in the range will receive the same value, unless the name ends with a number; in that case they will receive increasing numbers.

Example: `N 1-4 Dim_1`Enter

In this example, the first four dimensions will be called `Dim_1`, `Dim_2`, `Dim_3` and `Dim_4`. The code character (`N`) may be omitted the second and following times.

The option **M** can be used to define missing values in the trial configuration. If a coordinate value for a dimension is equal to the missing value for that dimension or greater than that missing value, it will be interpreted as missing (unknown) and the program will replace it by some estimate of its own. See 2.6. In order to define missing values you must enter the code (`M`) followed by a stimulus number, or a range of stimulus numbers, and the missing value.

Examples: `M 1 99`Enter
`M 2-4 999`Enter

In the examples, the first dimension will have 99 as its missing value and the dimensions 2 through 4 will have 999. If you give several specifications of this type after each other, you need not to repeat the code (`M`) each time.

The option **D** can be used to delete dimensions from the list. Type the character `D`, followed by a dimension number or a range of numbers.

Examples: `D 1`Enter
`D 2-3`Enter

The number of dimensions (in the *model* menu) will be adjusted automatically.

If you give several consecutive specifications of this type, you can omit the option character (`D`) in all but the first one.

Note that this option does not modify the trial configuration file but its description.

6.3.2.2 Completely free format

To specify a completely free format you must activate the datalist and set the line number for the first stimulus to zero. For details see 6.3.2.1.

6.3.2.3 Fixed format

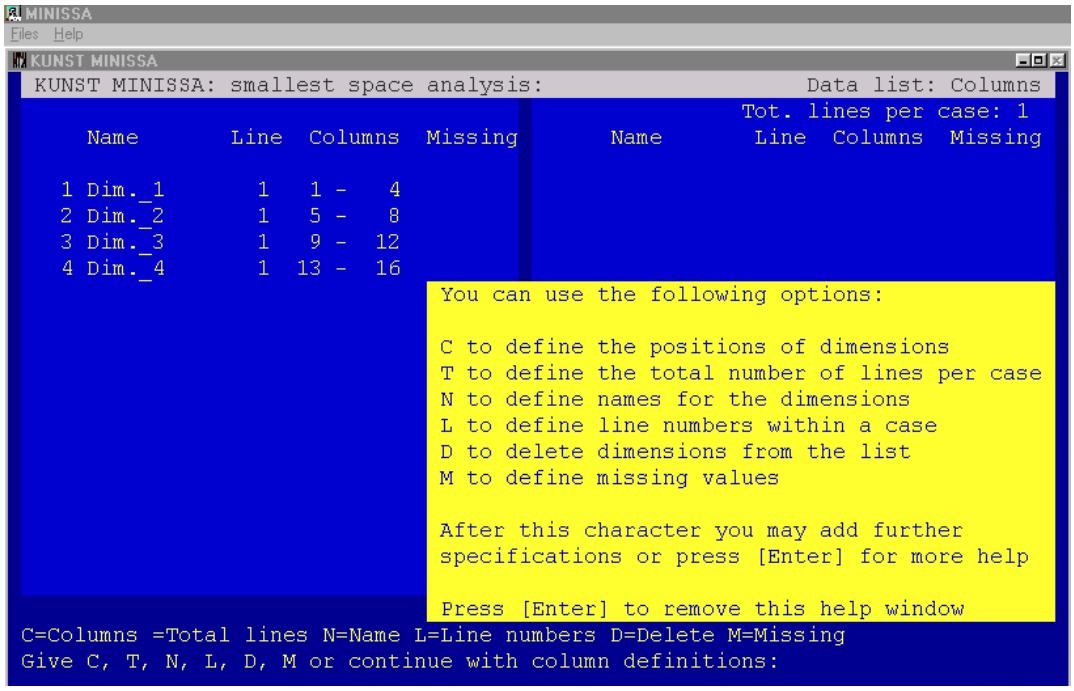

Figure 27: Example of a data list for a trial configuration in fixed format.

Figure 27 shows an example of a data list for a fixed format.

With this format type, the exact line number and position are specified for each coordinate in a row. According to the example in figure 27 the coordinate for the first dimension must be taken from positions 1-4 in the first line, the second coordinate from positions 5-8 in the first line and so on.

To specify the **positions** for the stimuli you must use the code **C**. The use will be clear from the following list of possibilities:

- One coordinate in one column: specify just the dimension and its position:
  `C 3 6Enter`
  {dimension 3 is in position 6}.
- One coordinate in more positions: specify the dimension and the range of positions:
  `C 1 1-4Enter`
  { dimension 1 is in positions 1-4}.
- More coordinates in consecutive columns: specify the range of dimensions and the first position:
  `C 1-4 10Enter`
  { dimensions 1 to 4 are in positions 10-13, each occupying one position}.
- More coordinates in consecutive columns with more than one position per coordinate: specify the range of dimensions and the range of positions. The number of positions must be a multiple of the number of dimensions in the command:
  `C 1-4 1-16Enter`
  {dimensions 1 to 4 are in positions 1-16, each occupying four positions: see figure 27}.

After you have specified the positions for a dimension or a range of dimensions, you may continue defining positions without repeating the character `C` itself.

To specify line numbers you must type the code **L** and then the sequence number of the coordinate and its line number. You may also type a range of sequence numbers.

Examples: `L 2-3 2Enter`
`L 4 3Enter`

These specifications change the line numbers for the dimensions 2 and 3 to 2 and the line number for stimulus 4 to 3. The code (`L`) needs not to be repeated. The line numbers must be in ascending order. Consequently, if there were more than 4 dimensions, the program would automatically change the line number for any further dimensions to 3.

The total number of lines per dimension will automatically be adjusted to the last specified line number, but if the actual number of lines is larger than that, you can specify it explicitly by the code **T**:

Example: `T 4Enter`

This specification informs the program that each row will occupy 4 lines.

From the data list you can also define names for the dimensions. In order to specify such a name you must type the code **N** followed by the sequence number of the dimension and its name. The name will be truncated to its first eight characters. If a name contains spaces or commas or if it starts with a dash (a minus sign) you must put it between quotes. If such a quoted name contains a quote itself, that quote must be replaced by two adjacent quotes.
You may also specify a range of sequence numbers: all dimensions in the range will receive the same value, unless the name ends with a number; in that case they will receive increasing numbers.

Example: `N 1-4 Dim_1Enter`

The code character (`N`) may be omitted the second and following times. In the example, the dimensions will be called `Dim_1`, `Dim_2`, `Dim_3` and `Dim_4`.

The option **M** can be used to define missing values in the trial configuration. If in the file a coordinate value for a dimension is equal to the missing value for that dimension or greater than that missing value, it will be interpreted as missing (unknown) and the program will replace it by some estimate of its own. See 2.6. In order to define missing values you must enter the code (`M`) followed by a stimulus number, or a range of stimulus numbers, and the missing values.

Examples: `M 1 99Enter`
`M 2-4 999Enter`

In the examples, the first dimension will have 99 as its missing value and the dimensions 2 through 4 will have 999. If you give several specifications of this type after each other, you need not to repeat the code (`M`) each time.

The option **D** can be used to delete dimensions from the list. Type the character `D`, followed by a dimension number or a range of numbers.

Examples: `D 1`Enter
`D 2-3`Enter

The number of dimensions (in the *model* menu) will be adjusted automatically.

If you give several consecutive specifications of this type, you can omit the option character (`D`) in all but the first one.

Note that this option does not modify the trial configuration file but its description.

6.3.3 Building the trial configuration

You can also define the trial configuration immediately while entering the options. Therefore, you must choose the option **B** within the *model* menu. If you do so, a trial definition screen shows up as in figure 28.

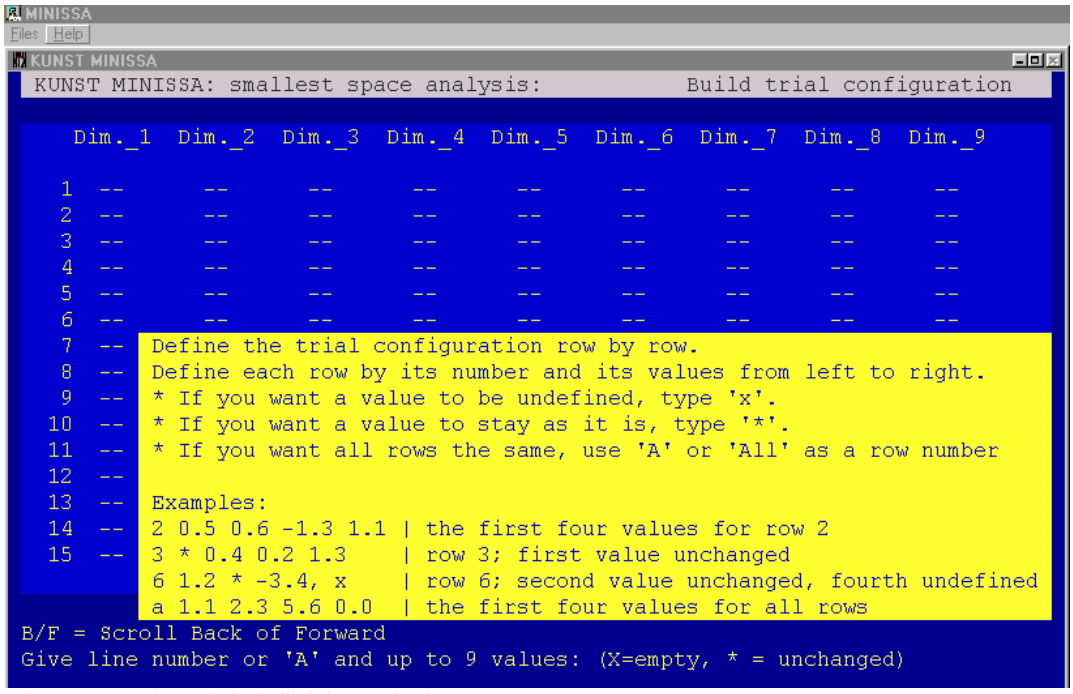

Figure 28: The trial definition window

Now you can enter a row number, defining the stimulus, and a list of coordinates on the dimensions. The number of dimensions is of course limited by the number of dimensions in the trial configuration as you defined it in the *model* menu.

You may however leave some rows empty, and even within one row you may skip dimensions by the use of comma's: `5 22,,17` (or `5, 22,,17`) defines the first and the third coordinate of row five, but not the second. And `6,,16,12` defines the second and the third coordinate of row six, but not the first.

You can also skip coordinates by giving the symbol ***** instead of a value: the asterisk indicates that the corresponding coordinate must stay as it was. If you want to indicate that a coordinate is unknown, you can use the symbol **X** instead of a number.

With the symbol **A** instead of a row number, you specify that the list of values applies to **all** rows in the configuration.

6.4 Additional results in the listing file

If you type `A`Enter in the *main* menu, a new menu appears, in which you can define which results should be shown in the listing file in addition to the standard output. Figure 30 shows an example with the default settings.

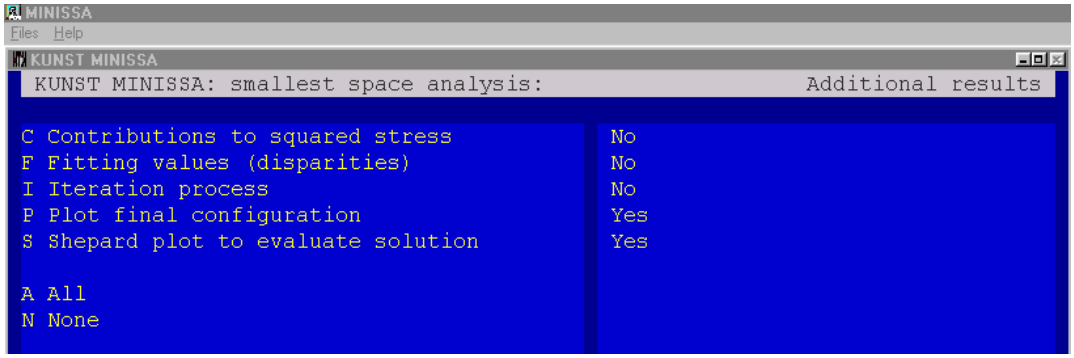

Figure 29: The *additional results* menu.

The five options are all switches: by typing the character code followed by an Enter, the specifications changes from `Yes` to `No` or from `No` to `Yes`.

6.4.1 Contributions to squared stress

If option **C** is chosen, the program will prepare a matrix that shows how each pair of stimuli contributes to the squared stress (based on d^) and also the contribution of each single stimulus.

The contribution for a pair is given by the formula:

$$C_{ij} = \frac{(D_{ij} - F_{ij})^2}{\sum_{i=2}^{n}\sum_{j=1}^{i-1} D_{ij}^2}$$

and the contribution of a single stimulus by the formula:

$$C_i = \frac{\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} (D_{ij} - F_{ij})^2}{2\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}^2}$$

The option **F** produces a list of the distances and the corresponding fitting values.

If **I** is chosen, some information will be given after each iteration. You must not use this option unless you really want to look into the details of the process.

With the option **P** one will get a plot (in the listing file and in a series of bitmap files) of each final configuration. The plot will produce one picture for each pair of dimensions.

With the option **S** one may ask for a Shepard plot (in the listing file and in a bitmap file). As indicated in 2.1 such a plot is helpful in determining the quality of a solution.

The option **A** will set all five switches to `Yes`, the option **N** will set them all to `No`.

7 Results

SocStat will produce one or more output files. We will discuss them now.

7.1 Results in the listing file

For each input file, there will be a listing file, containing the main results of the analysis. Its precise content depends on the specifications by the user and the input data. The lines in a listing have a length of 80 characters or less. View (and print) these files with a small non-proportional font like Courier New 9.

We will use the data of figure 2 to illustrate the results.

The listing will start with a text similar to the one in figure 30.

```
View (and print) this file with a small non-proportional font like Courier 9
 Lines contain 80 characters or less.
 ------------------------------------------------------------------------


 * * * * * * * * * * * * * * * * * * * * * * * * * * * * 13-01-2006  16:25:12

     MINISSA: KUNST program for smallest space analysis
               Interactive version  1.00, January 2005

 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 The operation and the accuracy of the program are not guaranteed.
```

Figure 30: Start of the listing file.

After this introductory text follows an overview of the options as they are chosen by the user. Figure 31 shows an example.

```
Sixteen cities
================================================================================

   1.  Summary of the user specifications and input data.

 ================================================================================

 The data are dissimilarities in a lower triangle with diagonal.
 They will be read from the following file:
  C:\MyProject\Minissa16.dat

 The number of stimuli is 16.

 From the input all rows will be echoed.

 Ties in the data may lead to unequal distances in the solution.

 The number of iterations will be at least 12 and at most 500:
  - At least 6 iterations with D* and at least 6 with D^.
  - At most 250 iterations with D* and at most 250 with D^.

 Metric to be used: Minkowski parameter = 2 (Euclidean).

 A solution will be sought in 2 dimensions.

 Solutions in 9 to 3 dimensions are performed
 to get a trial configuration in 2 dimensions.

 The program will generate a trial configuration of 9 dimensions.

In addition to the standard results, this listing will contain:
```

```
 - A report of the iteration process and the stress.
 - A table of fitting values, distances and contributions to squared stress.
 - Contributions of pairs of stimuli to the squared stress.
 - A Shepard diagram, showing dissimilarities, distances and fitting values.
 - A plot of the final configuration.

The final configuration will be stored in a separate file with the name:
 C:\MyProject\Minissa16.out
```

Figure 31: Overview of the chosen options.

Now follows a description of the format of the input proximities, similar to the text in figure 32.

```
 The data matrix:
 The selected input is supposed to contain the following stimuli:

 Stim.  Name            Line

     1 Leeuward          1
     2 Groninge          1
     3 Den Held          1
     4 Zwolle            1
     5 Amsterda          1
     6 Deventer          1
     7 Enschede          1
     8 Den Haag          1
     9 Utrecht           1
    10 Rotterda          1
    11 Nijmegen          1
    12 Middelbu          1
    13 Tilburg           1
    14 Eindhove          1
    15 Venlo             1
    16 Maastric          1

 The dissimilarities will be read now from the following file:
 C:\MyProject\Minissa16.dat
```

Figure 32: Description of the format of the proximity data.

Next comes a report of the proximity data as they are found in the main input file. The extensiveness of the report depends on the option Echo in the *data* menu. Figure 33 gives an example.

```
The dissimilarities:

 Row    1:     0.0000

 Row    2:    54.0000   0.0000

 Row    3:   287.0000 298.0000   0.0000

 Row    4:    95.0000 106.0000 192.0000   0.0000

 Row    5:   206.0000 217.0000  81.0000 111.0000   0.0000

 Row    6:   126.0000 137.0000 184.0000  31.0000 103.0000   0.0000

 Row    7:   162.0000 173.0000 245.0000  67.0000 164.0000  61.0000   0.0000

 Row    8:   242.0000 253.0000 121.0000 147.0000  63.0000 139.0000 200.0000
               0.0000

 Row    9:   182.0000 193.0000 120.0000  87.0000  39.0000  79.0000 140.0000
              60.0000   0.0000

 Row   10:   238.0000 249.0000 144.0000 143.0000  86.0000 135.0000 196.0000
              23.0000  56.0000   0.0000

 Row   11:   190.0000 201.0000 197.0000  95.0000 116.0000  64.0000 101.0000
             137.0000  77.0000 133.0000   0.0000
```

```
 Row   12:   365.0000 376.0000 271.0000 270.0000 213.0000 243.0000 280.0000
             150.0000 183.0000 127.0000 179.0000   0.0000

 Row   13:   252.0000 263.0000 190.0000 157.0000 109.0000 129.0000 166.0000
              94.0000  70.0000  71.0000  65.0000 114.0000   0.0000

 Row   14:   262.0000 273.0000 200.0000 167.0000 119.0000 139.0000 176.0000
             131.0000  80.0000 108.0000  75.0000 151.0000  37.0000   0.0000

 Row   15:   251.0000 262.0000 252.0000 156.0000 171.0000 125.0000 162.0000
             183.0000 132.0000 160.0000  61.0000 203.0000  89.0000  52.0000
               0.0000

 Row   16:   320.0000 331.0000 298.0000 225.0000 217.0000 194.0000 231.0000
             229.0000 178.0000 206.0000 130.0000 249.0000 135.0000  98.0000
              69.0000   0.0000

    16 rows are read.
```

Figure 33: Echo of the data.

After the report of the input proximities, the listing file will always show the dissimilarities matrix, as it is reconstructed from the input. Figure 34 shows an example.

```
Matrix of dissimilarities:

               1        2        3        4        5        6        7
             Leeuward Groninge Den Held Zwolle   Amsterda Deventer Enschede

 2 Groninge   54.0000
 3 Den Held  287.0000 298.0000
 4 Zwolle     95.0000 106.0000 192.0000
 5 Amsterda  206.0000 217.0000  81.0000 111.0000
 6 Deventer  126.0000 137.0000 184.0000  31.0000 103.0000
 7 Enschede  162.0000 173.0000 245.0000  67.0000 164.0000  61.0000
 8 Den Haag  242.0000 253.0000 121.0000 147.0000  63.0000 139.0000 200.0000
 9 Utrecht   182.0000 193.0000 120.0000  87.0000  39.0000  79.0000 140.0000
10 Rotterda  238.0000 249.0000 144.0000 143.0000  86.0000 135.0000 196.0000
11 Nijmegen  190.0000 201.0000 197.0000  95.0000 116.0000  64.0000 101.0000
12 Middelbu  365.0000 376.0000 271.0000 270.0000 213.0000 243.0000 280.0000
13 Tilburg   252.0000 263.0000 190.0000 157.0000 109.0000 129.0000 166.0000
14 Eindhove  262.0000 273.0000 200.0000 167.0000 119.0000 139.0000 176.0000
15 Venlo     251.0000 262.0000 252.0000 156.0000 171.0000 125.0000 162.0000
16 Maastric  320.0000 331.0000 298.0000 225.0000 217.0000 194.0000 231.0000


               8        9       10       11       12       13       14
             Den Haag Utrecht  Rotterda Nijmegen Middelbu Tilburg  Eindhove

 9 Utrecht    60.0000
10 Rotterda   23.0000  56.0000
11 Nijmegen  137.0000  77.0000 133.0000
12 Middelbu  150.0000 183.0000 127.0000 179.0000
13 Tilburg    94.0000  70.0000  71.0000  65.0000 114.0000
14 Eindhove  131.0000  80.0000 108.0000  75.0000 151.0000  37.0000
15 Venlo     183.0000 132.0000 160.0000  61.0000 203.0000  89.0000  52.0000
16 Maastric  229.0000 178.0000 206.0000 130.0000 249.0000 135.0000  98.0000


              15
             Venlo

16 Maastric   69.0000


 Total pairs of dissimilarities:                  7140.
 Between members of the same tie:                   15.
 ------------------------------------------------------------
 Proportion of information lost by ties:          0.0021
```

Figure 34: Report of the input data.

If the input consists of dissimilarities and all input lines have been echoed, this part will be almost identical to the information of the foregoing block, except for the layout. If the input consists of similarities, also the signs of all values will be reverted. If some of the stimuli are deselected for the analysis, they will be left out here and in all further results.

The listing will also show the proportion of information lost by ties in the data. This proportion is defined as the ratio of (1) the number of comparisons that can be made between proximities within ties and (2) the overall number of comparisons that can be made between proximities overall (in the example it is `15 / 7140 = 0.0021`)

If the trial configuration must be read from a file, the listing will also show the input from that file, depending on the option `Echo` in the *main* menu.

The following parts of output will be repeated for several numbers of dimensions, starting with the maximum number; As long as the number of dimensions exceeds the maximum for which a solution is asked (so the program is only preparing a good trial configuration) nothing is reported except the information under point 3 (see below).

1 The trial configuration for the current number of dimensions will be shown, as in figure 35.

```
  Trial configuration
  ___________________

 Stimuli         Dimensions

                  1        2

   1 Leeuward    4.0606   0.1723
   2 Groninge    4.2323  -0.2891
   3 Den Held   -0.2984  -3.0505
   4 Zwolle      1.7737   0.2552
   5 Amsterda   -0.0385  -1.3819
   6 Deventer    1.2613   0.4642
   7 Enschede    1.7037   1.7961
   8 Den Haag   -0.9455  -1.8861
   9 Utrecht    -0.1511  -0.6505
  10 Rotterda   -1.0618  -1.4364
  11 Nijmegen   -0.2064   1.1192
  12 Middelbu   -3.4044  -0.7421
  13 Tilburg    -1.5960   0.2112
  14 Eindhove   -1.7844   0.7791
  15 Venlo      -1.1706   2.0801
  16 Maastric   -2.3747   2.5593
```

Figure 35: The trial configuration.

2 If a detailed report of the iteration process is requested the listing contains at each iteration step the iteration number, the raw stress $d^{\wedge}$, the standardized stress $d^{\wedge}$, the raw stress d^{*}, the coefficient of alienation and the angle between successive gradients. Figure 36 shows an example.

```
Minimizing raw stress ('soft squeeze').
     Fitting values are rank images (D*).
     Numbered lines: computation of fitting values and evaluation of stress.
     Unnumbered lines: moving the points in the configuration.
     '#' indicates that the gradient angle is used.

     Iter          Raw stress     Coeff.alien   Cos. of
                   D*             D*            grad.
     ____________________________________________________


        1          10.01727114    0.11395076
                    5.87447343    0.08770591    0.00000000
```

```
         #          5.50397977    0.08483738    0.00000000
         #          5.40862019    0.08409163    0.00000000
         #          5.35942737    0.08370541    0.00000000

        2           4.88090505    0.08114567
                    4.69296969    0.07982645    0.00000000
         #          4.64603550    0.07942273    0.00000000
         #          4.61803759    0.07918157    0.00000000

        3           4.44688048    0.07759042
                    4.40413429    0.07744908    0.00000000

    ..... and so on .........

       25           4.25464485    0.07557943

     Algorithm switches to monotone regression,
        because stress did not decrease in the last 4 iterations.
     Retrieving best configuration so far: as it was after iteration 21.

     Minimizing standardized stress ('hard squeeze').
     Fitting values are based on monotone regression (D^).
     Numbered lines: computation of fitting values and evaluation of stress.
     Unnumbered lines: moving the points in the configuration.
     '#' indicates that the gradient angle is used.

     Iter           Raw stress                  Cos. of
                    D^            Stress D^     grad.
     ____________________________________________________


       22           2.51497186    0.05814543
       23           2.48166787    0.05787832    0.00000000
       24#          2.46893830    0.05777048    0.00000000
       25#          2.46171967    0.05770568    0.00000000
    ..... and so on .........
       48#          2.38060332    0.05690981    0.00000000
       49#          2.37986374    0.05690673    0.00000000
```

Figure 36: Details of the process

3 During the minimization process, the program reports when it changes from raw stress with d* to standardized stress with d^ and why it does so. It reports also when the iteration process stops and why. A summary of the results is given starting with the raw stress $d^{\wedge}$, the stress $d^{\wedge}$, the raw stress d^{*} and the coefficient of alienation (also based on d^{*}). Figure 37 contains an example of this basic information.

```
Sixteen cities
 ========================================================================

   2. Looking for a solution in 9 dimensions.
 ========================================================================

 Algorithm switches to monotone regression,
    because the maximum number of iterations is reached: 250.
 Results may be suboptimal, since no convergence was reached.
 You might repeat the analysis with a larger number of iterations.
 Retrieving best configuration so far: as it was after iteration 250.

 Iterations stop because relative decrease of stress is too slow or negative.
     Best stress so far:   0.00103844 at iteration     352
         Current stress:   0.00103834 at iteration     353
               Decrease:   0.00000010
                             < 0.0002 *   1**2 *   0.00103834
                                        ||          ||
                                   (steps)      (stress)

 Performing a principal axes rotation in 9 dimensions.

 Stress D^                          =      0.00103834
```

```
 Raw stress D^                        =      0.01588534
 Raw stress D*                        =      0.04501470
 Coefficient of alienation (D*)       =      0.00174791

 Number of iterations using D*:  250
 Number of iterations using D^:  103
```

Figure 37: Basic information on the process.

4 The final configuration with means and standard deviations per dimension are shown. See figure 38.

```
Final configuration:
     ____________________

     Stimuli         Dimensions

                      1        2

       1 Leeuward    2.5559  -0.2150
       2 Groninge    2.7921   0.1789
       3 Den Held   -0.0849  -2.3355
       4 Zwolle      1.1060   0.0720
       5 Amsterda    0.0561  -0.8919
       6 Deventer    0.7757   0.2118
       7 Enschede    1.1445   1.0098
       8 Den Haag   -0.6713  -1.0537
       9 Utrecht    -0.1126  -0.3733
      10 Rotterda   -0.8288  -0.7778
      11 Nijmegen   -0.0869   0.6169
      12 Middelbu   -2.7031  -0.4947
      13 Tilburg    -0.9971   0.1557
      14 Eindhove   -0.9926   0.5760
      15 Venlo      -0.6876   1.3052
      16 Maastric   -1.2656   2.0156




      Dim.  Means  Stand. dev.
      ____________________________

       1    0.0000    1.3691
       2    0.0000    1.0000
```

Figure 38: The final configuration with means and standard deviations per dimension.

5 If the fitting values are asked, MINISSA will show a long list of all pairs of stimuli ordered by their dissimilarities with the following pieces of information in its columns:

1 The sequence numbers of the stimuli in the pair.
2 The symbol `::` if this pair is tied with its predecessor.
3 The corresponding distance in the configuration.
4 The fitting value d*.
5 The fitting value d^.
6 The number of equal d^ values after each other (only with the last of the equal values).
7 The contribution of the pair to the squared stress ($d^{\wedge}$).

Figure 39 shows an example.

Table of fitting values and related information:

Legend:
Dissim. Pair: Sequence numbers of pairs of stimuli. The column is sorted according to the dissimilarities.
Tied: The mark '::' indicates that the pair has the same dissimilarity as the previous one.
Distance: The distance of the pair in the solution.
Fitting values D*: The rank images: this column is a copy of the Distance column but with its values sorted.
Fitting values D^: Fitting values according to the monotone regression method. The values stay as close as possible to the distances (least squares) but they conform to the ascending order in the dissimilarities.
Size of tie: The number of equal values in the D^ column.
Contribution to the squared stress: The squared difference between D^ and Distance, divided by the sum of the squared distances.

Dissimilarities Pair	Tied	Distance	Fitting values D*	D^	Size of tie	Contribution to squared stress (D^)
Rotterda Den Haag		0.3178	0.3178	0.3178	1	0.00000000000
Deventer Zwolle		0.3587	0.3587	0.3587	1	0.00000000000
Eindhove Tilburg		0.4203	0.4203	0.4203	1	0.00000000000
Utrecht Amsterda		0.5453	0.4593	0.5453	1	0.00000000000
Venlo Eindhove		0.7904	0.5453	0.6249		0.00003725040
Groninge Leeuward		0.4593	0.7452	" "	2	0.00003725040
..... *and so on*						
Maastric Den Held	::	4.5085	4.4249	4.4625		0.00000287262
Maastric Leeuward		4.4249	4.4541	" "		0.00000192270
Maastric Groninge		4.4541	4.5085	" "	3	0.00000009503
Middelbu Leeuward		5.2664	5.2664	5.2664	1	0.00000000000
Middelbu Groninge		5.5364	5.5364	5.5364	1	0.00000000000
			Total squared stress (D^):			0.00323837579
			Stress (D^):			0.05690672887

Figure 39: Report of the fitting values.

6 Optionally the contributions of the stimulus pairs to the squared stress are given in matrix form (based on $d^{\wedge}$). In addition, a list of the contributions per stimulus will be given along with the matrix. Figure 40 gives an example. Note that - in order to improve the readability - the values are multiplied by a suitable constant, so you must multiply them by the inverse of that constant to get the correct values.

Contribution to the squared stress (D^) for each pair of stimuli:

(The values must be multiplied by 0.00001)

	1 Leeuward	2 Groninge	3 Den Held	4 Zwolle	5 Amsterda	6 Deventer	7 Enschede
2 Groninge	3.7250						
3 Den Held	20.4350	0.0000					
4 Zwolle	1.0918	5.3564	0.2435				
5 Amsterda	7.2802	0.0000	7.7030	0.5873			
6 Deventer	0.4768	1.7449	0.0435	0.0000	0.6810		
7 Enschede	4.0034	13.7983	5.9687	0.0527	0.1001	0.0818	
8 Den Haag	0.0549	0.1507	13.5521	0.5471	1.6261	0.0489	0.7157
9 Utrecht	0.3265	4.1905	7.6378	0.3892	0.0000	0.0000	0.1602
10 Rotterda	0.7995	1.7059	5.0880	5.0880	13.9269	0.2220	1.9109
11 Nijmegen	0.2435	1.1228	2.3352	0.0000	3.2718	0.0000	1.2474
12 Middelbu	0.0000	0.0000	26.5906	5.8884	0.1530	5.1906	17.2993
13 Tilburg	0.5040	2.9126	0.0435	0.5641	0.0041	0.0008	2.7264
14 Eindhove	0.0073	0.1305	7.0915	0.0044	0.8421	1.3035	0.0000
15 Venlo	0.3554	0.0288	0.3039	2.5240	3.2879	0.4157	4.6319

```
16 Maastric    0.1923   0.0095   0.2873   0.6064   5.1303   0.4418  20.4526


                8        9       10       11       12       13       14
              Den Haag Utrecht  Rotterda Nijmegen Middelbu Tilburg  Eindhove

 9 Utrecht     0.0907
10 Rotterda    0.0000   0.0000
11 Nijmegen    0.7619   0.0000   4.9905
12 Middelbu    0.6105   0.1289   2.0972   6.4853
13 Tilburg     0.0000   0.6409   0.0243   0.5235   3.2718
14 Eindhove    1.6516   0.9148   2.2031   0.4155   0.0678   0.0000
15 Venlo       9.5555   0.0011   0.3055   0.4735   1.9239   0.3892   3.7250
16 Maastric    2.1866   0.1103   0.0051   0.4553  28.9587   0.1614   0.8050


               15
              Venlo

16 Maastric    0.2440




 Contribution to the squared stress (D^) for each stimulus:

                1 Leeuward          0.000197
                2 Groninge          0.000174
                3 Den Held          0.000487
                4 Zwolle            0.000115
                5 Amsterda          0.000223
                6 Deventer          0.000053
                7 Enschede          0.000366
                8 Den Haag          0.000158
                9 Utrecht           0.000073
               10 Rotterda          0.000192
               11 Nijmegen          0.000112
               12 Middelbu          0.000493
               13 Tilburg           0.000059
               14 Eindhove          0.000096
               15 Venlo             0.000141
               16 Maastric          0.000300
 ____________________________________________
 Total squared stress (D^):         0.003238
 Stress (D^)              :         0.056907
```

Figure 40: contributions of stimulus pairs and individual stimuli to the squared stress.

7 The resulting distances are shown, as in figure 41.

```
Distances:
 __________


                1        2        3        4        5        6        7
              Leeuward Groninge Den Held Zwolle   Amsterda Deventer Enschede

 2 Groninge    0.4593
 3 Den Held    3.3867   3.8209
 4 Zwolle      1.4780   1.6895   2.6859
 5 Amsterda    2.5898   2.9381   1.4505   1.4252
 6 Deventer    1.8306   2.0167   2.6887   0.3587   1.3176
 7 Enschede    1.8687   1.8453   3.5640   0.9386   2.1911   0.8791
 8 Den Haag    3.3343   3.6762   1.4095   2.1038   0.7452   1.9223   2.7487
 9 Utrecht     2.6732   2.9568   1.9624   1.2974   0.5453   1.0637   1.8690
10 Rotterda    3.4311   3.7452   1.7262   2.1132   0.8922   1.8851   2.6626
11 Nijmegen    2.7706   2.9121   2.9523   1.3114   1.5155   0.9530   1.2925
12 Middelbu    5.2664   5.5364   3.2005   3.8511   2.7877   3.5499   4.1313
13 Tilburg     3.5722   3.7893   2.6530   2.1048   1.4855   1.7737   2.3056
14 Eindhove    3.6355   3.8055   3.0497   2.1583   1.8040   1.8054   2.1807
15 Venlo       3.5820   3.6574   3.6902   2.1766   2.3195   1.8267   1.8557
16 Maastric    4.4249   4.4541   4.5085   3.0663   3.1938   2.7241   2.6116


                8        9       10       11       12       13       14
```

```
               Den Haag Utrecht  Rotterda Nijmegen Middelbu Tilburg  Eindhove

 9 Utrecht      0.8804
10 Rotterda     0.3178   0.8225
11 Nijmegen     1.7699   0.9905   1.5797
12 Middelbu     2.1074   2.5934   1.8956   2.8426
13 Tilburg      1.2526   1.0306   0.9486   1.0204   1.8258
14 Eindhove     1.6611   1.2944   1.3636   0.9066   2.0180   0.4203
15 Venlo        2.3590   1.7742   2.0877   0.9136   2.7022   1.1904   0.7904
16 Maastric     3.1264   2.6526   2.8274   1.8292   2.8928   1.8792   1.4653


               15
              Venlo
16 Maastric    0.9159
```

Figure 41: the distances according to the final solution.

8 Optionally a Shepard plot and plots of the configuration are given. In a Shepard plot degeneration shows by rows of distances and fitting values (*) on the same horizontal position. Violations of the model show as a descending sequence of distances (X). Figure 42 shows the Shepard plot, Figure 43 the final configuration from the example analysis.

```
Shepard diagram:
________________

               x-axis= rank order of dissimilarities
               y-axis= distances      (X)
                       fitting values (-)
                       both           (*)
       .+.........+.........+.........+.........+.........+.........+.
  5.536 :                                                           *:
  5.412 :                                                           :
  5.288 :                                                         * :
  5.164 :                                                           :
  5.039 -                                                           -
  4.915 :                                                           :
  4.791 :                                                           :
  4.667 :                                                           :
  4.542 :                                                         x :
  4.418 -                                                         ** -
  4.294 :                                                           :
  4.170 :                                                       x   :
  4.045 :                                                           :
  3.921 :                                                           :
  3.797 -                                                     x   x *-*  -
  3.673 :                                                     -***--    :
  3.548 :                                                   xx x        :
  3.424 :                                                   x         x :
  3.300 :                                                   -*--        :
  3.176 -                                                 xx        x   -
  3.051 :                                               x   -*-         :
  2.927 :                                           x x x  *    x       :
  2.803 :                                        x  x---*-**            :
  2.679 :                                       **-***x  x              :
  2.554 -                                         x       x   x         -
  2.430 :                                                           :
  2.306 :                                      xx  x                :
  2.182 :                                    x  **-*                :
  2.057 :                                 x  x***-                  :
  1.933 -                        x x   x-*--                        -
  1.809 :                      x x***-*-*xx    x  x                 :
  1.685 :                    x ----  x     x                        :
  1.560 :                       x     x                            :
  1.436 :              x  **-**  x                                   :
  1.312 -            x x**xxx                                       -
  1.188 :            --*                                          :
  1.063 :       xx  *                                             :
  0.939 :    xx***** x                                            :
  0.815 :  x*---                                                  :
  0.691 -      x                                                  -
  0.566 :  *-                                                     :
  0.442 : * x                                                     :
  0.318 :**                                                       :
       .+.........+.........+.........+.........+.........+.........+.
      1.0       20.8       40.7       60.5       80.3      100.2      120.0

 This Shepard plot is also given in file:
 C:\Kunst2005\Minissa\Minissa16S.bmp
```

Figure 42: A Shepard plot in the listing file.

```
              Plot of the configuration:

 Dimension  1 against  2.             Dimension       Scaling factor=100/  2.792
                                            2
         -100   -80   -60   -40   -20     0    20    40    60    80   100
          ..+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.
      100 :                               ::                              :  100
       96 :                               ::                              :   96
       91 :                               ::                              :   91
       86 :                               ::                              :   86
       81 :                               ::                              :   81
       77 :                               ::                              :   77
       72 :                   16          ::                              :   72
       67 :                               ::                              :   67
       62 :                               ::                              :   62
       58 :                               ::                              :   58
       53 :                               ::                              :   53
       48 :                               ::                              :   48
       43 :                         15    ::                              :   43
       39 :                               ::                              :   39
       34 :                               ::           7                  :   34
       29 :                               ::                              :   29
       24 :                               ::                              :   24
       20 :                      14       11                              :   20
       15 :                               ::                              :   15
       10 :                               ::                              :   10
        5 :                      13        ::       6                     2:    5
 Dim.   1  ::::::::::::::::::::::::::::::::::::::::::::4::::::::::::::::::::    0
       -5 :                               ::                            1 :   -5
      -10 :                               :9                              :  -10
      -15 :  12                           ::                              :  -15
      -20 :                               ::                              :  -20
      -24 :                        10     ::                              :  -24
      -29 :                               :5                              :  -29
      -34 :                           8   ::                              :  -34
      -39 :                               ::                              :  -39
      -43 :                               ::                              :  -43
      -48 :                               ::                              :  -48
      -53 :                               ::                              :  -53
      -58 :                               ::                              :  -58
      -62 :                               ::                              :  -62
      -67 :                               ::                              :  -67
      -72 :                               ::                              :  -72
      -77 :                               ::                              :  -77
      -81 :                               :3                              :  -81
      -86 :                               ::                              :  -86
      -91 :                               ::                              :  -91
      -96 :                               ::                              :  -96
     -100 :                               ::                              : -100
          ..+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.
         -100   -80   -60   -40   -20     0    20    40    60    80   100

 A graphical representation of the plot of dimension 1 against dimension 2
 is given in file:
 C:\MyProject\Minissa16P.bmp




 ======= Normal end of analysis =======
```

Figure 43: Plot of the final configuration.

7.2 Results in a raw text file

Dependent on the chosen options MINISSA will produce a raw output file for each main data file; it will just contain the final configurations from the highest dimensionality to the lowest as defined by the option `D` in the model menu. There will be one line per stimulus with its coordinates on the nine (or less) dimensions in positions 1-10, 11-20, and so on. Each coordinate value will end with a decimal point and four fractional digits in its last six positions. Figure 44 contains an example.

```
    2.5559   -0.2150
    2.7921    0.1789
   -0.0849   -2.3355
    1.1060    0.0720
    0.0561   -0.8919
    0.7757    0.2118
    1.1445    1.0098
   -0.6713   -1.0537
   -0.1126   -0.3733
   -0.8288   -0.7778
   -0.0869    0.6169
   -2.7031   -0.4947
   -0.9971    0.1557
   -0.9926    0.5760
   -0.6876    1.3052
   -1.2656    2.0156
```

Figure 44: A raw output file.

If a selection of stimuli has taken place, the output file will still have a row for each stimulus, even the ones that are dropped by the selection. Their (unknown) coordinates are replaced by the value 99.0000.

8 Literature

Extra, A.A.J.G., *Nederlands van buitenlanders: psycholinguïstische aspekten van vreemde-taalverwerving*, Nijmegen, 1978.

Kruskal, J.B., *Nonmetric multidimensional scaling: a numerical method*, Psychometrika, 29, pp.45-129, 1964a.

Kruskal, J.B., *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, 29, pp.1-27, 1964b.

Kruskal, J.B. and Wish, M., *Multidimensional scaling*, SAGE Publications, Beverly Hills, 1978, ISBN 0-8039-0940-3.

Lingoes, J.C. and Roskam, E.E.Ch.I. , *A mathematical and empirical analysis of two multidimensional scaling algorithms*, The Psychometric Society, Chicago, 1973. (Psychometrika; vol. 38, nr 4, pl. 2. Monograph suppl. 19)

Lingoes, J.C. editor, *Geometric representation of relational data, readings in multidimensional scaling*, 2nd ed., Mathesis Press, Ann Arbor, Michigan, 1979.

Roskam, E.E.Ch.I. , *Comparison of methods and principles for algorithm construction for nonmetric data analysis*, Michigan Math. psych. progr,. 1969, (2).

Roskam, E.E.Ch.I. and Lingoes, J.C. , *MINISSA-1, a FORTRAN IV(G) program for smallest space analysis of square symmetric matrices*, Behavioral Sciences, 1970, 15, 204-205.

Roskam, E.E.Ch.I. , *A documentation of MINISSA*, Report 75 MA 15, Department of Psychology, Nijmegen University, The Netherlands, 1975.

Schiffman, S.S., Reynolds, M.L. & Young, F.W., *Introduction to multidimensional scaling: theory, methods and applications*, New York, Academic Press, 1981

9 Appendix 1: Monotone regression

The fitting values according to the monotone regression method are computed as follows:

First a copy **F** of the distances is taken, ordered by the corresponding dissimilarities. If these values form an ascending order the stress is zero. Usually they do not. For the explanation in the sequel, we will indicate the fitting values with one single subscript like $\mathbf{F}_i$ to indicate the fitting value corresponding to the i-th dissimilarity.

Now we start from the first fitting value $\mathbf{F}_1$. If the next one ($\mathbf{F}_2$) is smaller we replace them both by their average. Then we check if $\mathbf{F}_3$ is less than this average. If it is, we replace the first three values by their average. If it is not, the average is called *up-satisfied*.

In general, for every **F**, or average of **F**'s, we check if it is less than the preceding **F** or average of **F**'s. If it is not it is *down-satisfied*. We look also if it is larger than the next **F** or average of **F**'s. If it is not, it is called *up-satisfied*. If it is not *down-satisfied* we merge with the preceding **F** or block of **F**'s and assign the average to the whole block. If it is not *up-satisfied* it is merged with the subsequent **F** or block and the average is taken.

If the secondary approach to ties is used (so ties in the data must be reflected by ties in the solution) the initial **F**-values within a tie are replaced by their average before the procedure starts.

As soon as an individual **F** or block is *down-satisfied* ánd *up-satisfied*, the procedure switches to the next **F** and so on, until the end of the list is reached. The result is a series of increasing or at least non-decreasing fitting values that minimize the stress for a given set of distances.

10 Appendix 2: Formulas

X = The configuration (n stimuli by k dimensions)
u = Minkowski parameter
D = The distances (n stimuli by n stimuli):

$$D_{ij} = \left(\sum_{p=1}^{k} \left| \left(X_{ip} - X_{jp} \right) \right|^{u} \right)^{\frac{1}{u}}$$

F = The fitting values (n stimuli by n stimuli)
L = the raw stress:

$$L = \sum_{i=2}^{n} \sum_{j=1}^{i-1} \left(D_{ij} - F_{ij} \right)^2$$

S = the normalized stress:

$$S = \sqrt{\frac{\sum_{i=2}^{n} \sum_{j=1}^{i-1} \left(D_{ij} - F_{ij} \right)^2}{\sum_{i=2}^{n} \sum_{j=1}^{i-1} D_{ij}^2}} = \sqrt{\frac{L}{\sum_{i=2}^{n} \sum_{j=1}^{i-1} D_{ij}^2}}$$

In Euclidean metric (u = 2) with the configuration standardized to zero mean for each dimension and to

$$\sum_{i=1}^{n} \sum_{p=1}^{k} X_{ip}^2 = n$$

we find $S = \sqrt{\frac{L}{n}}$

Adaptation of the configuration:

The adaptation is based on the partial derivatives of the raw stress L or of the standardized stress S. If the raw stress is used the method is called soft squeeze; if the standardized stress is used it is called hard squeeze. In the first stages of the analysis $F = d^*$ is used in combination with L for stress; in the final stages $F = d^{\wedge}$ is used with S for stress.

MINISSA uses the partial derivatives: $G_{ip} = \frac{\partial L}{\partial X_{ip}}$ or $\frac{\partial S}{\partial X_{ip}}$

If the stress (s) is at its minimum this derivatives will be zero, so $X_{ip} = X_{ip} - bG_{ip}$ for any b.

Using an initial estimate $X_{ip.0}$ for X_{ip} and a suitable value for b one may iteratively compute the right part of the equation and assign it to the left part; the stress will thereby converge towards a minimum (local or global).

$$X_{ip.t+1} = X_{ip.t} - b_t G_{ip.t}$$

For b_t MINISSA uses $\dfrac{\beta_t}{4\sum_{j=1}^{n} W_{ijp.t}}$

where:

$$W_{ijp.t} = \left(\frac{\left|X_{ip.t} - X_{jp.t}\right|}{D_{ij.t}}\right)^{u-2} \quad \text{if } D_{ijt} \neq 0$$

and $W_{ijp.t} = 1$ if $D_{ijt} = 0$

$\beta_t = 1$ for t = 0 (the first time)

$\beta_t = 4^{(q^3)} (\square_{t-1})^{1/3}$ for t > 0

where q is the correlation between successive gradients t and t-1:

$$q = \frac{\sum_{i=1}^{n}\sum_{p=1}^{k} G_{ip.t} G_{ip.t-1}}{\sqrt{\sum_{i=1}^{n}\sum_{p=1}^{k} G_{ip.t}^2 \, n \sum_{i=1}^{n}\sum_{p=1}^{k} G_{ip.t-1}^2}}$$

The rationale for the use of q is, that one can see whether the last step is too large or too small by looking at the angle between the gradient at a certain moment and the gradient along which the last movement has taken place. If the new gradient points in almost the same direction as the old one (so q is near 1) the step was too small. If the new gradient points back almost in the direction one came from (so q is close to -1) then the last step was too large.

For a dominance metric $W_{ijp} = 1$ if $X_{ip} - X_{jp}$ is (one of) the largest coordinate difference(s) and $W_{ijp} = 0$ otherwise.

10.1 The partial derivatives of the raw stress

If the soft squeeze is used the partial derivatives G_{kp} can be found as follows:

$$\frac{\partial L}{\partial D_{ij}} = D_{ij} - F_{ij} \qquad (1)$$

$$\frac{\partial D_{ij}}{\partial D_{ij}^u} = \left(\sum_{p=1}^{k}\left|X_{ip} - X_{jp}\right|^u\right)^{-1} = \frac{1}{u}\left(\frac{1}{D_{ij}}\right)^{u-1} \qquad (2)$$

$$\frac{\partial D_{ij}^u}{\partial X_{kp}} = \sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(e_{ki} - e_{kj}\right) u \left|X_{ip} - X_{jp}\right|^{u-2}\left(X_{ip} - X_{jp}\right) \qquad (3)$$

where $e_{ki} = 1$ if k = i and 0 otherwise

and $e_{kj} = 1$ if k = j and 0 otherwise

Combining (1), (2) and (3) we get:

$$\frac{\partial L}{\partial X_{kp}} = \sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(D_{ij} - F_{ij}\right)\frac{1}{u}\left(\frac{1}{D_{ij}}\right)^{u-1}\left(e_{ki} - e_{kj}\right) u \left|X_{ip} - X_{jp}\right|^{u-2}\left(X_{ip} - X_{jp}\right)$$

where $\frac{1}{D_{ij}}$ is set to zero if D_{ij} is zero.

$$= \sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(D_{ij}-F_{ij}\right)\frac{1}{D_{ij}}\left(\frac{1}{D_{ij}}\right)^{u-2}\left|X_{ip}-X_{jp}\right|^{u-2}\left(e_{ki}-e_{kj}\right)\left(X_{ip}-X_{jp}\right)$$

$$= \sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(D_{ij}-F_{ij}\right)\frac{1}{D_{ij}}\left(e_{ki}-e_{kj}\right)W_{ijp}\left(X_{ip}-X_{jp}\right)$$

$$= \sum_{i=2}^{n}\sum_{j=1}^{i-1}e_{ki}\left(D_{ij}-F_{ij}\right)\frac{1}{D_{ij}}W_{ijp}\left(X_{ip}-X_{jp}\right) +$$

$$+ \sum_{i=2}^{n}\sum_{j=1}^{i-1}e_{kj}\left(D_{ij}-F_{ij}\right)\frac{1}{D_{ij}}W_{ijp}\left(X_{jp}-X_{ip}\right)$$

$$= \sum_{j=1}^{n}\frac{D_{kj}-F_{kj}}{D_{kj}}W_{kjp}\left(X_{kp}-X_{jp}\right)+\sum_{i=1}^{n}\frac{Dik-F_{ik}}{D_{ik}}W_{kip}\left(X_{kp}-X_{ip}\right)$$

and since $D_{ij} = D_{ji}$ and $W_{ijp} = W_{jip}$:

$$= 2\sum_{j=1}^{n}\frac{D_{kj}-F_{kj}}{D_{kj}}W_{kjp}\left(X_{kp}-X_{jp}\right)$$

$$= 2\sum_{j=1}^{n}\left(1-\frac{F_{kj}}{D_{kj}}\right)W_{kjp}\left(X_{kp}-X_{jp}\right) \qquad (4)$$

Defining $C_{kjp} = \left(1-\frac{F_{kj}}{D_{kj}}\right)W_{kjp}$ if $k \neq j$ and arbitrary otherwise we have:

$$\frac{\partial L}{\partial X_{kp}} = 2\sum_{j=1}^{n}C_{kjp}\left(X_{kp}-X_{jp}\right)$$

$$= 2\left(\sum_{j=1}^{n}C_{kjp}X_{kp}-\sum_{j=1}^{n}C_{kjp}X_{jp}\right)$$

10.2 The partial derivatives of the standardized stress

If the hard squeeze method is used the partial derivatives $\mathbf{G}_{kp}$ can be found as follows:

$$S = \sqrt{\frac{L}{N}} \quad \text{with } N = \sum_{i=2}^{n}\sum_{j=1}^{i-1}D_{ij}^{2}$$

If in the Euclidean metric the configuration is standardized to mean zero for each dimension and the sum of all squared coordinates equal to n, we have: $N = n$

$$G_{kp} = \frac{\partial S}{\partial X_{kp}} = \frac{\partial\sqrt{\frac{L}{N}}}{\partial X_{kp}} = \frac{1}{2S}\frac{\partial\frac{L}{N}}{\partial X_{kp}} = N\frac{\partial L}{\partial X_{kp}} - L\frac{\partial N}{\partial X_{kp}} =$$

$$= \frac{1}{2NS}\left(\frac{\partial L}{\partial X_{kp}} - S^{2}\frac{\partial N}{\partial X_{kp}}\right)$$

Now we know from (4) that $\frac{\partial L}{\partial X_{kp}} = 2\sum_{j=1}^{n}\left(1-\frac{F_{kj}}{D_{kj}}\right)W_{kjp}\left(X_{kp}-X_{jp}\right)$ and since N can be written as $\sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(D_{ij}-0\right)^2$ it is clear that $\frac{\partial N}{\partial X_{kp}}$ has the same form as $\frac{\partial L}{\partial X_{kp}}$ with F_{kj} replaced by zero.

So $\frac{\partial N}{\partial X_{kp}} = 2\sum_{j=1}^{n} W_{kjp}\left(X_{kp}-X_{jp}\right)$ and

$$G_{kp} = \frac{\partial S}{\partial X_{kp}} = \frac{1}{NS}\sum_{j=1}^{n}\left(1-\frac{F_{kj}}{D_{kj}}-S^2\right)W_{kjp}\left(X_{kp}-X_{jp}\right)$$

If a solution is sought in only one dimension, MINISSA multiplies S^2 by a factor n/(n-2) in the computation of the partial derivative. We don't know why, but left it that way in order to maintain compatibility with older versions of the program and obeying to the rule: never change a winning team.

11 INDEX