

# ScenLab v1.7 Users Manual

evolve:IT  
Scientific Software Systems  
Erik Gauger and Marc Müller-Stoffels GbR

April 9, 2007



# Preface: Differences between full and light edition of ScenLab

Please note that some of the features described in this manual are only available in the full edition of ScenLab and not in ScenLab Light. The chapters that are only relevant for the full edition are marked with an asterisk (\*) throughout this manual.

The light edition of ScenLab is restricted to 10 key factors and does not include GenScen. Furthermore, the export of results and professional analysis tools such as clustering and multi-dimensional scaling of projection bundles are not supported. The ScenLab Light application window cannot be customized with individual background images and the light edition offers no support for group functionality such as merging of consistency matrices.

In contrast to the full edition, no licence key is needed to clear the light edition which somewhat simplifies the installation process.



# Contents

<b>I</b>	<b>Theoretical Background</b>	<b>11</b>
<b>1</b>	<b>Scenario-Management</b>	<b>13</b>
1.1	Introduction to Scenario-Management . . . . .	13
1.2	The Consistency and Robustness Analysis . . . . .	15
1.2.1	Key factors and their future projections . . . . .	15
1.2.2	Assigning plausibility values to future projections . . . . .	15
1.2.3	Projection Bundles . . . . .	16
1.2.4	Assigning consistency values . . . . .	16
1.2.5	Obtaining overall consistency values for the projection bundles . . . . .	17
1.2.6	The combinatorial problem of the consistency analysis . . . . .	18
1.2.7	The Robustness of a projection bundle . . . . .	19
1.3	Disruptive event analysis - Wild Cards . . . . .	20
<b>2</b>	<b>Algorithms used in ScenLab</b>	<b>23</b>
2.1	The standard Consistency Algorithms . . . . .	23
2.1.1	The main Algorithm . . . . .	23
2.1.2	Reduction modules . . . . .	24
2.2	Introduction to Genetic Algorithms* . . . . .	26
2.2.1	Main features of a genetic algorithm . . . . .	26
2.2.2	Why use a genetic Algorithm? . . . . .	29
2.2.3	What genetic algorithms cannot do . . . . .	30
2.3	Multi-Dimensional-Scaling* . . . . .	30
2.3.1	Visualizing high-dimensional data . . . . .	30
2.3.2	The distance metric and the proximity matrix . . . . .	30
2.3.3	MDS using the SMACOF algorithm . . . . .	31
2.4	Clustering* . . . . .	32
2.4.1	Clusters and projection bundles . . . . .	32
2.4.2	Clustering and MDS . . . . .	33
2.4.3	Automatic recognition of optimal number of clusters . . . . .	33
2.4.4	Clustering algorithms in ScenLab . . . . .	34
2.4.5	Interpretation of results from clustering and MDS . . . . .	35
<b>II</b>	<b>Users' Manual</b>	<b>39</b>
<b>3</b>	<b>Getting started</b>	<b>43</b>
3.1	Installing the Java Virtual Machine . . . . .	43

3.1.1	Installation from CD	43
3.1.2	Downloading the current Virtual Machine	43
3.2	Installing ScenLab	44
3.2.1	Installing under Microsoft Windows	44
3.2.2	Installing under Mac OS X	44
3.2.3	Installing under Linux	44
3.2.4	Alternative installation (all operating systems)	44
3.3	Clearing ScenLab*	44
3.3.1	Clearing under Microsoft Windows	45
3.3.2	Clearing under Mac OS X	45
3.4	Starting ScenLab	45
<b>4</b>	<b>How to use ScenLab</b>	<b>47</b>
4.1	Data input in ScenLab	47
4.1.1	Starting a new project	47
4.1.2	Opening an existing project	47
4.1.3	Saving a project	47
4.1.4	Setting the project parameters	49
4.1.5	The key factor input wizard	51
4.1.6	Editing key factors	51
4.1.7	Creating and editing wild cards	51
4.1.8	The consistency matrix wizard	52
4.1.9	Editing the consistency matrix	52
4.1.10	Merging input data from multiple files*	54
4.1.11	Exporting the consistency matrix to CSV*	55
4.2	Data processing	55
4.2.1	Editing the program parameters	55
4.2.2	Including or excluding wild cards	56
4.2.3	Running the standard algorithm	56
4.2.4	Configuring the genetic algorithm*	57
4.2.5	Saving the GenScen configuration*	59
4.2.6	Loading a GenScen configuration*	59
4.2.7	Running the genetic algorithm*	59
4.3	The results - analysis and export	59
4.3.1	The result viewer	60
4.3.2	Saving results	61
4.3.3	Loading existing results	61
4.3.4	Exporting results*	61
4.3.5	Clustering results*	62
4.3.6	Running MDS on results*	63
4.3.7	The distribution plot	63
4.3.8	Truncating the result list	64
4.3.9	The Selection Basket	64
4.3.10	The Morphological Matrix	65
<b>5</b>	<b>Special features*</b>	<b>67</b>
5.1	The ScenLab Client	67
5.2	XML definitions of the result export	67
5.3	Adjusting header icon and background	68
5.3.1	Adjusting the Look-and-Feel	68

# List of Figures

2.1	MDS plot of results from synthesized data. Highlighted are the two clusters found by hierarchical clustering. . . . .	35
2.2	MDS plot of results from synthesized data. Highlighted are the six clusters found by hierarchical clustering. . . . .	36
2.3	MDS plot of results from synthesized data. Highlighted are the four clusters found by k-means. . . . .	37
4.1	After starting ScenLab the main window is shown. . . . .	48
4.2	Project parameters dialog, default settings. . . . .	50
4.3	The key factor input wizard. . . . .	50
4.4	The consistency matrix wizard. . . . .	53
4.5	The full consistency matrix editor. . . . .	53
4.6	The program settings dialog. . . . .	55
4.7	The Result Viewer. . . . .	60



# List of Tables

1.1	Example of a consistency matrix . . . . .	17
2.1	Overview of the procedures run by a genetic Algorithm. . . . .	29
2.2	Proximity matrix relating to Example 2.1.1 . . . . .	31
5.1	XML tag hierarchy. . . . .	69



Part I

Theoretical Background



# Chapter 1

## Scenario-Management

This chapter will mainly discuss the features of Scenario-Management that are supported by ScenLab - the Consistency Analysis, the Robustness Analysis and the Disruptive Events Analysis (Wild Cards). A short introduction to the process of scenario building will be given.

In order to obtain useful results from the scenario process a good amount of expert knowledge about the field of investigation is necessary. ScenLab is not capable of building good scenarios from bad data. It is merely a tool to manage the collected data at a crucial point in the process. Also it provides some useful tools to support the interpretation of the results. But again, some expert knowledge and some experience in the field of scenario writing is necessary to build useful scenarios, even from good data.

All examples used in the below sections are for the sake of example only and do not represent any actual projects.

### 1.1 Introduction to Scenario-Management

What are scenarios? In our context a scenario is one of many possible pictures of the future. A scenario is not meant to forecast the future but to set the reader thinking about possible futures. Usually a scenario process provides three to five scenarios from the very pessimistic to the very optimistic range. These scenarios are then used to develop a strategy that is flexible and robust enough to work for most of the scenarios and contains an ‘exit strategy’ for the worst case.

We distinguish between narrative scenarios and explorative scenarios. Narrative scenarios are creative stories about the future that are not necessarily based on research. Whereas, explorative scenarios are based on extensive research and analysis of the data. ScenLab that supports the building of explorative scenarios.

A good scenario not only represents a certain picture of the future but explains how the present will evolve into this future. It is required that the developments towards this future are consistent and plausible. In narrative scenarios this task can be easily accomplished by doing ‘backwards engineering’, i.e. the scenario is written first and then how it happened that the present developed in this certain future.

Developing explorative scenarios is a more difficult task. Ideally, we the goal is to find, with as little prejudice and assumptions as possible, of all theoretically possible developments the ones that are most likely.

A standard scenario describes the ‘smooth’ (continuous) development from the present to a possible future. In order to account for discontinuities it is possible to perform a disruptive events analysis by using so-called wild cards. Wild cards are highly unlikely events with a major impact on future developments.

The process of building explorative scenarios consists of five stages. In the first stage the field of research is determined and the time and scope of the analysis are set.

In the second stage the key factors and their respective future projections are found. A key factor is a driving force of the researched field. Its development will strongly influence the development of the whole field. Key factors can be anything from environmental influences, over economic and social developments, to the behavior of stake-holders and competitors, and wild cards. Each of these key factors can usually develop in more than one way. Therefore, future projections are assigned to each key factor. Future projections describe possible developments of a key factor. Depending on the required accuracy of the scenarios and the complexity of the field under investigation usually between 10 and 35 key factors with 1 to 5 respective future projections are identified.

The crucial question for stage three is, which future projections of the key factors are likely to occur in the same consistent and plausible future? Of course it would be easy to simply pick a set of future projections (a so-called projection bundle) and check if they make a consistent and plausible future. However, this is not very accurate and very likely to miss crucial combinations of future projection. To avoid this problem the consistency and robustness analysis are carried out, these are described in detail in the following section.

The consistency and robustness analysis provide the data for stage four - the identification of raw scenarios. This is done by analyzing the data with the goal to find the best candidates for scenarios out of the, usually very large, list of projection bundles. The data analysis tools provided by ScenLab are multi-dimensional scaling, clustering and distribution plots. For more information see chapter 2.

The identified future projections build the skeleton for the final scenarios written in stage five. So far, we only have a accumulation of data that characterizes likely future developments. The task is to write good, easily understandable pictures of the future using this data.

Keep in mind that almost the whole process of building explorative scenarios is based on human intelligence and expert knowledge. A software tool is not able to actually write scenarios, it is only able to support the expert user in finding ‘good’ scenarios. There is a lot of ambiguity in the process due to the fact that different evaluations of the data lead to different results. Therefore, it is highly advisable to perform the scenario process as a team discussing the selection of key factors, future projections, the evaluation of the input and output data as extensively as possible.

## 1.2 The Consistency and Robustness Analysis

This section will explain the consistency analysis and how it can be extended to a robustness analysis by assigning plausibility values to future projections. The computational (combinatorial) problem arising due to a growing amount of key factors will be explained.

### 1.2.1 Key factors and their future projections

The data input for the consistency analysis are key factors or more precisely the selected possible future projections of each key factor.

Key factors are factors that primarily characterize the field under investigation. It is crucial to the whole scenario process that these key factors are selected very carefully. Missing important key factors or using the wrong assumptions will jeopardize the whole scenario process.

One or more future projections are assigned to each key factor. It is important to analyze the field under investigation carefully, but in the process of finding future projections some creativity is needed as well. One has to ‘think what is out of the box’ to account for possible future developments.

Using the standard algorithms for consistency analysis the number of key factors is limited by the computational power available. Using more than about 30 key factors with two to three future projections will result in an unacceptably long duration of calculations. It might even exceed the computational power available. See 1.2.6 for further explanation.

### 1.2.2 Assigning plausibility values to future projections

It is possible to assign plausibility values to the future projections of a key factor, i.e. to state that one of the future projections is expected to be much more likely to occur than the others.

#### **Example 1.2.1**

*The key factor ‘Foreign Policy’ has the future projections ‘Containment’, ‘Pre-emptive Strikes’ and ‘Isolation’. One might consider ‘Containment’ to be the most likely future foreign policy and therefore assign it a plausibility value of 0.65, while ‘Isolation’ is very unlikely with just 0.1. Since plausibility values need to add up to 1 that leaves ‘Pre-emptive Strikes’ with 0.25.*

Keep in mind that these occurrence plausibilities are ad-hoc plausibilities, i.e. they are based on assumptions of one future development without considering dependencies between two or more future developments. Plausibility values are merely a tool to identify futures that are regarded more likely than others; they are not capable of giving an absolute occurrence probability for a future projection.

Not assigning plausibility values is equivalent to assuming an equal distribution of plausibility values among the future projections of a key factor. This is the default setting in ScenLab.

### 1.2.3 Projection Bundles

Thus far, we simply have a set of key factors with one or more future projections each. These future projections need to be grouped into so called projection bundles, i.e. grouping one of the future projections of each key factor. The central task is to identify future projections that can occur simultaneously and therefore can form a consistent picture of the future. To do this manually would be, depending on the size of the project, very tedious if not impossible.

#### Example 1.2.2

Assume we have the key factor ‘Foreign Policy’ as in Example 1.2.1 and another key factor ‘Defense Budget’ with the future projections ‘Increasing Budget’, ‘Stable Budget’ and ‘Decreasing Budget’. Clearly a picture of the future where ‘Preemptive Strikes’ of ‘Foreign Policy’ and ‘Decreasing Budget’ of ‘Defense Budget’ take place is inconsistent, i.e. it is very unlikely that these two developments occur together.

*It is more likely that ‘Preemptive Strikes’ and ‘Increasing Budget’ will occur together, i.e. a future development containing these two projections would be very consistent.*

A usual Scenario project consists of 20 or more key factors with one to four future projections each. All these future projections need to be paired and their respective consistency needs to be evaluated.

A projection bundle consisting of  $n$  future projections requires  $(n/2) \cdot (n-1)$  evaluations of paired consistency values. For a project of ten key factors this means 45 evaluations per projection bundle. For a project of 30 key factors this number grows to 435 evaluations per projection bundle.

#### Example 1.2.3

*For a projection bundle containing 4 future projections one has to do  $\frac{4}{2} \cdot (4-1) = 2 \cdot 3 = 6$  evaluations, namely*

**future projection No.1 needs to be paired and evaluated**

*with No.2*

*with No.3*

*with No.4*

**future projection No.2**

*with No.3*

*with No.4*

**and future projection No.3**

*with No.4.*

### 1.2.4 Assigning consistency values

The best way to assign the paired consistency values is to write all the future projections into a matrix.

Table 1.1 shows such a consistency matrix for just three key factors with two or three respective future projections. Even from this very small example it is obvious that this kind of matrix will grow very rapidly with the number of

		FP			DB			OKF	
		PS	C	I	IB	SB	DB	P1	P2
Foreign Policy	Preemptive S								
	Containment								
	Isolation								
Defense Budget	Increasing	2	1	-1					
	Stable	0	2	0					
	Decreasing	-2	-1	1					
Other key factor	Projection 1	0	1	-1	0	0	1		
	Projection 2	1	0	2	-1	0	0		

Table 1.1: Example of a consistency matrix

key factors. Because of the symmetry of the matrix it is sufficient to fill in the sub-diagonal blocks.

Table 1.1 shows the paired consistency values for the future projections as well. In this example the consistency values range from  $-2$  to  $2$ . Where  $-2$  means ‘fully inconsistent’,  $0$  ‘neutral’ and  $2$  ‘fully consistent’.  $-1$  and  $1$  stand for ‘partially (in)consistent’. ScenLab allows the user to adjust the range of values to a convenient range for the project.

For computational reasons ScenLab will internally re-scale any set range to the interval  $[0, 1]$ . However, this does not change anything for the user.

It is crucial to use expert knowledge and experience to assign consistency values to the pairs of future projections. It is especially important not to use extreme judgements too frequently, that is total (in)consistencies. Usually most of the future projections should be neutral to each other. Many will just be partially (in)consistent.

### 1.2.5 Obtaining overall consistency values for the projection bundles

After building the consistency matrix the task is to find all possible projection bundles and evaluate their overall consistency. The **overall consistency value** is the sum of the paired consistency values of a projection bundle. The overall consistency value of a projection bundle is the main feature used to interpret the quality of such a bundle. A projection bundle with a high consistency value is a good basis for a final scenario.

Other parameters to interpret the goodness of a projection bundle are:

**The average consistency value**, is the consistency value of a projection bundle divided by the number of future projections contained in the bundle. If this value close to the upper limit of the range of consistency values the bundle is very consistent, and therefore a good candidate for a final scenario. The average consistency value is especially helpful to compare projects containing different numbers of key factors, because independently from the project it ranges between the lower and upper limit of paired consistency values.

**The number of total inconsistencies.** A projection bundle might carry a very high overall consistency value, but at the same time contain one or more

total inconsistencies, i.e. the lowest possible paired consistency value for two future projections. If any total inconsistency occurs the projection bundle must be dropped, because a projection bundle containing a total inconsistency is inconsistent itself. ScenLab drops projection bundles containing total inconsistencies automatically.

**The number of partial inconsistencies.** These are very low paired consistency values. A projection bundle with a high overall consistency but with too many of partial inconsistencies might be doubtful. How many partial inconsistencies per projection bundle are acceptable depends on the structure of set of projection bundles. If there are many projection bundles without any partial inconsistencies the projection bundles with partial inconsistencies should not be considered at all. But if the number of projection bundles with no partial inconsistencies is low it might be useful to consider projection bundles with one or two partial inconsistencies as well.

**The overall plausibility** of a projection bundle is derived by multiplying the plausibility values of the future projections contained in a given projection bundle. The value of the overall plausibility ranges from zero to (technically) one, but usually the highest overall plausibility is significantly smaller than one. Hence, down to what value a projection bundle is considered to be plausible depends on the highest overall plausibility value found in a project.

**Example 1.2.4 (Overall plausibility)**

*If the plausibility values of the three future projections contained in a bundle are 0.3, 0.45 and 0.1 respectively, then the overall plausibility of the bundle will be  $0.3 \cdot 0.45 \cdot 0.1 = 0.0135$ .*

*One can already see that the overall plausibility values of a projection bundle usually is a very small number.*

**The robustness.** The robustness is an indicator of the quality of a projection bundle unique to ScenLab. It is a combination of the plausibility value, consistency value and the number of partial inconsistencies of a projection bundle. It will be discussed further in 1.2.7.

## 1.2.6 The combinatorial problem of the consistency analysis

As mentioned earlier it takes  $n/2 \cdot (n - 1)$  evaluations per projection bundle to evaluate all the pairs of future projections, where  $n$  is the number of key factors. Now how many projection bundles are there for a number of  $n$  key factors?

Say each of the  $n$  key factors has  $m$  future projections. In this case there are  $m^n$  possible projection bundles.

**Example 1.2.5**

*Assume we have the key factors A, B and C with their respective future projections A1 and A2, B1 and B2, C1 and C2, i.e. three key factors with two future projections each. In this case we would expect that there are  $2^3 = 8$  projection bundles, namely*

- A1B1C1
- A1B1C2
- A1B2C1
- A1B2C2
- A2B1C1
- A2B1C2
- A2B2C1
- A2B2C2

Now, all these projection bundles need to be checked for their overall consistency value, by performing  $n/2 \cdot (n - 1)$  calculations per projection bundle. To obtain the smallest number of calculations required to find all projection bundles and their respective consistency values we multiply  $m^n \cdot (n/2) \cdot (n - 1)$ , this is a very fast growing number.

**Example 1.2.6**

*For a project with 10 key factors and 2 future projections each the required overall number of calculations amounts to*

$$2^{10} \cdot 5 \cdot 9 = 46,080.$$

*This is already a number which requires computational support.*

*For a project with 25 key factors and 3 future projections each this number amounts to about 254 trillion calculations. This illustrates how fast the number of calculations grows.*

Even with an above average desktop computer a project with about 30 key factors is not solvable (in a decent amount of time) with the classical algorithms used for the consistency analysis.

### 1.2.7 The Robustness of a projection bundle

To make the evaluation of projection bundles easier evolve:IT has developed the robustness value. As mentioned before the robustness of a projection bundle is a combination of the consistency value  $c$ , the plausibility value  $p$  and the number of inconsistencies  $I$ . The robustness  $R$  is derived as follows

$$R = \left( \frac{p \cdot c}{(1 + I)} \right)^{\frac{1}{2}},$$

where the factors  $p$  and  $c$  are dynamically scaled to keep them in a range between 0 and 1 and the 1 in  $(1 + I)$  is added to avoid division by zero.

In contrast to  $c$ ,  $p$  varies over several orders of magnitude. Thus, to obtain a well balanced robustness value  $p$  is not scaled linearly to avoid too heavy dependence of  $R$  on  $c$ .

Why introduce another value that just consists of values already in use? There are two reasons: (i) as discussed above a good projection bundle not only

needs to have a high consistency value, a high plausibility value or a low number of partial inconsistencies, but all three. Therefore the introduced robustness combines these three components in a convenient way and takes care of the problem of the very small plausibility values, i.e. it makes it easier for the user to interpret the quality of a projection bundle.

(ii) The fitness function of the genetic algorithm (see 2.2) needs to evaluate the quality of the projection bundles (individuals). It can do this by only considering the consistency value, but this would neglect the other features of a good projection bundle. By using the robustness as measure of fitness the genetic algorithm is able to search for projection bundles with a favorable combination of good features.

### 1.3 Disruptive event analysis - Wild Cards

As mentioned in section 1.1 scenarios usually describe a continuous development from the present towards a possible future. However, this method does not account for very unlikely events that have a strong impact on the field. Such discontinuities of developments are called Wild Cards.

Since the goal of the scenario process is to provide pictures of the future as a tool for strategy building it is useful to include wild cards in the process in order to identify very robust scenarios, i.e. scenarios that are not affected by wild cards, or worst case and best case scenarios that call for an adjustment of strategy.

A wild card should be characterized by topic, impact, plausibility, time scale and causes<sup>1</sup>.

1. Topic: The subject of the wild card or the sector on which the wild card has direct impact (e.g. technological or social wild card).
2. Impact: What are the consequences of the wild card? Wild cards can have a strong, wide-spread impact, or an impact limited to a specific field. Only include wild cards into the analysis that are relevant to, i.e. have an impact on, the field under investigation.
3. Plausibility: All wild cards are by definition unlikely but some are highly improbable, some are less improbable and some are simply not very probable. Seen from a methodical perspective, it might make sense to take even impossible wild cards into account because the demarcation (often fuzzy at best) between the possible and the impossible is based on the knowledge available at the moment and even on ones personal view of the world.
4. Time-scale: Here it would be necessary to differentiate between wild cards, which are sudden, unique events, and processes, i.e. improbable short-term, medium-term or longer-term developments.

---

<sup>1</sup>After: Steinhilber, Karlheinz: 'The future as Wild Card. A short introduction to a new concept', in: Susan Brockett and Margareta Dahlström (eds.): *Spatial Development Trends. Nordic Countries in a European Context*, Stockholm: Nordregio 2004 (Nordregio Report 2004: 6), pp. 193 - 202

5. Cause: One may distinguish between wild cards which occur without any preparation whatsoever, often in the form of accidents or catastrophes due to a chance coincidence of circumstances, and wild cards which are the result of longer-term processes, typically creeping catastrophes.

In ScenLab wild cards are special key factors. They carry two future projections, the likely (plausible) future projection ‘wild card does not occur’ and the unlikely (not plausible) future projection ‘wild card occurs’. The plausibility of wild card occurrences can be adjusted by the user. However, it is highly advisable that this plausibility is significantly lower than the lowest plausibility value for any future projection of the regular key factors. If this is not the case it needs to be discussed if it is really a wild card, or a key factor with a low plausibility future projection.

ScenLab incorporates the wild cards into the consistency matrix like all other key factors. However, special care must be taken in evaluating the paired consistency values. The future projection ‘wild card does not occur’ should be neutral to any other future projection. This is because if the wild card does not occur nothing should happen to influence the other future projections. The future projection ‘wild card occurs’ should not be totally inconsistent with all future projections of any key factor. This is because this would cause projection bundles containing this future projection to be deleted automatically. Therefore no projection bundles containing this future projection would ever show up in the result list.

For data comparison ScenLab allows the user to run the algorithms with and without including wild cards into the analysis.



## Chapter 2

# Algorithms used in ScenLab

This chapter will briefly explain the main features of the standard algorithm for the consistency analysis. A longer explanation is given for the methods on how efficiently reduce the usually large number of projection bundles to an amount that is manageable without losing important projection bundles and on how genetic algorithms work in principle and about some of the special features of the genetic algorithm used by ScenLab.

An outline of the multidimensional scaling (MDS) using the SMACOF<sup>1</sup> algorithm is given without going into mathematical detail.

### 2.1 The standard Consistency Algorithms

The standard algorithm for the consistency analysis is relatively simple. There are some enhancements to speed it up a little, but performance can not be boosted significantly by them and they do not add anything in principle. Therefore we will just discuss the core algorithm.

#### 2.1.1 The main Algorithm

As mentioned in 1.2.5 the exhaustive analysis of the possible projection bundles requires to build and evaluate all existing projection bundles. This is very easy for a small project (compare Example 1.2.5), but as it is shown in Example 1.2.6 the problem grows rapidly with project size.

The implemented algorithm looks up all possible projection bundles, i.e. it builds all possible combinations of future projections. It then checks the paired consistency values stored in the consistency matrix for every bundle. If a bundle contains a total inconsistency it is deleted right away. If it does not contain any total inconsistencies its overall consistency, overall plausibility, number of partial inconsistencies, and robustness are calculated. This is done by the methods described in 1.2.5 and 1.2.7.

If the list of projection bundles is too long to display, because there are simply too many consistent projection bundles, the algorithm runs a reduction module (see 2.1.2) before displaying the results. The reduction method and the maximum number of projection bundles displayed can be chosen by the user.

---

<sup>1</sup>SMACOF stands for Scaling by Majorizing a Complicated Function.

### 2.1.2 Reduction modules

The standard consistency analysis algorithm implemented in ScenLab provides three different reduction methods, trivial reduction, complete projection scanning and complete combination scanning. The goal of all three methods is to reduce the list of projection bundles to human-manageable size. All three methods have their advantages and drawbacks.

#### Trivial reduction.

This method simply cuts off the list of projection bundles after a certain number given by the user, i.e. it sorts the list according to their overall consistency value and displays the chosen number of projection bundles starting with the highest consistency value.

This method has two drawbacks: (i) there might be a number of projection bundles that have a high consistency value, but at the same time have a relatively high number of partial inconsistencies. This can lead to the loss of fully consistent projection bundles with lower consistency values. (ii) It can happen that all projection bundles representing a certain scenario get deleted from the list because all of them do not make the cut, i.e. have a too low consistency value. This will lead to the loss of this scenario for future considerations. Problem (i) can be fixed by spreading the evaluation scale for the paired consistency values while keeping the threshold value for partial inconsistencies low. This will keep projection bundles containing numerous partial inconsistencies from reaching a high overall consistency value. To fix (ii) it is inevitable to use more sophisticated methods, like the ones described below.

The trivial reduction method is especially useful if one is only interested in highly consistent projection bundles.

#### Complete projection scanning.

To avoid the loss of whole subsets of projection bundles that might represent a certain scenario it is necessary to scan the whole list of projection bundles. One way to do this is by a complete projection scanning; the other by a complete combination scanning (see next section).

The complete projection scanning searches the list of projection bundles for the three (this number can be changed by the user) projection bundles containing future projection 1 of key factor A with the highest consistency value. These bundles will be written to the result list. It then proceeds searching for the high consistency value bundles containing future projection 2 of key factor A, then with future projection 1 of key factor B and so forth. The size of the result list is determined by how many representatives of each bundle containing a certain future projection are picked.

#### Example 2.1.1

Assume we have a project with four key factors A, B, C, D with two future projections each. The following table shows all possible projection bundles with their overall consistency values.

No.	Projection bundle	cVal	No.	Projection bundle	cVal
1	A1B2C1D2	45	9	A1B1C1D1	29
2	A2B1C2D2	44	10	A2B2C1D1	26
3	A1B1C2D2	41	11	A1B2C2D2	24
4	A1B1C1D2	41	12	A2B2C2D2	14
5	A2B2C1D2	39	13	A1B2C1D1	14
6	A2B1C2D1	37	14	A1B2C2D1	11
7	A2B1C1D2	37	15	A2B1C1D1	9
8	A1B1C2D1	33	16	A2B1C1D1	7

If we now run a complete projection scanning picking the three bundles with the highest consistency value for each future projection we will obtain the following reduced list.

P-Bundle No.	cVal	No. of Picks
1	45	4
2	44	4
3	41	4
4	41	3
5	39	3
6	37	3
8	33	1
10	26	1
11	24	1

The result is, that bundles 8, 10 and 11 will be in the list even though they have a considerably smaller consistency value. They would not have been in a list if it would just have been truncated at one third for example. So even in this small project the different reduction method makes significant changes to the outcome of the reduction.

Another important point that can be seen in the example is that reduction by complete projection scanning ensures that the projection bundles with the highest consistency values will remain in the list. In fact they usually represent more than one future projection and are therefore picked more than once.

In a small project like this it might be significant if even one of the future projections in a projection bundle is different. But in bigger projects, where the use of a reduction method is necessary to keep the amount of results manageable, this is not the case. If two projection bundles differ only in one or two future projections they are still very similar and it might therefore be sufficient to keep just one of them.

### Complete combination scanning.

This is the most thorough reduction method provided by ScenLab. The basic idea is the same as for the complete projection scanning. But rather than scanning just for single future projections the scanning is done on combinations of future projections. In the example above one would look for all projection bundles that contain the pair A1B1, then A1B2 ... C2D1 and C2D2.

The advantage is that even projection bundles with a relatively poor consistency value are considered for the reduced list if they contain a combination of future projections not represented in the high consistency value bundles.

The disadvantage is that the reduced result list might still be very long, especially if more than two representatives per combination are picked.

## 2.2 Introduction to Genetic Algorithms\*

What are genetic algorithms? In short: genetic algorithms are special artificial intelligence techniques. They are stochastic algorithms whose search methods model some natural phenomena: genetic inheritance and Darwinian strife for survival. ‘Special artificial intelligence techniques’ refers to the fact that the algorithm is working its way through a problem with a minimum number of boundary conditions. Of course it is not really intelligent by any means, but the implementation is. ‘Stochastic algorithm’ refers to the fact that a genetic algorithm combines both random elements and directed search, i.e. it is not completely random, nor completely deterministic.

### Example 2.2.1

*Suppose you borrowed a friend’s car. She gave you the keys and left without telling you where its parked. And you do not know what her car looks like. You could just randomly try your keys on the cars around until you find the one they open. This strategy will be successful in the long run, but it might take a while.*

*Or you could just try all the red cars because you know that your friend really likes this color and you figure that she would drive a red car. So you put some assumption about a good solution to your problem (finding the car) into your random effort to find a solution.*

‘Genetic inheritance’ is a biological concept. Abstractly, it describes the processing of information from one generation to the next. Genetic algorithms use the concept of genetic inheritance in this abstract way without considering the complex processes used by biological organisms. Usually a very simple mechanism of cross-over and mutation is used.

‘Darwinian strife for survival’, or the concept of selective pressure forces the processing of information (genetic inheritance) in a direction that is favorable for a populations survival. Again, the concept is used in the most abstract way. Natural evolutionary processes depend on many interacting parameters found within the environment. In genetic algorithms the goal is to keep this ‘environment’ as simple as possible.

### 2.2.1 Main features of a genetic algorithm

This section will explain in more detail how the concept of genetic inheritance and Darwinian strife for survival are used in a genetic algorithm. We will also point out the random and deterministic features of the algorithm. The discussion is held as general as possible to keep the level of complexity as low as possible.

There are five main features to a genetic algorithm:

- The representation of potential solutions
- The initial population of potential solutions
- The evaluations function and the selection mechanism
- The genetic operators

- Various parameters (population size, probabilities of applying genetic operators, etc.)

### The representation of potential solutions

Unlike an algorithm that solves a problem analytically a genetic algorithm works not only on one solution at a time but on a whole set of **candidate solutions**. These candidate solutions are referred to as **individuals** or **chromosomes**. The set of candidate solutions is called the **population**.

In simple genetic algorithms the chromosomes usually are bit-strings<sup>2</sup>. Each chromosome possibly represents a solution to the given problem. In the case of the consistency analysis such a solution would be any consistent projection bundle. The units, i.e. the single bits in the chromosome, are called **genes**. The positions of the genes in the chromosome are called **loci**. The state of a gene, i.e. 1 or 0 is referred to as an **allele**. In the case of the consistency analysis these alleles would be the different future projections of a key factor. More complex genetic algorithms might not use bit-string chromosomes, but more complex representations.

Genetic algorithms usually work with individuals that consist just of a chromosome. They do not have a ‘body’ or any other functions since the only aim of their existence is to pass on the information stored in their chromosome. Population sizes are usually held constant or within a certain range. Their size might vary considerably depending on the actual problem.

### The initial population of potential solutions.

Before a genetic algorithm does anything else it needs to have an initial population. This population should be as random as possible because we though know what a good solution looks like, but we do not know where in the search space it is located. Therefore, we want to spread our population over the whole search space as evenly as possible. If the problem sets some constraints on candidate solutions, these constraints must be taken into consideration during the initialization process. But one has to be very careful about these constraints, i.e. total inconsistencies in a projection bundle are no constraint in the initial population.

### The evaluation function and the selection mechanism.

The concept of the Darwinian strife for survival is ‘survival of the fittest’. In order to find out which the fittest individuals of a genetic algorithm population are all individuals need to be evaluated. Their fitness is measured by how good the solution (to the problem they represent) is. In the case of the consistency analysis the fitness value would be equivalent to the robustness or the consistency value. After all individuals are evaluated the ones that are allowed to pass on their information to the next generation have to be selected. Obviously, as in nature and according to the concept of evolution by selective pressure, the fitter individuals will have better chances of being selected.

There is a large variety of selection mechanisms. Some are completely random, i.e. they pick one individual and create a random number in the range of

---

<sup>2</sup>A computer internally represents all numbers as bit-strings. A bit can be either 1 or 0.

fitness values. If the random number is lower than the individuals fitness value it will be allowed to pass on its information, otherwise it will not. These kind of selection mechanisms are called ‘roulette wheels’. Other selection mechanisms simulate a tournament, much like a sports event. They randomly pick a number of individuals and let them compete in a tournament. Like in real life the fitter individuals have a higher probability of winning such a tournament, but every once in a while an underdog (a less fit individual) wins. All selection mechanism can be configured to keep the population size constant, i.e. fitter individuals might be selected more than once, or in a certain range.

Due to the random element of the selection mechanism it is not guaranteed that the fittest individuals will proceed to the next generation, i.e. that they will be selected. However, it is sometimes favorable to ensure that the fittest individuals proceed no matter what. To ensure this a mode called **elitism** is used. This mode lets a selected number of fittest individuals (**elitists**) bypass the selection mechanism, i.e. they will always proceed to the next generation. The number of elitists should not exceed 2 or 3 because they would be too dominant in the next generations.

Which selection mechanism is used depends on the problem. Some let the genetic algorithm converge very rapidly, others explore the search space more thoroughly.

### The genetic operators.

After the next generation of individuals is selected the population needs to be altered in order to explore the search space for better solutions (fitter individuals). This is done using two genetic operators: crossover and mutation.

The **crossover-operator** selects two or more individuals randomly. After the individuals for a crossover are selected the crossover-operator selects a random position in the chromosomes to cut. It then interchanges the parts of the chromosomes that lie behind the cutting point. These new individuals are added to the population. The old ones (their parents) are discarded.

Not all individuals of a population will be allowed to perform crossover. Depending on the problem crossover-rates between 15 and 55 percent might be chosen.

#### Example 2.2.2

Suppose two individuals are selected for crossover. The two individuals are

$$(a_1, a_2, a_3, \dots, a_{cut-1}, a_{cut}, a_{cut+1}, \dots, a_{n-2}, a_{n-1}, a_n)$$

and

$$(b_1, b_2, b_3, \dots, b_{cut-1}, b_{cut}, b_{cut+1}, \dots, b_{n-2}, b_{n-1}, b_n).$$

The cutting point is picked randomly, suppose, at positions  $a_{cut}$  and  $b_{cut}$ . The chromosomes are cut at this position and their tails are interchanged. This leaves us with two new individuals

$$(a_1, a_2, a_3, \dots, a_{cut-1}, b_{cut}, b_{cut+1}, \dots, b_{n-2}, b_{n-1}, b_n)$$

and

$$(b_1, b_2, b_3, \dots, b_{cut-1}, a_{cut}, a_{cut+1}, \dots, a_{n-2}, a_{n-1}, a_n).$$

```

begin
   $t := 0$ ;                               Set iteration number to zero.
  initialize population  $P(t = 0)$ ;
  evaluate  $P(0)$ ;
  while (not termination – condition)      Do this until termination condition is reached.
     $t := t + 1$ ;                           Increase iteration number by one.
    select  $P(t)$  from  $P(t - 1)$ ;           Select new generation from current.
    alter  $P(t)$ ;                             Apply genetic operators.
    evaluate  $P(t)$ ;
  end
end

```

Table 2.1: Overview of the procedures run by a genetic Algorithm.

Unlike the crossover-operator the **mutation-operator** works on single individuals by altering one of its genes. It picks an individual randomly, then randomly picks the position in the chromosome for the mutation to occur, and finally randomly chooses which allele the gene is switched to. The mutated chromosome is added to the population, the old one is deleted. The mutation-operator is a completely random element of the genetic algorithm, therefore the mutation-rate must not exceed about 1 percent, because this would strongly interfere with the deterministic elements of the algorithm, i.e. make it a random search algorithm.

**Example 2.2.3**

Suppose the following individual is selected for mutation

$$(a_1, a_2, a_3, \dots, a_{m-1}, a_m, a_{m+1}, \dots, a_{n-2}, a_{n-1}, a_n).$$

The position for the mutation to occur is  $a_m$ . Now this gene will be altered to a different allele  $a_m^*$ . The new individual is then

$$(a_1, a_2, a_3, \dots, a_{m-1}, a_m^*, a_{m+1}, \dots, a_{n-2}, a_{n-1}, a_n).$$

**Putting the pieces together.**

How does a genetic algorithm work? Table 2.1 shows the procedures of a genetic algorithm. After initializing and evaluating the initial population, it proceeds by selecting the next generation from the current one. The genetic operators are applied to the new generation and the resulting population is evaluated. This procedure runs until a certain termination condition is reached. A termination condition is either that the anticipated result is reached or that a given number of runs (**generations**) is reached.

**2.2.2 Why use a genetic Algorithm?**

As mentioned in 1.2.6 the evaluation of the consistency matrix is a fast growing problem. Even with state-of-the-art desktop computers a moderate amount of key factors is not manageable with the standard algorithm.

The consistency analysis can be considered an optimization problem. That is, one searches for the optimal projection bundles. Optimization problems are the main class of problems tackled with genetic algorithms, these are problems like wire routing, scheduling, aerodynamic design, the traveling sales person problem, optimal control problems, etc. Applied to the consistency analysis, a genetic algorithm enables the user to solve bigger scenario projects, i.e. projects with more than 25 key factors with more than 4 future projections each.

### 2.2.3 What genetic algorithms cannot do

A genetic algorithm is not able to solve a problem analytically, i.e. it will not solve the combinatorial problem explained in 1.2.6, but it will work its way around it.

Usually this method works quite well, but there is absolutely no guarantee whatsoever that the genetic algorithm will find the optimal solution (the best projection bundle) in each run. Therefore it is strongly recommended to run the consistency analysis using the genetic algorithm more than once and to compare the results.

## 2.3 Multi-Dimensional-Scaling\*

This section will briefly outline the problem of visualizing high-dimensional data on a two-dimensional plane. It will explain the metric used in ScenLab and briefly sketch the main features of the SMACOF algorithm, which is used to do the multi-dimensional scaling (MDS).

### 2.3.1 Visualizing high-dimensional data

Mathematically speaking, the projection bundles of the consistency analysis are elements of a high dimensional vector space. Therefore, they can not be visualized in their original dimensionality because visualization of objects is limited to three dimensions at a maximum.

Just as objects in a three dimensional space might be similar or totally different to each other, so are objects in higher dimensions. These similarities can be visualized in two dimensions as distances, i.e. very similar objects will be plotted close together on a plane; very different objects will be far apart from each other. In terms of the consistency analysis projection bundles that differ in only a few future projections are considered similar; projection bundles differing in almost all future projections are considered to be dissimilar. We will refer to similarities and dissimilarities as **proximities**.

### 2.3.2 The distance metric and the proximity matrix

As said in 2.3.1, proximities can be represented by distances between projection bundles. The important task is to use the right kind of metric. In ‘real life’ we usually use a metric called the Euclidian metric to measure distances. Mathematics utilizes many different metrics. These might not be useful to measure distances in real life problems, but they suit other problems much better than the Euclidian metric does.

PB-No.	1	2	3	4	5	6	8	10	11
1	0								
2	3	0							
3	2	1	0						
4	1	2	1	0					
5	1	2	3	2	0				
6	4	1	2	3	3	0			
8	3	2	3	2	4	1	0		
10	2	3	4	3	1	2	3	0	
11	1	2	1	2	2	3	2	3	0

Table 2.2: Proximity matrix relating to Example 2.1.1

Since the space of the projection bundles is a discrete space, a discrete metric is used. To measure the distance between two projection bundles the number of dissimilarities is counted<sup>3</sup>. This number ultimately describes the distance between these projection bundles.

### Example 2.3.1

Consider projection bundle No.1 and No.2 of Example 2.1.1, namely  $A1B2C1D2$  and  $A2B1C2D2$ . They differ in three future projections (in  $A, B$  and  $C$ ). So there distance would be 3.

To setup a visualization we need to calculate the distances between all the projection bundles of a result list. These distances will be written in the so called proximity matrix. For the reduced result list of Example 2.1.1 this proximity matrix is shown in Table 2.2. Due to the symmetry, the distance between bundle 3 and 5 is the same as the one between 5 and 3, it is sufficient to fill out the sub-diagonal fields of the matrix only.

### 2.3.3 MDS using the SMACOF algorithm

Obviously, visualizing a high-dimensional picture in two dimensions will result in the loss of information. Nonetheless, the degree of loss of information can be kept at a minimum.

### Example 2.3.2

Imagine a modern art picture with the title ‘The House on the Beach’. You might be happy that the artist told you in the title what you see, because the painting consists just of a couple of lines in different colors. However nice this picture is its information content about that certain house on the beach is rather poor. It might not look like a house at all. But even if you have a high resolution photograph of the house you will not be able to retrieve all the information about the house. You can not walk around it and look how the back of it looks, for example. Still the loss of information is kept as low as possible when scaling this three-dimensional object on a two-dimensional carrier of information.

<sup>3</sup>Mathematically speaking this is some kind of ‘City-Block-Metric’.

In MDS the task is to map the information content of the data to two dimensions as accurately as possible. Without going through complex mathematical detail, the way the SMACOF algorithm works is to work its way from the artist picture to a photograph of the highest possible resolution.

First the algorithm sets up a random distribution of points on the two-dimensional plane. Each point represents one element of the data set, i.e. a projection bundle. Obviously the information content of this picture relative to the original data will be very low. One says the **stress** of the mapping is very high. This stress will be lowest when points representing very dissimilar projection bundles are as far apart as possible and similar ones are very close together. The algorithm will try new configurations of points that decreases the stress, until the algorithm has worked its way from the abstract picture to a high resolution picture. It will keep on running either until the stress reaches an acceptably low level set by the user, or until a maximum number of iterations is reached.

The process of arranging a large amount of points of which only their distances are known is by no means a trivial task. The SMACOF algorithm uses a considerably high amount of computational power to do.

## 2.4 Clustering\*

This section will outline the general problem of clusters in high-dimensional data sets. The clustering algorithms used provided by ScenLab - k-means and hierarchical clustering - will be explained and differences between MDS and clustering will be discussed. It is advisable that the reader is familiar with section 2.3.

### 2.4.1 Clusters and projection bundles

A cluster is a subset in a set of data whose members share more characteristics with each other than with other members of the set. In most cases these relations are expressed as distances in a vector space, where the ‘coordinates’ in this vector space are the characteristics of the members of the data set.

#### Example 2.4.1

*Let a parking lot be our ‘vector space’. Then all the cars in this parking lot are the members of our data set. Every car is slightly different to all the others. However, there are similarities as well. One subset of the set of cars would be all red cars. If our criterion for comparison of cars was color the red cars would all be in one cluster. If we choose a second criterion for comparison, say the type of a car, then all cars that are a sedan would be in the same cluster. Some of our red cars would be in this sedan cluster as well, but there will be cars of different colors that are sedans as well. These will also be members of other color clusters. Combining clusters red sedans are the closest related to each other. They form a smaller cluster with closer related members.*

Projection bundles are members of the vector space of all possible projection bundles. How close they are related can be measured using a metric described in section 2.3, i.e. the more future projections two projection bundles have in common the closer they are related. The goal of clustering is to subdivide

the set of all projection bundles into a given number of groups of projection bundles that are closely related. In contrast to that, in order to obtain strong clustering the members of different groups need to be as different as possible. For example: a mostly neutral consistency matrix and no specific reduction module do not produce clustered data, because the differences in the resulting projection bundles will be too small.

A division of the projection bundles into explicit well-defined clusters can generally speaking not be expected in a scenario project. Clustering through selection of subsets through reduction modules is not guaranteed to happen either. Only a propitious arrangement of exclusions through total inconsistencies definitely leads to strong clustering of the resulting projection bundles. Therefore, results from the clustering algorithm always ought to be studied with care. It would be unscientific and unreasonable to presuppose the existence of clusters in all scenario projects.

### 2.4.2 Clustering and MDS

Both Clustering and MDS produce results no matter if the data under review has a clustering structure or not. Clustering algorithms always come up with the desired number of clusters whether or not this makes sense for the data. Thus a critical analysis of the outcome is essential.

There is no need whatsoever for clustering and MDS to coincide with their results as these are two completely independent processes. A flagrant mismatch is, however, a strong hint for only a weak structure or maybe even no cluster structure at all.

Please note that structure may be present and may be captured by clustering but not by MDS as it cannot be reduced to two dimensions. In this case clustering and MDS do not match and from the results we just do not know if clusters really exist. Only a detailed manual inspection of similarities within the clusters might give more information.

On the other hand, if the MDS graphical output shows coherent areas of well defined clusters, there is evidence for a real underlying structure in the data corroborated by two independent sources.

### 2.4.3 Automatic recognition of optimal number of clusters

One of the main problems of clustering is that the number of clusters needs to be specified by the user. This approach works well with visual feedback for two or three dimensional data but not on higher dimensional sets of data. However, the set of projection bundles represents a high-dimensional data set and as mentioned above the MDS is only a rough two-dimensional representation of this data.

As a common work-around one specifies a range in which the true amount of clusters is expected. Afterwards, the clustering is performed once at a time for each number  $k$  in the interval and the quality of the results is then recorded. Subsequently the number of clusters that produces the best match is chosen.

The difficulty lies in how to assess the quality of the clustering. This is done by scoring each run with the BIC (Bayesian Information Criterion)<sup>4</sup> and looking

---

<sup>4</sup>The BIC is a common element of Bayesian statistics. However, it is too complex to be discussed here.

for the highest score. Recognizing the number of clusters is still a subject of scientific research and no ‘one size fits all’ solution exists.

#### 2.4.4 Clustering algorithms in ScenLab

ScenLab supports two clustering algorithms: hierarchical clustering and k-means clustering. These two methods differ in their advantages and drawbacks as well as in their results. What follows is a brief overview of the merits and disadvantages.

##### The k-means algorithm

For the k-means algorithm we assume that there has got to be a certain number of clusters in the space of projection bundles. Since the algorithm cannot know before hand where the centers of these clusters are it places these centers randomly in the projection bundle space. After the random placement of cluster centers it assigns each projection bundle to the nearest cluster center. Then the average distance from the projection bundles within the cluster to the cluster center is calculated. The cluster center is moved to the point where the average distance to all projection bundles within the cluster is minimal. Then the projection bundles are again assigned to the nearest cluster center. Since the centers were moved this might be another center than before. Again, the cluster centers are moved to the point where their average distance to their assigned cluster members is minimal. This is done until the set of members to each cluster does not change anymore or only marginally.

The advantage of k-means is that it converges very fast and scales fairly well with the size of the project. Especially for large data sets, i.e. 500 and more projection bundles and 5 to 10 clusters, this algorithm is highly efficient.

The drawbacks are that the results from k-means depend on the random initial configuration of cluster centers. This affects clustering from small data sets, i.e. 50 projection bundles and 5 to 10 clusters. Due to this random element the results of two runs of k-means are usually not completely identical. However, if the underlying data set has a strong structure the results will be similar. Also, k-means is sensitive to outliers and structures with large blank spaces in between, especially for small data sets.

##### Hierarchical clustering

The hierarchical clustering starts from the idea that, since every projection bundle is different to all others it is a cluster of its own. This does not yield the desired cluster number though. To reduce the number of clusters the clusters closest to each other will be merged. This goes forth until the desired number of clusters is reached.

The advantage of hierarchical clustering is that there is no random element. That is that the results of any run of hierarchical clustering on the same data set will yield the same results. Especially for small data sets hierarchical clustering is the better choice.

The drawback is that outliers in the distribution will remain individual cluster for a long time, while big clusters are merged. This can lead to a very uneven number of projection bundles per cluster.

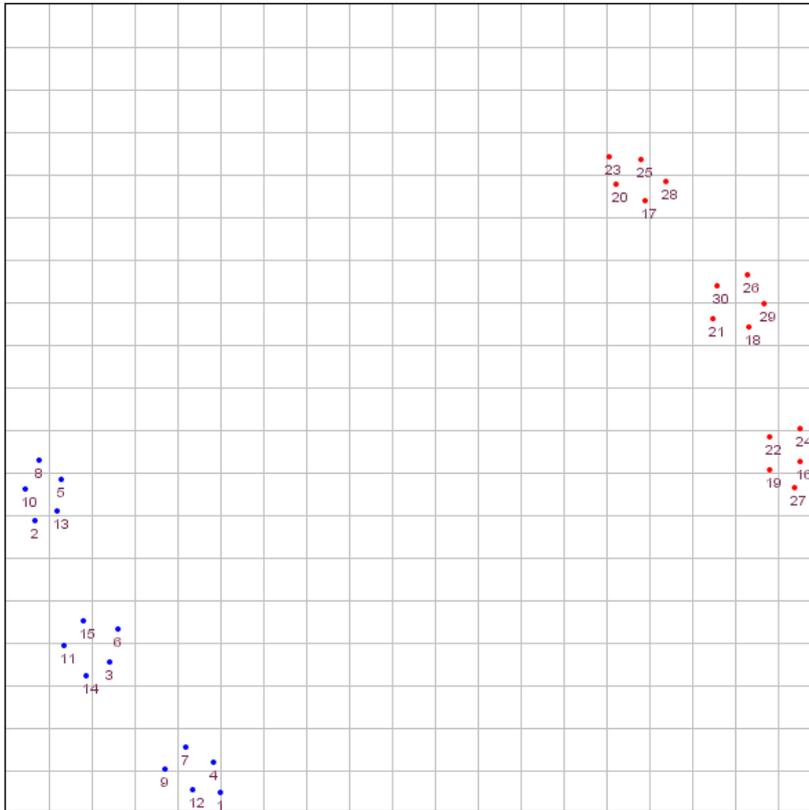


Figure 2.1: MDS plot of results from synthesized data. Highlighted are the two clusters found by hierarchical clustering.

### 2.4.5 Interpretation of results from clustering and MDS

To demonstrate how different the results of the clustering algorithms and the MDS can be we have synthesized a project that per definition has two clusters only. However, these two clusters are not necessarily shown in the MDS plot and due to the small number of projection bundles k-means performs poorly for the reasons mentioned above.

As has been pointed out before, clustering is by no means a process in which pushing a single button and waiting for results is enough. Rather a bit of playing about and a critical analysis as well as comparison of different models is crucial. For this purpose ScenLab provides two clustering algorithms and MDS as independent ways of verifying the integrity and trustworthiness of results.

Not only clustering but also MDS results can be misleading as the enclosed figures show. After MDS it looks as if there are 6 clusters (see figure 2.1, which is, in fact, not true as we know from our synthesized data.

Hierarchical clustering always finds the right number of clusters (see 2.1) but intuitively seems to be wrong for a manual number of 6 clusters (see figure 2.2). A closer analysis shows that it is not wrong at all, as there are many ways of subdividing the two real clusters that are worth exactly the same.

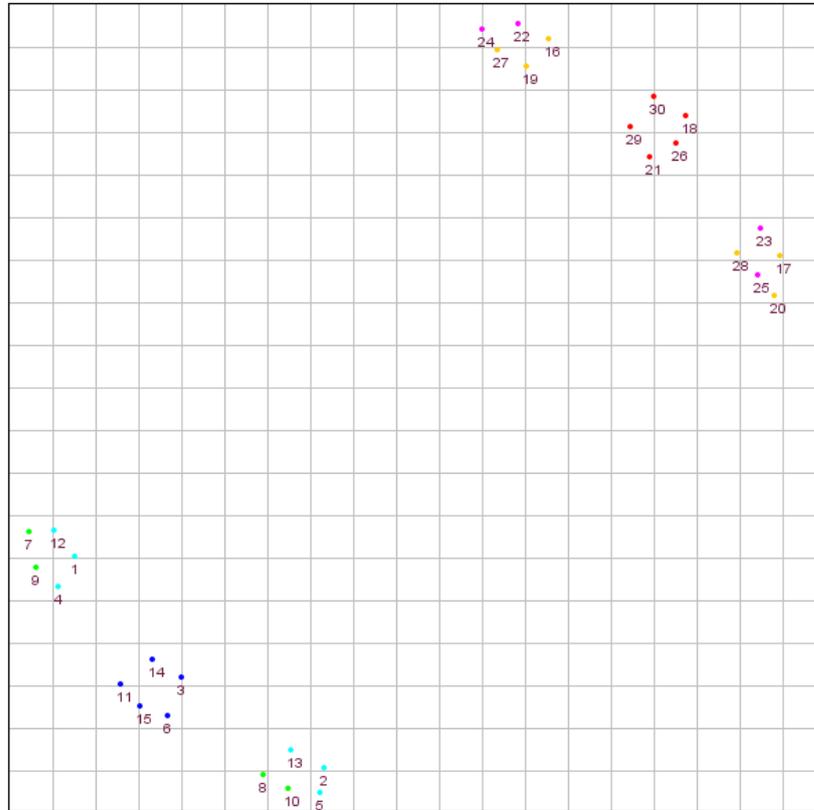


Figure 2.2: MDS plot of results from synthesized data. Highlighted are the six clusters found by hierarchical clustering.

Bearing this in mind even the strange looking results of k-means make sense (see figure 2.3), although the project is too small for decent statistics that is needed for k-means.

So why does MDS show 6 clusters? MDS tries to get relative distances for all point combinations right, which in two dimensions is rarely ever possible. Clustering on the other hand is concerned with grouping similar elements and does not care about the other ones. For that purpose a reduction of dimensions is not necessary, so we can expect clustering to be more accurate. As for the MDS plot: it looks exactly the way as predicted for the SMACOF algorithm. However, it does not necessarily represent clusters, due to the loss of higher dimensional information.

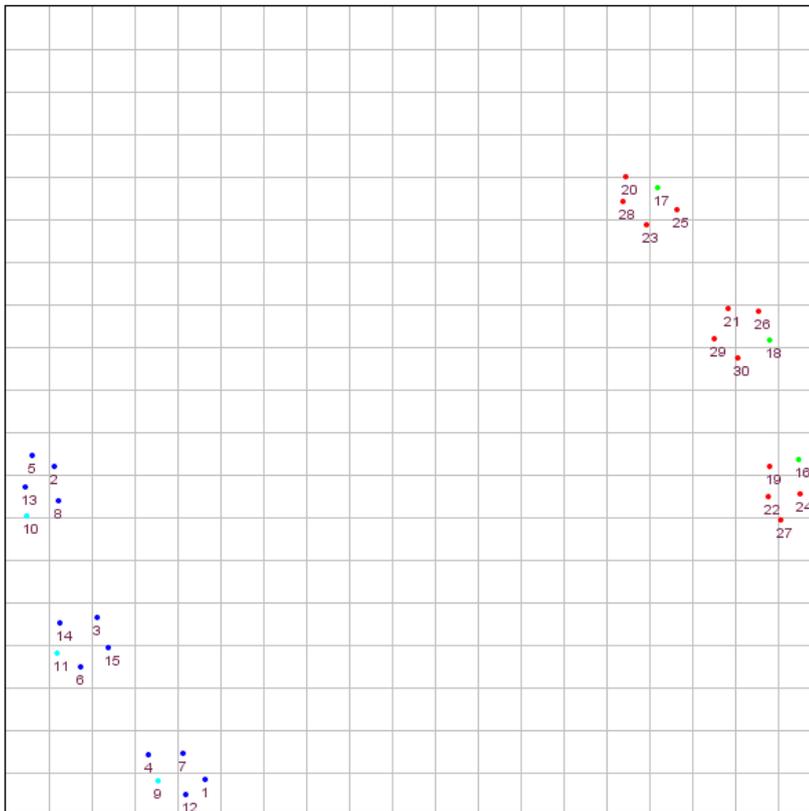


Figure 2.3: MDS plot of results from synthesized data. Highlighted are the four clusters found by k-means.



**Part II**  
**Users' Manual**



This Part provides a practical manual for ScenLab. It will explain how to get the Software setup and running. The Chapter 'How to use ScenLab' contains detailed information about the data input, processing and analysis. It is assumed that the user is familiar with the functionality of the algorithms and the terms used in Part I, particularly for the section on 'Data Processing'.

The last chapter 'Special Features' contains information about features of ScenLab you need not know about to use the software. Nonetheless, this information will help you to customize ScenLab.



# Chapter 3

## Getting started

### 3.1 Installing the Java Virtual Machine

ScenLab is programmed using Java. Java is a high level programming language provided by Sun Microsystems. The advantage of Java is that it is platform-independent, i.e. the same software works on different operating systems, such as Microsoft Windows, Linux distributions or MacOS. To enable your respective operating system to interpret Java you need to install the Java Virtual Machine on your Computer.

#### 3.1.1 Installation from CD

If you have access to the internet we recommend to proceed to section [3.1.2](#).

A version of the Java Virtual Machine for Microsoft Windows, MacOS and Linux is located on your ScenLab CD in the folder 'JavaVM'.

In this folder please choose the respective folder of your operating system, 'MSWin', 'MacOS' or 'Linux'.

Close all applications running on your computer and execute the file you find in the folder.

Please follow the instructions given by the automatic installer.

You might get a message that you already have a newer version of the Virtual Machine. In this case you may proceed to section [3.2](#).

#### 3.1.2 Downloading the current Virtual Machine

Sun Microsystems updates the Java Virtual Machine frequently.

Please visit <http://www.java.com> to download the current Virtual Machine, and follow the instructions given by Sun Microsystems. The web site should detect your operating system automatically and guide you to the right installation.

You might get a message that the most recent Virtual Machine is already installed on your system. In this case you may proceed to section [3.2](#).

## 3.2 Installing ScenLab

Depending on the operating system ScenLab can be installed using the installers distributed. There is an alternative way to install ScenLab which works on all operating systems (see 3.2.4).

### 3.2.1 Installing under Microsoft Windows

To install ScenLab on computers running Microsoft Windows double-click on the file called `installScenLab.msi` and follow the instructions on the screen.

You will need to remember in which directory ScenLab is installed in in order to be able to clear ScenLab (see ??).

The installer will setup ScenLab on your computer and place an entry called `ScenLab` in the Windows Start menu.

### 3.2.2 Installing under Mac OS X

To install ScenLab under Mac OS X you will need the file `ScenLab.dmg`. This is a disk image, that is it mounts like a memory stick for example, that contains the application files.

Move these files to your `Applications` folder (for german users `Anwendungen`). Alternatively you can move them to any other location on your `Macintosh HD`.

Store the PDF `ScenLab Users Manual` in the same directory.

Unmount the `ScenLab.dmg` disk image file.

### 3.2.3 Installing under Linux

Unzip `ScenLab.tar.gz` to your computer. In the resulting folder is a shell script called `install.sh`. Execute this script using a command shell with the command `./install.sh`.

### 3.2.4 Alternative installation (all operating systems)

Since ScenLab itself does not have to be registered on your machine - all the interaction with the operating system is done via the Virtual Machine - it is sufficient to copy ScenLab to a directory on your computers hard drive. In order to do this just copy the folder `ScenLab` from the CD to a folder of choice on your hard drive.

This folder contains a file called `ScenLab.jar`. This is an executable Java archive. For convenience you might want to setup a short-cut from your desktop to this file, but this is optional.

## 3.3 Clearing ScenLab\*

In order to run the full edition of ScenLab you need a licence key. This is a file called `licence.dat`. Usually you will receive this file separately by e-mail or on an information carrier. Save this file in the folder `ScenLab` on your hard drive (compare 3.2).

Please keep a copy of this file in a safe place. ScenLab will not run without this file. Please contact `info@evolveit.biz` in case your licence key is lost or corrupted or if you are unsure that you are using the right key.

### 3.3.1 Clearing under Microsoft Windows

Copy the file `licence.dat` to the directory ScenLab is installed in. By default this is the the subfolder `evolveIT/ScenLab` in the `Program Files` folder (for German users this is the `Programme` folder).

### 3.3.2 Clearing under Mac OS X

This section applies only to users that installed ScenLab in the way described in [3.2.2](#).

For your convenience the program `ClearScenLab` (provided with your distribution) will clear ScenLab for you, provided that you have a valid licence file.

Run `ClearScenLab` and give it the location of your ScenLab bundle (the `ScenLab.app` folder, note that the extension `".app"` is usually suppressed by Mac OS X and the folder behaves like a single file) and the licence file in the respective dialogs.

## 3.4 Starting ScenLab

If you used the alternative installation method, to start ScenLab all you have to do is to double-click on the file `ScenLab.jar` or its short-cut (see [3.2](#)). This will start the main window of ScenLab (see [Figure 4.1](#)). If you used the specialized installation methods ScenLab can be started like any other program on your computer.

Depending on your operating system and Virtual Machine the above procedure might not work. In this case there are two ways to get ScenLab running by starting the packed or unpacked application using the ‘Command Prompt’

**Using MS Windows.** You can start ScenLab from the ‘Command Prompt’. In the command prompt switch to the directory where the `ScenLab.jar` file is located. Type the following in your command prompt:  
`java -jar ScenLab.jar.`

**Using other operating systems.** You can start ScenLab using the command shell. Open the shell you want to use and switch to the directory where the `ScenLab.jar` file is located. Type the following in your command shell:  
`java -jar ScenLab.jar.`



# Chapter 4

## How to use ScenLab

This chapter will explain all features of ScenLab and how to use them. ScenLab consists of three main parts: the data input, the data processing and the result output.

ScenLab has three main user interfaces as shown in Figure 4.1. The menu bar and the toolbar are for user input. The status bar shows information about the last actions performed by ScenLab, such as the status of the algorithms, export actions, etc.

### 4.1 Data input in ScenLab

#### 4.1.1 Starting a new project

When ScenLab is opened it automatically starts with a blank project, i.e. you may begin data input immediately.

To start a new project at any other time one can either click on the **New Project** icon in the toolbar or choose **File>New Project** from the menu bar (see Figure 4.1). This will start a blank project.

#### 4.1.2 Opening an existing project

To open an existing project click on the **Open Project** icon on the toolbar or choose **File>Open Project** from the menu bar. Both actions will call the open dialog. Here you can pick the project that you want to open.

##### **Note 4.1.1**

*The project files of ScenLab end with the file extension `.smp` for scenario management project. Only these files will be recognized as project files by ScenLab.*

#### 4.1.3 Saving a project

To save a project click on the **Save** icon at the toolbar or choose **File>Save Project** from the menu bar. If you are working on a project that has already been saved before, it will be saved automatically. If the project is a new project you will be asked to enter a file name and choose a location to save the project.

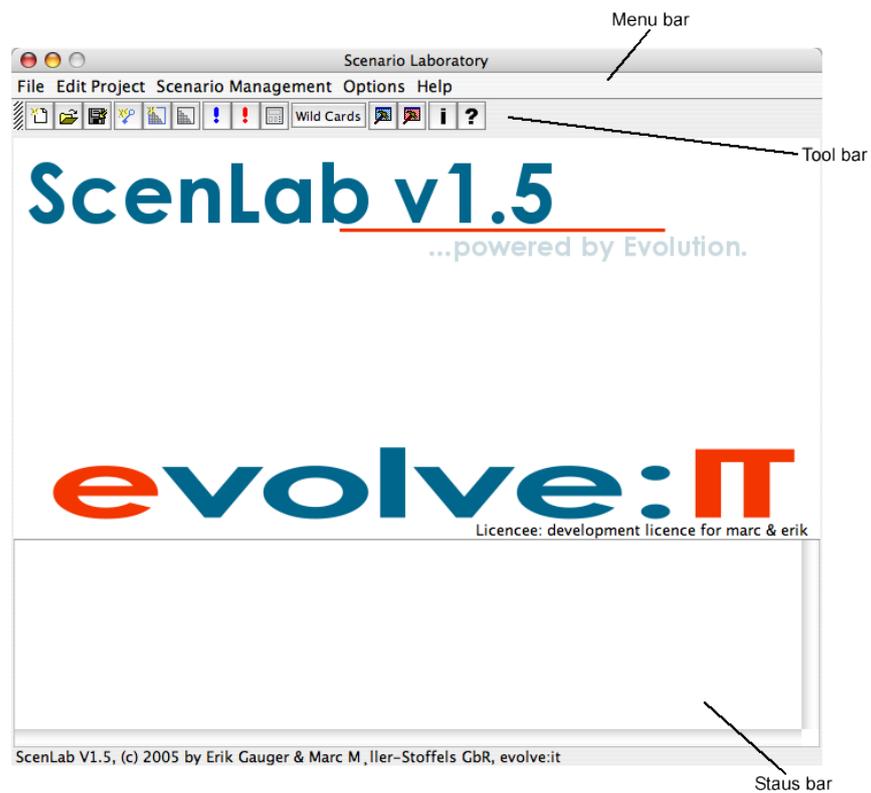


Figure 4.1: After starting ScenLab the main window is shown.

If you would like to save an existing project under a different file name or at a different location choose **File>Save Project As...** from the menu bar. This will open a save dialog. Here you can choose file name and location.

**Note 4.1.2**

*ScenLab will save all project files with the extension .smp. If you use a different file extension ScenLab will append .smp after this extension to make sure that the file will be readable for ScenLab.*

*If you do not use a file extension ScenLab will automatically append .smp to file name as well.*

**Note 4.1.3**

*Even though ScenLab is a very stable software it is strongly recommended that you save your work frequently to avoid the partial loss of data. Always save your project before you run the algorithms, because in the event of mishandling algorithms might stall.*

#### 4.1.4 Setting the project parameters

ScenLab allows the user to edit certain project parameters. The range of the paired consistency values can be edited, i.e one can choose the upper and lower limit for the paired consistency values.

**Note 4.1.4**

*The paired consistency value range should be set before entering values in the consistency matrix. This is because changing (especially narrowing) the range later might prevent the algorithms from running due to invalid inputs in the consistency matrix, i.e. out of range paired consistency values.*

The threshold values for total and partial inconsistencies can be edited as well. These values set the threshold below which a paired consistency value will be accounted for as partial or total inconsistency respectively.

**Note 4.1.5**

*ScenLab does not evaluate if your choice of threshold values makes sense from a methodical point of view. Nonetheless, it checks if the threshold values make sense for the further processing of data. Obviously, a threshold value for partial inconsistencies that is lower than the one for total inconsistencies does not make sense, as well as a threshold value that is out of range of the paired consistency values. In these cases you will be asked to change your choice of values.*

For convenience one can choose the default value for the unedited consistency matrix. It can be useful to set this value to an out of range value to mark unedited fields of the consistency matrix. In this case all fields must be set to a value in range before being able to run the algorithms.

You can reset all values by clicking on the **Default** button. Clicking **Okay** will save the changes made to the project parameters and close the dialog. Clicking **Cancel** will discard the changes made and close the dialog.

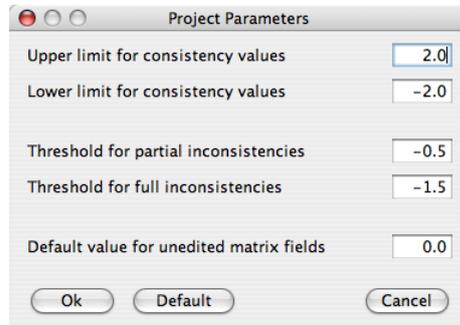


Figure 4.2: Project parameters dialog, default settings.

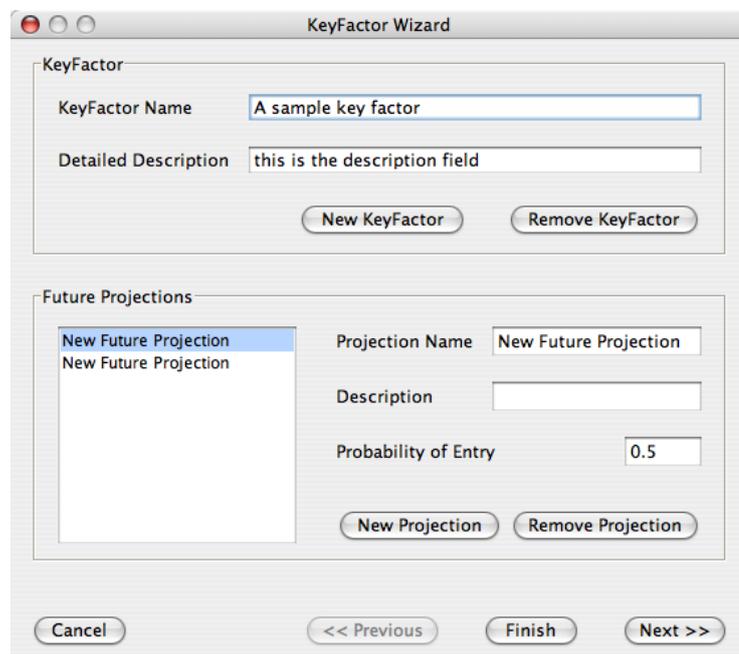


Figure 4.3: The key factor input wizard.

### 4.1.5 The key factor input wizard

**Note 4.1.6**

For ScenLab v1.7 LE the maximal number of key factors plus wild cards cannot exceed ten.

To input new key factors click on the **Open KeyFactor Wizard** icon on the toolbar or go to **Edit Project>KeyFactor InputWizard...**

Figure 4.3 shows the key factor input wizard. To add a new key factor click on the **New KeyFactor** button. This will add a blank key factor with one blank future projection. Now you can assign a name and add a description to the key factor and the future projection. If you want to add more future projections click on the **New Projection** button. This will add another blank future projection to the respective key factor.

**Note 4.1.7**

Before ScenLab proceeds to the next key factor it will check if the plausibility values assigned to the future projections sum up to 1. If this is not the case ScenLab will suggest to use an equal distribution of plausibility values over the future projections of the key factor. Using this option means that a previous input in the plausibility value field will be overwritten.

To remove a key factor use the button **Remove KeyFactor**. This will remove the key factor and all its future projections. To remove a single future projection use the **Remove Projection** button. This will remove the picked future projection.

You can browse through the key factors using the **Next>>** and **<<Previous** button.

To leave the key factor wizard and keep the input click the **Finish** button. Your work will be saved temporarily until you save the project (see 4.1.3).

To leave the wizard without saving the input use the **Cancel** button.

**Note 4.1.8**

Clicking the **Cancel** button will discard **all** new key factors, i.e. it will not only discard the most recently added key factor but all input added after opening the key factor wizard the last time.

### 4.1.6 Editing key factors

You can edit the list of existing key factors using the key factor input wizard (see 4.1.5). To conveniently browse the list of existing key factors open **Edit Project>Edit KeyFactors...** from the menu bar. This will open the key factor selection dialog. A list of all existing key factors will be displayed. To edit a key factor click on it in the list, then click on the **Select** button. This will open the key factor input wizard displaying the selected key factor.

To leave the key factor selection dialog click on the **Close** button.

### 4.1.7 Creating and editing wild cards

To create a new wild card or to edit existing wild cards open **Edit Project>Edit Wild Cards...** This will open the wild card selection dialogue.

To create a new wild card push the **New** button. To edit an existing wild card pick the desired wild card from the list and push the **Select** button. Both actions will open the wild card editor. Here you can enter or change the wild cards name, description and the plausibility value for the occurrence of this wild card. To save your changes push the **Finish** button. To discard your changes push the **Cancel** button. Both actions will close the wild card editor and return you to the wild card selection dialogue.

To delete a wild card from the list pick the desired wild card from the list and push the **Remove** button. This will permanently delete the wild card.

Leave the wild card selection dialogue by pushing the **Close** button.

### 4.1.8 The consistency matrix wizard

The consistency matrix wizard allows the user to conveniently fill in the sub-matrixes of the consistency matrix.

To open the wizard click the **Open Consistency Matrix Wizard** icon from the toolbar or chose **Edit Project>Consistency InputWizard** from the menu bar. This will open the consistency matrix wizard (see Figure 4.4) displaying the first combination of key factors and the sub-matrix of their respective future projections.

To fill in paired consistency values pick the respective field in the matrix and enter the chosen value.

To browse through the sub-matrixes use the **Next>>** and **<<Previous** buttons.

To leave the consistency matrix wizard and keep the input click the **Finish** button. Your work will be saved temporarily until you save the project (see 4.1.3).

To leave the wizard without saving the input use the **Cancel** button.

#### Note 4.1.9

Clicking the **Cancel** button will discard **all** newly entered values of the current sub-matrix, i.e. different to the key factor wizard pressing the **Next>>** and **<<Previous** button will save the current sub-matrix.

#### Note 4.1.10

After entering the last value in a sub-matrix you need to hit 'Enter' in order to save this entry. If you do not do this the entry will be lost once you press **Next>>**, **Finish** or **<<Previous**.

### 4.1.9 Editing the consistency matrix

To display and edit the full consistency matrix click the **Open entire Consistency Matrix** icon from the toolbar or chose **Edit Project>Edit Consistency Matrix** from the menu bar. This will open the full consistency matrix editor as shown in Figure 4.5.

To get information as to which future projection pair a consistency value is assigned point with the mouse to the consistency value in question, a 'tool tip' will be displayed.

To edit paired consistency values in the full consistency matrix single-click on them.

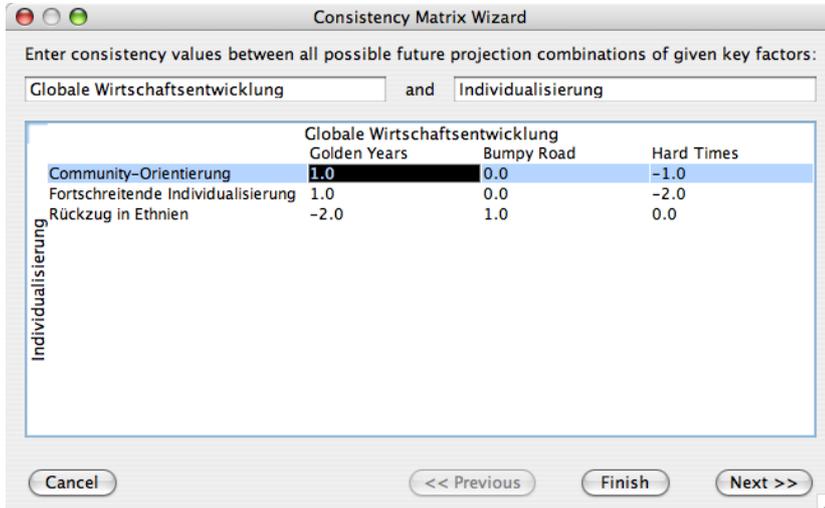


Figure 4.4: The consistency matrix wizard.

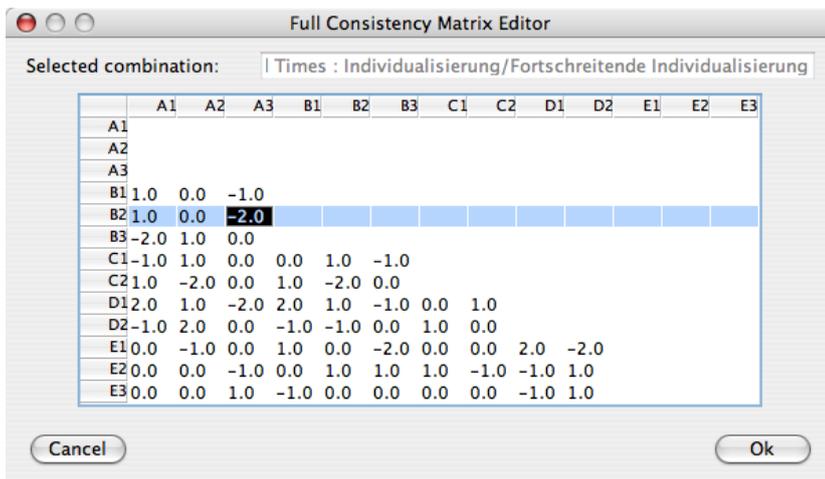


Figure 4.5: The full consistency matrix editor.

To leave the consistency matrix and keep the input click the **Ok** button. Your work will be saved temporarily until you save the project (see 4.1.3).

To leave the wizard without saving the input use the **Cancel** button.

**Note 4.1.11**

Clicking the **Cancel** button will discard **all** newly entered values, i.e. it will not only discard the most recently added paired consistency value but all input added after opening the consistency matrix editor the last time.

**Note 4.1.12**

After entering the last value in a sub-matrix you need to hit 'Enter' in order to save this entry. If you do not do this the entry will be lost once you press **Ok**.

#### 4.1.10 Merging input data from multiple files\*

The ScenLab Client (see 5.1) feature of ScenLab enables multiple users to fill in the same consistency matrix, setup by a master user, independently.

We recommend the following procedure:

After it is decided which key factors and future projections will be in the project a master file should be set up. This can only be done using the full ScenLab distribution. The consistency matrix of the master project should either be filled in or set to the mean value of the range of paired consistency values (see 4.1.4).

After saving the master project it can be distributed to the users to fill in the consistency matrix independently using either the full ScenLab distribution or, preferably, the ScenLab Client (see section 5.1. The advantage of using the client is that the users are not able to change the key factors and future projections and global project parameters from the master project.

Once the filled in client projects are returned to the project manager the files can be merged with the master project. This is done by loading the master project like any other project (see section 4.1.2). Then choose **Scenario Mangement>Merge Consistency Matrix...** from the menu bar. This will open a dialog where you can choose the client project to merge with the master project. By pressing the **Open** button the project will be merged with the master project. To add another client project repeat this action.

Save the project under a new name or under the name of the master project, but note that this will overwrite the consistency matrix of the master project permanently.

**Note 4.1.13**

When merging two consistency matrixes ScenLab will calculate the arithmetic mean of each paired consistency value field of the matrix.

ScenLab keeps track how many projects have been merged with the master project in order to calculate the proper arithmetic mean of all merged files.

To weight a project stronger than others simply merge it several times with the master project. That is to weight one project five times merge it with the master project five times.

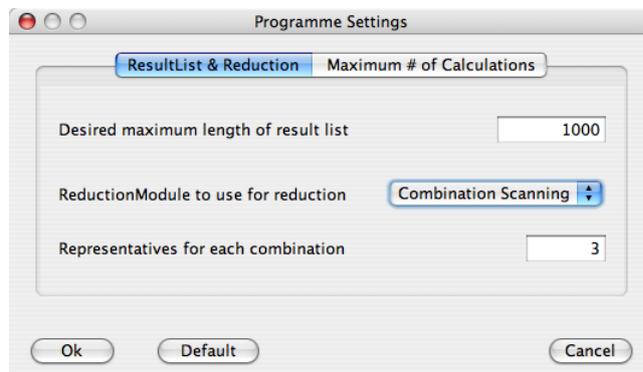


Figure 4.6: The program settings dialog.

#### 4.1.11 Exporting the consistency matrix to CSV\*

The consistency matrix of a project can be exported to a comma separated values (.csv) file. These files can be imported into Microsoft Excel or similar software.

To export a consistency matrix to the CSV-format choose Scenario Management>Export Consistency Matrix to CSV from the menu bar. This will open a dialog where you can choose the location and file name for the CSV file. By pressing the Save button the currently loaded consistency matrix will be converted and exported to CSV.

## 4.2 Data processing - setting up and running the algorithms

ScenLab allows the user to adjust the settings of both the conventional and the genetic algorithm to your specific project. This section describes how this is done and how changing the settings might affect the results you will receive. It is highly recommended that you read Chapter 2 before working with the settings of the algorithms.

### 4.2.1 Editing the program parameters

There are some global settings you can adjust to your specific projects. If you open the program parameters dialog by choosing Options>Program settings.. from the menu bar the window shown in Figure 4.6 is displayed. Alternatively the Dialog opens by pressing the Display Programme Settings Dialog button on the toolbar. It contains two index cards. The card 'Result List and Reduction' controls settings for the conventional algorithm **only**. The card 'Maximum Number of Calculations' controls global parameters.

**The desired maximum length of the result list.** ScenLab will try to keep the length of the result list obtained by the conventional algorithm below this value. That is, if you use the 'Trivial Reduction' module this is the length where

the list is cut. If you are using the ‘Scanning’ modules ScenLab will adjust the number of representatives until the length of the result list is close to the chosen value.

**Reduction Module to use for reduction.** This drop down menu lets you choose which reduction module is used. For further information refer to Chapter 2.

**Representatives for each combination.** This field controls the desired number of representatives for the ‘Scaling’ reduction modules. Note that only integer values are valid. If the choice of this value results in an excessively long result list, ScenLab will re-scale it dynamically to meet the desired length.

**Maximum Number of Calculations.** You can pick a maximum number of calculations for each algorithm. Before running the algorithms ScenLab evaluates the number of calculations needed to run the algorithm. If this number exceeds the maximum number of calculations given for the respective algorithm you will be warned before running the algorithm (see note 4.2.1). The default value is set to 1,000,000. This is the amount of calculations an average computer can handle in a convenient amount of time. As computers develop this value might rise significantly.

Pressing the **Default** button will reset the current index card to the default values hard coded in ScenLab.

Pressing the **Ok** button will save the changes made.

#### 4.2.2 Including or excluding wild cards from the algorithm

ScenLab allows the user to run the algorithms with or without wild cards. To include wild cards into the analysis the toggle button **Wild Cards** on the toolbar needs to be pushed or the menu entry

**Scenario Management>Include Wild Cards in Analysis** needs to be selected. Selection is visualized by a check mark left to the menu entry.

#### 4.2.3 Running the standard algorithm

To run the standard algorithms for the consistency analysis, including the reduction module, press the blue exclamation mark on the toolbar or choose **Scenario Management>Run Standard Analysis** from the menu bar. ScenLab will then analyze the current project and run the algorithm.

##### Note 4.2.1

There is a situation where the algorithm might not run right away. If ScenLab detects that the number of calculations needed exceeds the maximum number of calculations the user will be warned. It is possible to ignore this warning at the risk of stalling the computer.

If the project requires only a few more calculations above the designated limit it will simply take a while longer until it is done.

However, if the project requires lots of calculations above the designated limit it is very likely that errors occur and the software stalls. These errors are

not caused by ScenLab itself - theoretically the algorithm could do the required calculations. However, trying to solve a big problem will result in a shortage of the memory allocated for the `Java Virtual Machine`. This is the cause of the failure, not the software itself. Depending on the central processing unit (CPU), and the read and write memory (RAM) of the machine, as well as the version of the Virtual Machine, projects containing more than 20 key factors with about 3 future projections each will cause these kind of problems.

#### 4.2.4 Configuring the genetic algorithm\*

There are several parameters of the genetic algorithm that can be adjusted. The GenScen configuration dialog can be opened by pressing the `Display GenScen settings dialog` button on the toolbar, or by choosing `Options>GenScen Configuration...` from the menu bar. The parameters and their usage are explained in detail below.

**Weighting between consistency and plausibility.** The slider allows you to choose the fitness criterion of projection bundles (individuals) to either be the consistency value, the plausibility value or a weighted mix of both. That is, by setting the slider to the middle an even mix of both values will be used to evaluate the fitness of the projection bundles.

**Searching for highest Robustness.** Checking this tick box will set the fitness criterion to be the robustness value. That is, the fitness of projection bundles will be evaluated based on their robustness. If this option is picked the settings in the slider ‘Weighting between consistency and plausibility’ will not be used.

**Number of concurrent solution candidates.** Using this field the user can set the population size to a desired value.

Setting this value has a strong effect on the length of the result list, i.e. the result list will never be longer than this value, but usually will be much shorter. If your result list does not contain enough projection bundles you might want to raise this number.

**Number of runs of the evolutionary algorithm.** This field sets the number of generations (runs) GenScen will run.

The longer the algorithms runs the higher the overall fitness of the population (projection bundles) will be. Usually 500 generations are more than sufficient.

The number of runs has a weak effect on the length of the result list. The more runs the shorter the list as more and more individuals evolve to the optimum. This might not be the desired effect, because too many weaker (but still good) individuals might be lost.

**Mutation rate.** This field sets the mutation value that is defined as the probability that a mutation will occur in an individual.

The higher the mutation rate, the higher the random element of the search for fit individuals is. This might result in a slightly lower average fitness of the population, but in a higher variety of the results.

*The higher the mutation rate is, the longer the result list is. Still the mutation rate should not exceed about 0.05, because above this value too many good solutions are destroyed by mutations, i.e. the search becomes nearly completely random.*

**Crossover rate.** *This field sets the probability for crossover of two individuals.*

*The higher the crossover rate the higher the exchange of information between the individuals is. In other words, below a certain threshold of the crossover probability (about 0.35) individuals can interchange good sequences of their chromosomes in order to form fitter individuals. Above the threshold the information of the population is mixed up too much and good individuals are rather destroyed by the crossover process.*

**Selection Manager.** *There are two different mechanism for the user to choose that select the individuals that will proceed to the next generation. The selection mechanisms are tournament selection and roulette wheel selection.*

**Tournament Selection.** *This selection mechanism picks a certain percentage of the population randomly. The chosen individuals will compete in a tournament. That is, the fittest out of the group of chosen individuals will proceed to the next generation. This process is repeated until the new generation has the size of the old one.*

*This selection mechanism is the most useful for searching a huge search space. It prevents strong individuals from getting very dominant too fast. Because not every single tournament will contain one of the strongest individuals weaker one get the chance to proceed as well. This is desirable because they might carry important information.*

**Roulette Wheel Selection.** *The software sets up a virtual roulette wheel. Each slot represents one individual. The slots of the roulette wheel are not distributed evenly but weighted by the respective individuals fitness, i.e. the slot representing a strong individual will be bigger than the one for a weak individual. The roulette wheel is 'spun' as many times as the population has individuals. The bigger a slot the higher the chance that the ball will fall into it.*

*Using this kind of selection mechanism strong individuals have high advantage to proceed to the next generation. This will lead to a fast and strong convergence of the algorithm. But there is a high risk that it might get stuck in a sub-optimal space. That is, the algorithm might find very strong individuals, but miss the strongest because they differ too strongly from the dominant ones.*

**Use elitism and number of elitists.** *Selecting the check box will set the algorithm to use elitism with the given number of elitists.*

*When using the tournament selection it is especially important to ensure that the fittest individuals proceed to the next generation no-matter-what. Using elitism the algorithm will scan the population for the fittest individuals and copy them to the next generation before running the selection mechanism.*

The number of elitists should not exceed three, because this would cause an undesirable dominance of the fittest individuals.

All these parameters interact with each other and the characteristics of the project. Therefore it has proven useful to 'play' with different settings and compare their results.

**Use ReArrange.** The re-arrange option is a feature that alters the performance of the crossover operator. Using it will result in a longer result list.

#### 4.2.5 Saving the GenScen configuration\*

The current GenScen setting can be saved. This feature is very useful in order to run the genetic algorithm using several different settings without having to change them by hand every time.

To save the current settings choose `Options>Safe GenScen Config...` from the menu bar. This will open a file save dialog.

##### Note 4.2.2

The file extension for genetic algorithm settings is `.gas`. If this extension is not chosen by the user ScenLab will append it automatically to the file name.

#### 4.2.6 Loading a GenScen configuration\*

To load an existing GenScen Configuration choose `Options>Load GenScen Config...` from the menu bar. This will open a file open dialog where the user can select the desired settings file.

##### Note 4.2.3

Genetic algorithm files need to have the file extension `.gas`.

#### 4.2.7 Running the genetic algorithm\*

To run the genetic algorithm press the red exclamation mark from the toolbar or choose

`Scenario Management>Run GenScen Analysis` from the menu bar. This will start the algorithm.

A progress window will open that shows the average fitness of the population and the fitness value of the best individual so far.

Depending on the size of the project and the chosen settings it might take a while until the algorithm terminates. The algorithm can be stopped at anytime by pressing the `Stop Algorithm` button in the progress window.

### 4.3 The results - analysis and export

The third feature of ScenLab is the result viewer. It displays the results calculated by the algorithms and supports several ways in order to analyze the data.

The screenshot shows the 'Result Viewer' window with a table of results. The table has columns: Id #, Short Name, Consistency, Plausibility, # plnC, Robustness, Avg. Cons., Cluster #, and W.C. Row 3 is highlighted in blue.

Id #	Short Name	Consistency	Plausibility	# plnC	Robustness	Avg. Cons.	Cluster #	W.C.
1	A483C3D5E1F3G1H3	39	1,526E-5	0	0,9210	1,3929	0	
2	A483C3D4E1F3G1H3	39	1,526E-5	1	0,6512	1,3929	0	
3	A483C3D5E1F4G1H3	37	1,526E-5	0	0,9112	1,3214	0	
4	A483C3D4E1F4G1H3	36	1,526E-5	1	0,6409	1,2857	0	
5	A483C3D4E1F3G1H2	36	1,526E-5	1	0,6409	1,2857	0	
6	A184C1D2E1F4G4H2	36	1,526E-5	1	0,6409	1,2857	0	
7	A183C3D5E1F4G1H3	36	1,526E-5	0	0,9063	1,2857	0	
8	A183C3D5E1F3G1H3	36	1,526E-5	0	0,9063	1,2857	0	
9	A484C3D5E1F3G1H3	35	1,526E-5	0	0,9014	1,2500	0	
10	A483C4D4E1F3G4H3	35	1,526E-5	0	0,9014	1,2500	0	
11	A483C4D4E1F3G1H3	35	1,526E-5	1	0,6374	1,2500	0	
12	A483C3D5E1F3G1H2	35	1,526E-5	0	0,9014	1,2500	0	
13	A483C3D4E1F2G1H3	35	1,526E-5	2	0,5204	1,2500	0	
14	A483C3D2E1F3G1H3	35	1,526E-5	1	0,6374	1,2500	0	
15	A482C3D5E1F3G1H3	35	1,526E-5	1	0,6374	1,2500	0	
16	A483C3D4E1F3G1H2	35	1,526E-5	1	0,6374	1,2500	0	

Below the table, there are sections for 'Selection Details' (Full Text Name, Description Text, Scenario Title & Comment) and 'Analysis Tools' (MDS Plot, Clustering, Morph. Matrix, Distribution Plot, Info, Truncate List, Include Wild Cards, Show Selection Basket). Buttons for 'Open Results', 'Save Results', 'Export Results', and 'Close' are at the bottom.

Figure 4.7: The Result Viewer.

### 4.3.1 The result viewer

Figure 4.7 shows the main window of the result viewer with loaded results. The first column of the table displays the short names of the calculated projection bundles.

#### Note 4.3.1

The short name of a projection bundle consists of letters for the respective key factors as they are sorted in the list of key factors (see 4.1.6) starting with the letter 'A' and integers for the respective future projection of a key factor contained in the projection bundle starting with the integer '0'. 'A1B0...' means that this projection bundle contains the second future projection of key factor 'A' and the first future projection of key factor 'B'.

Highlighting a row of the table will display the long names of the key factors and the future projections contained in that projection bundle in the Full Text Name field. Highlighting a key factor or future projection in this field will display the description (if any was given) in the Description Text field.

The Scenario Title and Comment field can be used to add comments about the selected projection bundle. These comments will be saved once the result list is saved (see section 4.3.2).

The other columns of the table show the values that characterize the projection bundles. Clicking on the header of any of the columns will sort the table

by the values given in this column in descending order. Clicking twice on the header will sort the table according to the respective value in ascending order.

**Note 4.3.2**

To sort the result table by more than one criterion keep the **Ctrl**-key pressed while choosing the further sorting criteria.

Pressing the **Wild Cards** button will show/hide the projection bundles from the list that contain wild cards with the future projection "Wild Card occurs". Pushing the button will only affect the result list if wild cards were included in the algorithms (see section 4.2.2).

**Note 4.3.3**

All plotting operations, MDS, Clustering and Distribution Plot, will operate on the current result list. That is, to include projection bundles containing the future projection "Wild Card occurs" these projection bundles need to be shown in the result list.

Pressing the **Info** button will show additional information to the results such as the length of the result list, which algorithm was used and the overall number of projection bundles.

### 4.3.2 Saving results

The result viewer opens automatically after each run of any of the algorithms. However, the displayed results will not be saved automatically.

To save the currently displayed result list press the **Save Results** button. This will open a dialog where you can choose the location and file name for the result file. By pressing the **Save** button the results will be saved.

**Note 4.3.4**

Result files will have the extension **.smr** for scenario management results. If this extension is not given by the user ScenLab will append it automatically.

### 4.3.3 Loading existing results

To open an empty result viewer choose **Scenario Management>Open Result Viewer** from the menu bar. To load an existing result list press the **Open Results** button. This will open a dialog where you can pick the location and the respective result file (file extension **.smr**) you want to load. Pressing the **Open** button will display the results in the result viewer.

### 4.3.4 Exporting results\*

ScenLab allows the user to export results to **HTML**, that is plain webpage, simple text or **XML** (extensible markup language).

To export your results from the result viewer press the **Export Results** button. This will open the export dialog. In this dialog the user can choose the format to export to and which features of the results to export.

Once the output format and the export features are chosen click on the **Export** button. This will open a file save dialog. Here the file name and location need to be chosen.

### **Export to HTML**

If the **Export to HTML** option is selected the export engine will create a webpage from the result list including the selected features.

#### **Note 4.3.5**

The created *HTML* file contains a reference to a cascading style sheet (*CSS*) with the file name `scenLabStd.css`. *ScenLab* stores this style sheet in the same folder that the *HTML* file is placed in. The style sheet is not vital to view the webpage, however without it being in the same folder the color and font information of the webpage is missing, that is the readability of the page is not as good.

The style sheet may be changed (e.g. by your website administrator) in order to change the look of the exported *HTML* file. Just note that the style sheets file name must not be changed.

### **Export to Text**

This option exports the results to simple text, that is to the *TXT* format. This format is the simplest text format and can be read by virtually any text editor.

### **Export to XML**

*XML* is a powerful markup language that allows convenient processing of data using *XSL* style sheets for example. The *XML* definitions necessary to write custom *XSL* style sheets, or to do other data processing, is given in section 5.2. Information about a *XSL* style sheet need to be included into the plain *XML* file after the export.

### **4.3.5 Clustering results\***

Pushing the **Clustering** button will open the clustering dialogue window. The user can choose the desired algorithm, hierarchical clustering or *k*-means clustering, by selecting the respective radio button.

It is possible to manually enter the desired number by un-checking the tick-box **Automatically scan for optimal number of clusters**. Or the user can enter a desired range of clusters, by leaving the tick-box checked.

Run the clustering by pushing the **Run Clustering** button. The progress of the algorithm will be shown in the **Clustering Progress** status bar.

#### **Note 4.3.6**

Running the clustering algorithms requires many calculations. Depending on the length of the result list and on the computational power of the computer used clustering can take a while.

#### **Note 4.3.7**

Running the clustering while the projection bundles containing wild cards are not displayed in the result list will overwrite any prior cluster numbers of these

projection bundles. This is, because these projection bundles will not be considered in the current clustering and therefore their old cluster numbers would be wrong.

Once the clustering is done the cluster plot can be opened by pushing the **Cluster plot** button.

The plot can be exported to the PNG format by pressing the **Export** button. This will open a dialog where you can choose location and file name of the PNG file. The plot will be exported as you see it, that is if dots in the plot are highlighted or if the clustering is visualized this will be visible in the exported PNG file as well.

#### 4.3.6 Running MDS on results\*

To run a multi dimensional scaling on the result data press the **MDS plot** button. This will open a dialog where you can adjust the parameters

**Stress convergence criterion** and **Maximum # of iterations**. Please refer to section 2.3 to learn more about these parameters. Pressing the **Run MDS** will start the calculations of the MDS.

Once the necessary calculations are done a plot will open displaying the MDS of the projection bundles. The numbers of the points refer to the projection bundles as ordered by the consistency values in the result list. By hovering the mouse over the points the short names of the respective projection bundle will be displayed.

#### **Note 4.3.8**

The MDS algorithm will generally run until it meets the stress convergence criterion. However, meeting this criterion is not always possible. In these cases, it will stop the calculations after it reaches the maximum number of iterations. Depending on the length of the result list and on how you choose the MDS parameters the calculations may require several minutes. You can cancel the process at any time by pressing the **Cancel MDS** button.

To visualize the cluster affiliation of the projection bundles tick the **Visualize clustering permanently** checkbox.

Clicking on one of the dots in the MDS plot will highlight the projection bundle in the result list and all other dots representing members of the same cluster.

The plot can be exported to the PNG format by pressing the **Export** button. This will open a dialog where you can choose location and file name of the PNG file. The plot will be exported as you see it, that is if dots in the plot are highlighted or if the clustering is visualized this will be visible in the exported PNG file as well.

#### 4.3.7 The distribution plot

The distribution plot shows the distribution of the characteristic features of the projection bundles in the result list in their respective ascending order. To open the distribution plot press the **Distribution plot** button.

It is possible to reduce the number of projection bundles plotted by using the slider.

The plot can be exported to the PNG format by pressing the **Export** button. This will open a dialog where you can choose location and file name of the PNG file.

**Note 4.3.9**

The average consistency is not plotted because the plot would look like the one for the consistency values, but re-scaled to the range of paired consistency values.

### 4.3.8 Truncating the result list

The result list can be shortened to a desired length by pressing the **Truncate List** button. This will open the truncation dialog.

The default setting for truncation is to truncate the list descending by consistency value, where wild cards are included in the truncation, if there are any present.

You can choose to truncate the result list sorted by plausibility or robustness instead of consistency and in ascending order, i.e. keeping the bottom of the list, instead of descending. Also the truncation dialog offers the option to remove all projection bundles that contain wild cards from the list, if there are any in the list.

Type the desired new length of the list in the text field. To perform the truncation hit the **Truncate** button. To discard the truncation dialog without making changes to the list press the **Cancel** button.

By choosing the **Truncate to selection in Selection Basket** option the list can be truncated to any manual selection of projection bundles, that is to the projections bundles that are contained in the Selection Basket (see 4.3.9) at the time of truncation.

**Note 4.3.10**

Truncation is a ir-reversible operation. Unless saved prior to truncation the truncated projection bundles will be lost permanently.

### 4.3.9 The Selection Basket

The Selection Basket feature allows the user to manually select a desired number of projection bundles for further truncation (see 4.3.8) of the list and for viewing the Morphological Matrix (see 4.3.10) with multiple entries.

To show the Selection Basket check the **Show Selection Basket** check box. Now arrange the **ResultViewer** window and the Selection Basket in a way that you can view both at once.

In order to select a projection bundle from the Result List single click on it and then drag it from the Result List to the Selection Basket.

To clear a projection bundle from the Selection Basket highlight it in the list in the Selection Basket by clicking on it. Then press the **Clear current** button.

To clear the whole basket press the **Clear basket** button. This will remove all projection bundles from the basket.

The buttons **Morph. Matrix** and **Morph. basket** are explained in section 4.3.10.

### 4.3.10 The Morphological Matrix

The Morphological Matrix is another convenience feature for the data analysis. ScenLab can either show the morphological Matrix for a single projection bundle or for multiple projection bundles.

It is possible to adjust the font size by using the slider **Font size** and the column width by selecting the respective radio button in the **Column layout** field.

#### Morphological Matrix for single projection bundles

There are four ways of invoking the view of the morphological matrix for a single projection bundle. i) by simply double-clicking on it in the **Result List**; ii) by double-clicking on it in the **Selection Basket**; iii) by selecting it using a single click in the **Selection Baskets** list and then pressing the **Morph. Matrix** button; iv) by selecting it in the result list and clicking on the **Morph. Matrix** button in the result viewer.

#### Morphological Matrix for multiple projection bundles

By clicking on the button **Morph. basket** in the **Selection Basket** the morphological matrix for all the projection bundles contained in the basket will be displayed.

If the multiple line make the text in the matrix unreadable click on the check box **Draw line before text**, this will improve the readability of the text.

The legend for the multiple entries colors is shown below the matrix. To improve the readability the comment of the projection bundle is used instead of the name, if it is longer than five characters.

The export of the morphological matrix to the PNG format is possible by clicking the **Export** button.

#### Note 4.3.11

It is a known problem that the Java Virtual Machines version 1.4.2 and lower do not render the text very well in some cases. If this is a problem on your computer it is advisable to download the current version of the JRE from [www.java.com](http://www.java.com).



## Chapter 5

# Special features\*

### 5.1 The ScenLab Client

*The ScenLab Client is a smaller version of the full ScenLab distribution. It enables users to edit the consistency matrix and to view result files. It does not allow the user to edit key factors, future projections, plausibility values and wild cards, and it does not contain the algorithms.*

*To install the ScenLab Client copy the folder ScenLabClient from the ScenLab client CD to any directory of your computer.*

*This folder contains a file called ScenLabClient.jar. This is an executable Java archive. For convenience you might want to setup a short-cut from your desktop to this file, but this optional.*

*To run the ScenLab client double-click on the ScenLabClient.jar or its short-cut.*

*Depending on your operating system and Virtual Machine the above procedure might not work. In this case there are two ways to get the ScenLab Client running.*

**Using MS Windows.** *You can start the ScenLab Client from the ‘Command Prompt’. Once in the command prompt, switch to the directory where the ScenLabClient.jar file is located. Type the following in your command prompt:*

```
java -jar ScenLabClient.jar.
```

**Using other operating systems.** *You can start the ScenLabClient using the command shell. Open the shell you want to use and switch to the directory where the ScenLabClient.jar file is located. Type the following in your command shell:*

```
java -jar ScenLabClient.jar.
```

*Chapters of this manual relevant for the ScenLab Client are: [4.1.2](#), [4.1.3](#), [4.1.8](#), [4.1.9](#), [4.1.11](#) and [4.3](#).*

### 5.2 XML definitions of the result export

*The hierarchy of the XML-tags is shown in [Table 5.1](#).*

## 5.3 Adjusting header icon and background to your corporate design

*ScenLab* is provided with a default background image and a default header icon. These images can be adjusted to your company's corporate design.

The background image needs to be a 700x350 PNG file named `WndImg.png`. The icon needs to be a 17x17 PNG file named `WndIcon.png`. Copy these files to the `ScenLab` folder or the folder where your `ScenLab.jar` file is located (this should be the `ScenLab` folder). The next time you run *ScenLab* the new images will be loaded automatically.

To reset *ScenLab* to the default images remove the files from the folder.

To do the same to the *ScenLab Client* replace 'ScenLab folder' and 'ScenLab.jar file' with 'ScenLabClient folder' and 'ScenLabClient.jar file'.

### 5.3.1 Adjusting the Look-and-Feel

If you start *ScenLab* using the command prompt (or shell) you can choose the Look-and-Feel. By default the Look-and-Feel is the systems Look-and-Feel. To use the cross-platform Look-and-Feel provided by Sun Microsystems use the following command: `java -jar ScenLab.jar cross` or `java smsapp.ScenLab cross`.

To do the same with the *ScenLab Client* replace 'ScenLab folder' and 'ScenLab.jar file' with 'ScenLabClient folder' and 'ScenLabClient.jar file'.

