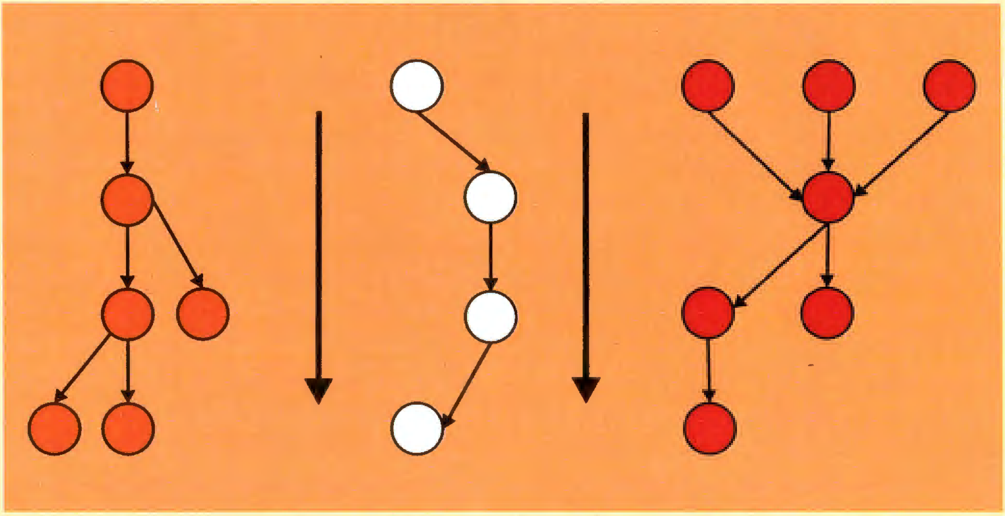
SYSTEMS RESEARCH INSTITUTE
POLISH ACADEMY OF SCIENCES

MULTICRITERIA ORDERING AND RANKING:
PARTIAL ORDERS, AMBIGUITIES AND APPLIED ISSUES



Jan W. Owsiński and Rainer Brüggemann
Editors

Warsaw 2008

SYSTEMS RESEARCH INSTITUTE
POLISH ACADEMY OF SCIENCES

MULTICRITERIA ORDERING AND RANKING:
PARTIAL ORDERS, AMBIGUITIES
AND APPLIED ISSUES

Jan W.Owsiński and Rainer Brüggemann
Editors

Warsaw 2008

This book is the outcome of the international workshop held in Warsaw in October 2008 within the premises of the Systems Research Institute. All papers were refereed and underwent appropriate modification in order to appear in the volume. The views contained in the papers are, however, not necessarily those officially held by the respective institutions involved, especially the Systems Research Institute of the Polish Academy of Sciences.

ISBN 83-894-7521-9
EAN 9788389475213

Technical editing and typesetting:
Jan W.Owsiński, Anna Gostyńska, Aneta M.Pielak

2001	Iffeldorf	Germany	K. Voigt and G. Welzl
2002	Roskilde	Denmark	P. Sørensen and L. Carlsen
2004	Bayreuth	Germany	H. Frank, A. Kerber and R. Brüggemann
2006	Verbania	Italy	R. Todeschini, M. Pavan
2008	Warsaw	Poland	J.W. Owsiński, R. Brüggemann

The scope of these workshops was primarily constituted by theoretical developments and practical applications of partially ordered sets in the fields of chemistry and environmental systems. One may characterize the application of partial order theory as an attempt of "let first the data speak", before a decision is derived. As such the application of partial order can be seen as a multivariate statistical approach to support decisions regarding complex systems. Up to now most applications were concerned with chemistry, biology, ecology, data availability and water management. The papers, devoted primarily to applications, associated with partial orders, contained in this volume, are the ones by S. Tsakovski and V. Simeonov: "*Hasse diagrams as exploratory tool in environmental data mining: a case study*", P. Annoni, E. Garofalo, R. Brüggemann: "*Partial order theory for assessing the sensitivity of planktonic algae to anthropogenic disturbances in regulated lakes*", M. Weckert, S. Gerstmann and H. Frank: "*Comparison of LCA results with those based on METEOR. Case study of refrigerants in mobile air-conditioning systems in passenger cars*", P. Annoni, M. Fattore and R. Brüggemann: "*Analysing the structure of poverty by fuzzy partial orders*" and L. Carlsen: "*Partial order ranking methodologies in CSR driven innovation*". As can be seen, not only we deal with a variety of application domains, but also a range of approaches, either directly related to Hasse-Diagram paradigm, or used in a complementary manner.

A separate stream has been made up of the work devoted to algorithmic and software tools for dealing with visualisation, analysis and aggregate ranking. These traditions are well apparent in the present volume, as represented by papers of R. Brüggemann, G. Restrepo and K. Voigt: "*Towards a new and advanced partial order software: PYHASSE*", and K. Voigt, R. Brüggemann, K.-W. Schramm, M. Kirchner: "*PYHASSE: A new software tool for partially ordered sets: ranking soil against needles*".

The studies on theoretical or methodological aspects are represented in the volume by the papers of K. De Loof, B. De Baets and H. De Meyer: "*Properties of mutual rank probabilities in partially ordered sets*", M. Rademaker, B. De Baets and H. De Meyer: "*Informative combination of multiple partial order relations*",

Hasse Diagram Techniques and Softwares

Multicriteria Ordering and Ranking: Partial Orders, Ambiguities and Applied Issues
Jan W. Owsiński and Rainer Brüggemann, Editors

Towards a New and Advanced Partial Order Program: Pyhasse

Rainer Brüggemann[*], **Guillermo Restrepo**[**] and **Kristina Voigt**[***]

[*] Department of Ecohydrology, Leibniz-Institute of Freshwater Ecology and Inland Fisheries, Müggelseedamm 310, Berlin, Germany (brg_home@web.de)

[**] Laboratorio de Quimica Teórica, Universidad de Pamplona, Pamplona, Colombia (grestrepo@unipamplona.edu.co),

Department of Mathematics, Universitat Bayreuth, D-95440 Bayreuth, Germany and Texas A&M University at Galveston, Galveston TX 77553, USA

[***] Helmholtz Zentrum München, German Research Center Environmental Health, Institute of Biomathematics and Biometry, Ingolstaedter Landstr., 85764 Neuherberg, Germany (kvoigt@helmholtz-muenchen.de)

The development of partial order tools applicable for case studies gets more and more enhanced. Although there is established software, like WHasse and additionally a new complex software RAPID is under development, a program system should be made available for everyone, which can be quickly adapted to any new theoretical aspect and can be helpful in application. The interpreter language PYTHON provides an ideal basis for this aim. It is an object-oriented language, available on multiple platforms (hardware systems) and multiple operating systems. PYTHON, in cooperation with the graphical user interface programming tool Tkinter (of Tk/Tcl), allows quick development and testing. Here the current version of PyHasse is shown. PyHasse, written in PYTHON, is still under development and is considered as a test version. Not all its features can here be demonstrated. In this paper we concentrate on some selected ones. PyHasse will be applied in order to model toxicities of chemicals by topological indicators and to perform a sensitivity study in a ranking study of pollution in Baden-Württemberg, a south-western state in Germany.

Keywords: partial order theory, software PYHASSE, environmental chemicals

1. Introduction

In the last decades, new tools to apply partial orders in application studies were developed. Furthermore, in view of the increasing community of scientists applying partial orders or deriving new concepts, which are of relevance for applications, there is a need for new software, which is

- easily applicable
- freely available and which
- can be rather quickly extended.

Taking into account that there is already good software (see for an overview: Halfon, 2006) and that new software is under development, like RAPID and that almost any lab has its own software (for example based on JAVA) the question arises, whether a new software is needed. We think: yes, indeed! The reason is that big and comfortable software needs some time for development, it is costintensive and may quickly be out of date. Hence, there is a need of software between lab-specific software and the 'great solutions' like WHasse (Brüggemann et al., 1999a), or PRORANK (Pudenz, 2005) and - in the future - RAPID. One may consider PyHasse as a 'rapid prototyping' software, where concepts can be tested and where anybody can quickly modify the software for own purposes.

Here a program system is shown, which is still under development (by 'learning by doing'), considered as a test version and which is certainly not following the lines of professional software development: PyHasse. The name is selected in analogy to WHasse, 'Py' refers to the programming language PYTHON. Besides the central platform, called 'pyhassemenue2', there are 17 programs, developed for specific tasks. Here we cannot explain all of the 17 programs, however, the interested reader is referred to the contributions of Annoni et al. and Voigt and Brüggemann, all in this volume.

We will give an introduction to the general principles behind PyHasse and explain some programs more specifically. We do not intend to provide explanation of the theoretical background. Specific references can be found in Table 1, where the single programs are explained. For general information the books by Trotter (1992), Davey and Priestley (1990), and Stanley (1986) are recommended.

In order not to give a systematic and boring description, we pose and address two questions:

(1) How to model toxicity of seven chemicals by topological indicators (TI)?

(2) How sensitive are polluted regions with respect to the attributes used to rank them?

The paper is organised as follows: In Section 2 an overview of PyHasse is given. In Section 3, question (1) will be answered, together with a sidestep on chemical structures and toxicity. In Section 4 question (2) is briefly discussed and in Section 5 some problems and possible future developments are outlined.

2. General information about PyHasse

2.1 PYTHON

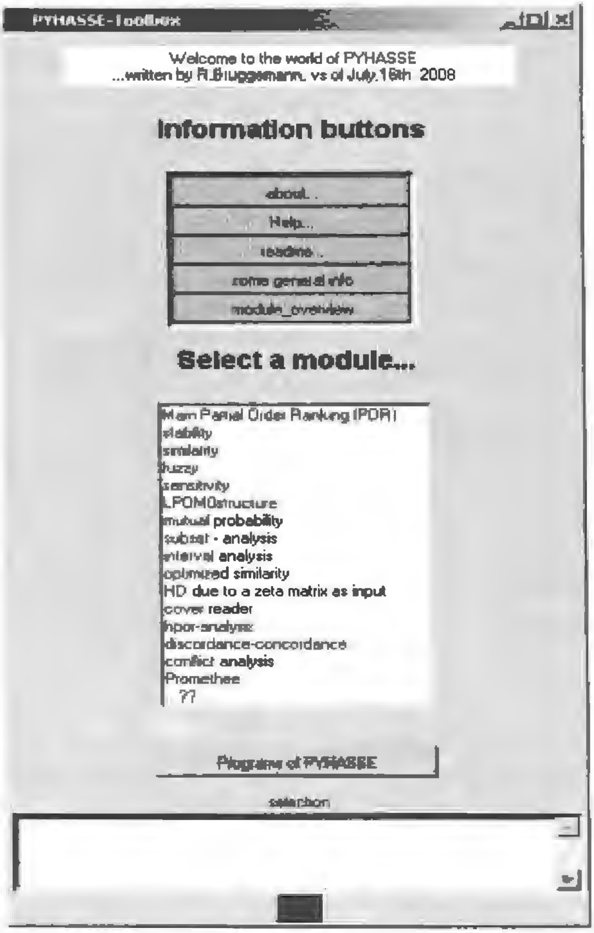
The PYTHON version 2.5 is available and was used for PyHasse. PYTHON is a free downloadable package and is an interpreting object-oriented programming language. There are good books about PYTHON available, see for instance Telles (2008). Programs written in PYTHON are open to everyone and can be easily adapted to specific purposes. PYTHON provides comfortable statements for pretty complex algorithms (e.g. set operations, the concept of 'dictionaries', realizing the general informatic concept of mappings) and is well legible, albeit its background is programmed in C. PYTHON-programs can be used on different platforms and under different operating systems; there are many specific and powerful libraries available and free downloadable, for example 'numpy' for numerical calculations, 'linear algebra' for matrix- and vector operations or 'PIL' for graphics. If there are time-sensitive numerical calculations, programs written in C can be easily embedded in the PYTHON software.

Graphical user interfaces (GUIs) can be programmed by Tkinter, which is available together with PYTHON and allows to have access to the tools of Tk/Tcl. PYTHON supports testing and can be thought of as an 'experimental software', allowing the programmer to quickly gain experience in programming work.

More information about PYTHON, including its history, can be found in Telles (2008). Before details are explained, a word in relation to WHasse and PyHasse may be useful: PyHasse is understood not as replacing WHasse, because WHasse was tested by far more intensively (just because of its earlier availability) and has excellent facilities to edit the Hasse diagrams. PyHasse contains some facilities, which are available in WHasse, too, only for user convenience. Examples are the sensitivity analysis and the p-algorithm (see below) tools.

2.2 General principles of PyHasse

First of all, PyHasse is under continuous development; and although carefully checked, it cannot be guaranteed that there are no bugs. PyHasse should be considered a test version, where a user has to be aware that there are still undetected problems. The first author welcomes any feedback from users. In its present form PyHasse contains 17 specific programs (called 'modules'). One of them is 'pyhassemenue2', the central platform from where general information can be obtained (for example a tutorial) and the module to apply selected, see Fig. 1.



Welcome to the world of PYHASSE
...written by R.Bruggemann, vs of July.16th 2008

Figure 1. User interface of pyhassemenue2

The appearance of the modules (i.e. their user interfaces) are as far as possible similar (see, e.g., Figs. 2 and 7). Each module has a 'help' function. This help function informs about:

- aim of the module,
- prerequisites (especially how to handle Excel®-data files as input),
- usage (or steps),
- known bugs or difficulties and
- recommended example files (provided with the PyHasse software).

In many cases there is also an 'about' button, which informs about the status of the module and gives background information, for example important literature.

Where appropriate, each 'select and open file' button is in white colour and each 'help' or 'about' information is green. Important buttons to proceed are in 'cyan' colour, the 'exit' button is red. Some other buttons, providing supporting information are grey. In some GUIs the buttons are numbered or are disabled/active to support their correct usage. Any module shows results in a series of extra windows, called 'daughter' windows, to allow the user to make appropriate screenshots. Any scrollable textbox can be copied and pasted into any Windows Office® program like Excel® or Word®.

All modules refer to a library, developed by the first author. This library is called 'rmod2' (after 'Rainer module version 2') and has to be installed at the location where the PYTHON software is situated (a detailed tutorial is available and provided with the software). The library 'rmod2' has - up to now - around 4,000 lines of code with a high variety of procedures. Some of them, however, are built for other purposes. Another library is under construction, which contains own classes in terms of object-oriented programming. In the future, mainly the programming of graphical user interfaces will be facilitated by this new library.

As mentioned, a user has access to all modules via pyhassemenue2. However, any module can be also started as a stand-alone program. In that case the technical name of the module must be known (the names, without the extension "py" but including the actual version number, can be found in pyhassemenue2 and in Table1, first column, italic letters).

PyHasse software can be obtained as a test version from the first author on request.

2.3 Brief discussion of the modules of PyHasse

In Table 1 we give the names as they appear in pyhassemenue2.

Table 1. Modules of PyHasse

Name	**Description**	**Graphics**	**Remarks**
Pyhassemenue *pyhassemenue2*	Central platform	Besides the graphical user interface there are no special graphics.	Access to a 'readme' file, delivered together with PyHasse, and a lot of helpful information.
Main Partial Order *mainhd7*	Equivalence classes, successors, predecessors and incomparabilities of each object. Population of levels, linear extensions. Averaged ranks due to the local partial order model. Averaged ranks applying the Bubley-Dyers algorithm (Bubley & Dyer, 1999; see also Denoeux et al., 2005).	Hasse diagram, dual Hasse diagram, principal order ideals and filters. Intervals. If the number of objects is too large then only the cover relations are shown.	The Bubley-Dyer algortihm needs a starting linear extension. The program provides two starters. See this contribution.
Fuzzy partial order *fuzzyHD10*	Following the fuzzy concept described in Van de Walle et al. (1995) and De Baets & De Meyer (2003a, b), fuzzy partial orders are derived after a series of steps.	Hasse diagram, depending on the α-cut selected. A histogram shows the distribution of α-cuts.	See the contribution of Annoni, Fattore, Brüggemann, in this volume.
Sensitivity *sensi5*	The Hasse diagram depends on the attribute set. Which attribute influences the Hasse diagram.	Hasse diagram, 'stability' as a function of a series of attribute subsets. Given an object x then the number of incomparable elements $y \parallel x$ is calculated as a function of attribute subsets.	See this contribution for more details.

Similarity *simi3*	Two posets, derived from two data matrices are compared.	Besides the corresponding Hasse diagrams, two bar diagrams are provided: (1) the similarity profile and (2) which objects contribute mainly to inversions.	See contribution of Voigt and Brüggemann, in this volume.
Optimised similarity *optimsim4*	Given a data matrix, consisting of one column, whose data are to be modelled, the subset of attributes of a second data matrix is found, optimising an objective function.	Hasse diagram of the optimal subset of attributes.	Also, a sensitivity analysis shows some of the best suboptimal solutions. See this contribution.
Subset analysis *dds8*	If the object set can be partitioned into subsets the dominance and separability degrees among these subsets is calculated.	Besides the Hasse diagram, a directed graph is derived, showing the dominance relations among subsets.	The acronym *dds* means "dominance-dominance-separability". See this contribution for an example and Restrepo & Brüggemann (2008a,b) for theoretical background.
p-algorithm *palg1*	The data matrix can be transformed according to 'cutting levels'. The resulting partial order is examined	Besides the Hasse diagram, several graphs, which may be helpful in selection of the appropriate cutting level	See Brüggemann et al. (1999b)
Stability analysis *stability3*	Attributes may be pairwise added to form 'super-attributes'. The superattribute depends on a weight *g*, called crucial weight. The linear order changes if *g* is varied.	Hasse diagram and the so-called *g*-spectrum (a bar diagram, showing the location of crucial weights).	The theoretical background is explained in Brüggemann et al. (2008) and Restrepo et al. (2008c).
LPOM structure *LPOMstruct1*	Based on structural properties of a Hasse diagram, averaged ranks due to Local Partial Order Model can be derived. In	Some approximations are shown by means of corresponding schematic diagrams.	The theoretical background is explained in Brüggemann et al. (2004, 2005).

	contrast to the Main Partial Order program, better approximations of local partial order are available.		
Mutual Probability *mutprobavrk*	Based on the structural properties of a Hasse diagram, mutual probabilities are calculated for a selected pair of incomparable objects.	No graphics are displayed.	For new theoretical developments, see De Loof et al. in this volume.
Interval order *interval3*	When objects are characterised by many attributes it can be meaningful to analyse the partial order just by the minimum and maximum value of the (normalised) attributes.	Both Hasse diagrams are shown: the usual Hasse diagram based on all attributes, and the one based only on the minimum and maximum values of the attributes of any object.	In most cases one is interested in the effect of any attribute. Hence, this program is only recommended in those cases where a rather general overview is wanted.
Hasse diagram from the ζ-matrix *zetareader1*	The order relation $\leq$ can be coded by a matrix, the ζ-matrix. Based on this matrix a Hasse diagram is obtained.	Hasse diagram.	This program is mainly thought of as interface to other theoretical approaches, e.g. to analyse hierarchical union, see Rademaker et al. in this volume.
Hasse diagram from the cover-matrix *covreader3*	The cover-relation can be used to obtain a Hasse diagram (Warshall-algorithm to obtain the transitive closure) and other basic information.	Hasse diagram.	This program is mainly thought of as interface to other theoretical approaches. For example to analyse the lattice theoretical approach to generate averaged ranks.
Hierarchical partial order *hpor1*	The averaged ranks due to the local partial order model of two data matrices are used as information basis for a new partial order.	The Hasse diagrams of the original data matrices as well as of the resulting new partial order are shown	This program is based on concepts, formulated by Helm (2006) and Carlsen (2008)

Discordance-Concordance *Disco2*	If a reference object is known then dis- and concordant objects can be found.	A Hasse diagram based on discordance and concordance is provided. The Hasse diagram, based on the original data matrix can be seen, too.	For the theoretical background: see Opperhuizen & Hutzinger (1982).
PROMETHEE *prom6*	A simple preference function is supposed. From that the dominance and subdominance vectors are derived.	A Hasse diagram based on dominance and subdominance vectors is shown. The Hasse diagram based on the original data matrix is accessible too.	For theoretical background, see for example: Brans & Vincke (1985).
Oreste6 *oreste6*	Let $x \parallel y$; then one may be interested in checking out whether $x > y$ or $y > x$ could be derived. Based on preferences for $x > y$ and for $y > x$ a graphic is obtained, which shows the positon of (x, y) as a function of additional stakeholder information.	A typical ORESTE-preference diagram is shown.	The crucial point is, how to get preferences for the pair (x, y). One idea may be to count (and weight) the attributes, where x is winning and those where y is winning. Background literature: Van Huylenbroeck (1995).

3. Modelling the toxicity of chemicals

3.1 Background

From Verhaar (1995) seven chemicals were arbitrarily selected, together with their toxicities (in mg/L), see Table 2. Multivariate statistics provide powerful tools to relate biological endpoints to molecular descriptors (see for example: Basak et al., 2000, 2003, and for an overview about application of QSAR-models in the context of REACH, Simon-Hettich, 2006 and 2008).

Table 2. Chemicals and their acute fish toxicity (mg/L)

Chemical	**Abbreviation**	**$\log_{10}$ (LC)**
1,1-Dichlorethane	Die	– 2.69
2-Butoxyethanol	Bue	– 2.08
1-Chlorbutan	Cb	– 2.98
1,3-Dihydroxy benzene	Dih	– 3.04
Ethanol	Et	– 0.31
2-Propanol	Pro	– 0.93
2,3,5,6-Tetrachloro aniline	Tca	– 5.93

However, if a prescreening and ranking is wanted, it may be useful to apply order theoretical techniques. In that case the molecular descriptors should be simple and easy to calculate, requirements met by topological indicators (abbr.: TIs) rather than geometric or quantum chemical indicators. From a set of 161 TIs, calculated by MOLGEN (Rücker et al., 2004), five TIs were selected after the calculation of their information content (Char-Tung, 1971) and the application of the K inflation factor technique developed by Todeschini et al. (1999). These are: J, the Balaban index; ch, the mean topological charge index of order 1; 6Ch, the chain connectivity index of order 6; S2, the structural information content of order 2 and cJ2, the mean topological charge index of order 2. A description of each one of these TIs can be found in Todeschini & Consonni (2000). As a priori it is unknown in which orientation the TIs should be used, the negative values of each TI (labelled negTI) should also be included in the indicator matrix, hence an analysis based on *m* TIs needs a descriptor matrix with 2 × *m* TIs. Therefore, the final descriptor-matrix contains 10 columns, i.e., 5 TIs + 5 negTIs.

3.2 Partial order

Let X be the set of chemicals, *Tox* the column of toxicities and D the attribute set of 10 columns containing molecular indicators. So we have, in general, two partial orders:

(X, *Tox*), which is a chain and

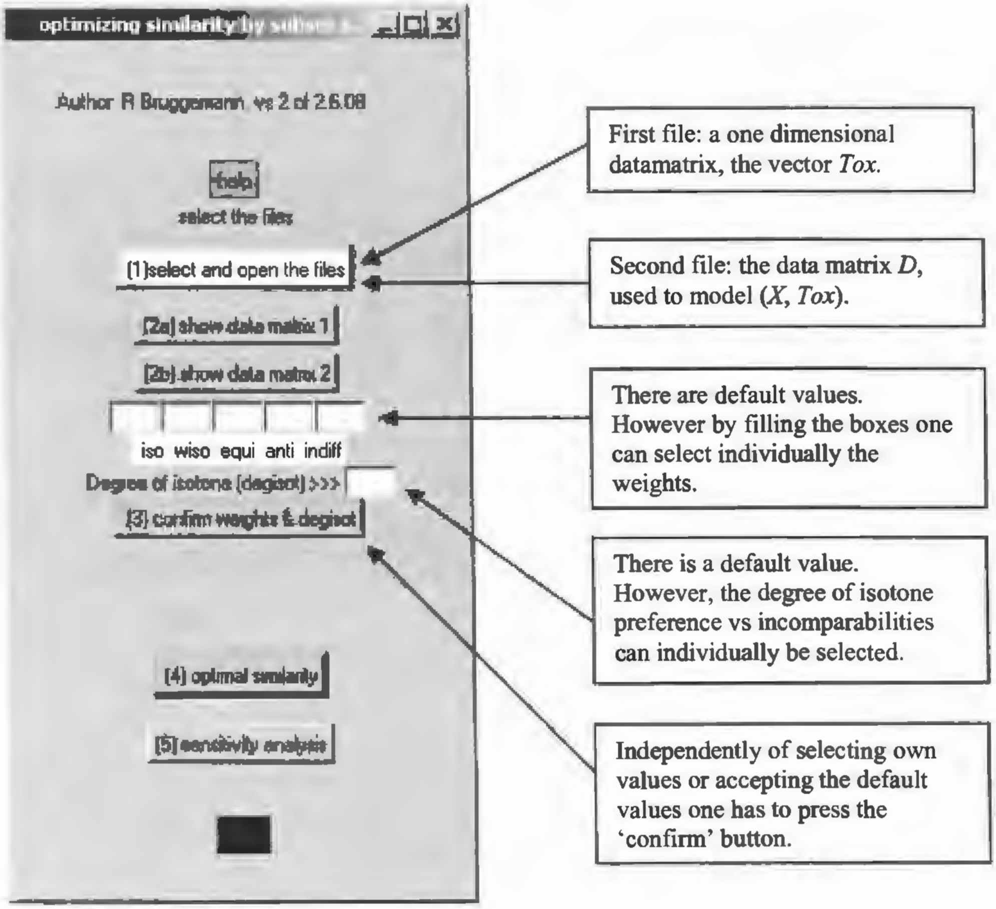
(X, D), which is partially ordered.

Now the task is: Which attribute subset out of D ($D_i \subseteq D$) can be found such that the map φ:

$$(X, D_i) \xrightarrow{\phi} (X, Tox)$$

is order preserving.

One difficulty arises: The set of five topological indicators does not differentiate between 2-Propanol and 1,1-Dichlorethan. Hence, 2-Propanol was eliminated. We avoid cumbersome notation and let *X* as symbol, even for referring to (*X*, *Tox*) and (*X*, D_i), now with only six chemicals.


optimizing similarity by subset ...
Author R Bruggemann vs 2 of 2.6.08
help
select the files
(1)select and open the files
(2a) show data matrix 1
(2b) show data matrix 2
iso wiso equi anti indiff
Degree of isotone (degisot) >>>
(3) confirm weights & degisot
(4) optimal similarity
(5) sensitivity analysis

First file: a one dimensional datamatrix, the vector *Tox*.

Second file: the data matrix *D*, used to model (*X*, *Tox*).

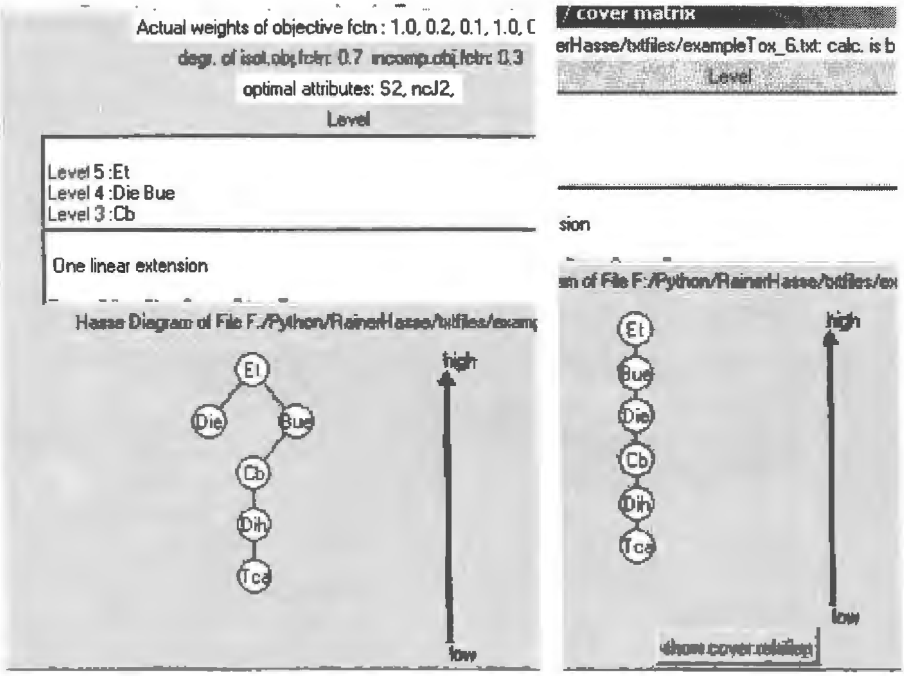
There are default values. However by filling the boxes one can select individually the weights.

There is a default value. However, the degree of isotone preference vs incomparabilities can individually be selected.

Independently of selecting own values or accepting the default values one has to press the 'confirm' button.

Figure 2. User interface of 'optimsim4'

To select $2^{10} - 1 = 1{,}023$ subsets of attributes (the empty set is not included) is certainly a job for the computer. Therefore a module for optimising the search (technical name 'optimsim4.py') was programmed, which (in principle) provides from a set of m attributes all $2^m - 1$ subsets. Instead of checking out whether the map $(X, D_i) \rightarrow (X, Tox)$ is order preserving an objective function was constructed, which is to be optimised.



Actual weights of objective fctn : 1.0, 0.2, 0.1, 1.0, C

degr. of isol.obj.fctn: 0.7 incomp.obj.fctn: 0.3

optimal attributes: S2, ncJ2,

Level 5 :Et
Level 4 :Die Bue
Level 3 :Cb

One linear extension

Hasse Diagram of File F:/Python/RainerHasse/txtfiles/examp

erHasse/txtfiles/exampleTox_6.txt: calc. is b

sion

am of File F:/Python/RainerHasse/txtfiles/ex

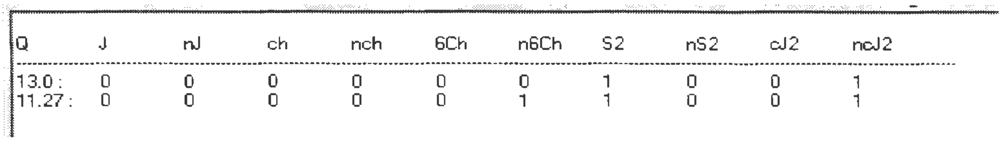
Figure 3. Screenshots of results of PyHasse. The Hasse diagram on the left corresponds to the partially ordered set (X, D_{i*}) with $D_{i*} = \{S2, ncJ2\}$ (see text). The one on the right to the partially ordered set (X, Tox)

The objective function, Q, is a weighted sum of two other functions Q_1 and Q_2. Q_1 selects the best single column out of D but accepts inversions, Q_2 is a function avoiding that any two objects be in an inverse relation $(a < b)$ due to Tox but $(b < a)$ due to a TI. Therefore, Q_2 is a function where the fraction of 'indifferent' (see Voigt et al., in this volume, for details) favours $(a \parallel b)$ in order to obtain an order preserving map.

In Fig. 2 the graphical user interface of the program 'optimsim4' is shown.

In Fig. 3 the two partial orders (referring to the six remaining chemicals) are presented and one can easily verify that there is indeed an order preserving map. It turns out that the attribute subset which optimises Q is {S2, ncJ2), the 'structural information content of order 2' and the negative values of 'mean topological charge index of order 2'. S2 is an indicator which quantifies the molecular complexity (Basak & Gute, 1997) and cJ2 quantifies the charge transfers between pair of atoms as well as the global charge transfer (Gálvez et al., 1995). Hence, in ncJ2 high scores correspond to molecules with few charge transfer and low scores to the opposite.

Is this optimum sharp? By means of a sensitivity analysis in 'optimsim4' we can obtain an answer, Fig. 4.



```
Q        J     nJ    ch    nch   6Ch   n6Ch  S2    nS2   cJ2   ncJ2
---------------------------------------------------------------------
13.0 :   0     0     0     0     0     0     1     0     0     1
11.27 :  0     0     0     0     0     1     1     0     0     1
```

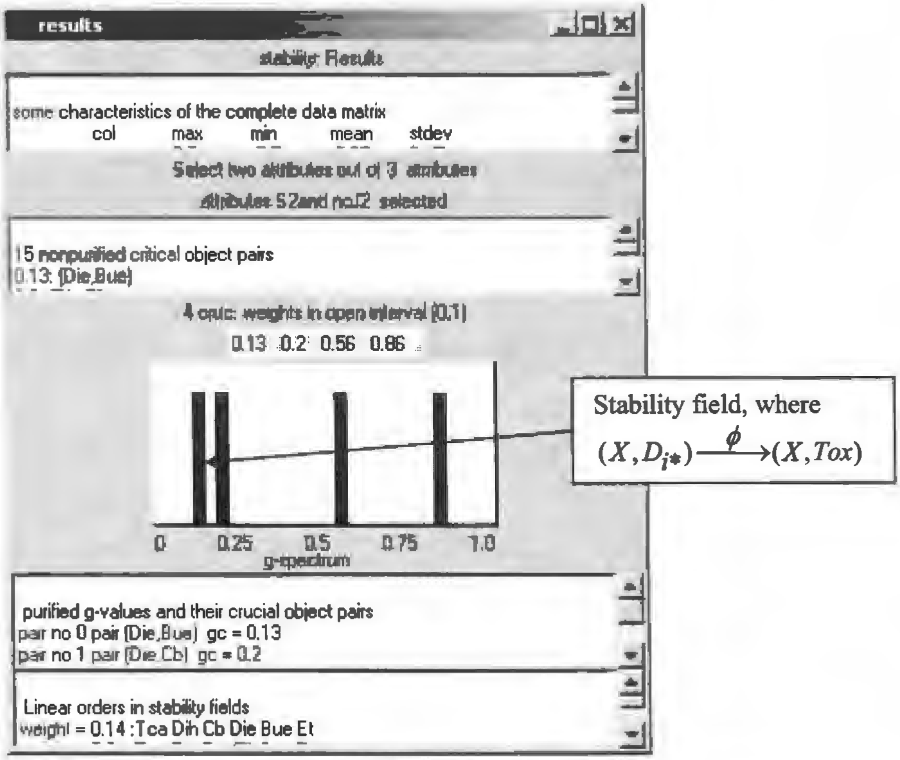
Figure 4. The optimal D_{i^*} = {S2, ncJ2} and one suboptimal attribute subset $D_{i^{**}}$ = {n6Ch, S2, ncJ2}, falling in the range corresponding to the 80% of the maximum value of Q

There is only one alternative, which adds n6Ch, the negative values of the chain connectivity index of order 6, which is related to the complexity of the molecular graph. With this extended, suboptimal set, we perform one sidestep: see Section 3.3.

Having a look at the Hasse diagram (X, D_{i^*}) one may argue that it must be possible to linearly aggregate the two optimal TIs of D_{i^*} in such a way that in one stability field (see Brüggemann et al., 2008; Restrepo et al., 2008c) the order is found which is isomorphic to that of (X, Tox). The linear aggregation used is $\phi = g \cdot \mathrm{S2} + (1-g) \cdot \mathrm{ncJ2}$, which comes to be the worked out mapping of (X, D_{i^*}) onto (X, Tox). In Fig. 5 the g-spectrum is shown.

In the text box (visible at the bottom of the screenshot, Fig. 5) all the linear orders of any stability field are listed on:

weight = 0.01 :Tca Dih Cb Bue Die Et
weight = 0.14 :Tca Dih Cb Die Bue Et
weight = 0.21 :Tca Dih Die Cb Bue Et
weight = 0.57 :Tca Die Dih Cb Bue Et
weight = 0.87 :Die Tca Dih Cb Bue Et



stability: Results

some characteristics of the complete data matrix
col max min mean stdev

Select two attributes out of 3 attributes
attributes S2and ncJ2 selected

15 nonpurified critical object pairs
0.13: (Die,Bue)

4 cruc: weights in open interval (0,1)
0.13 0.2 0.56 0.86

purified g-values and their crucial object pairs
pair no 0 pair (Die,Bue) gc = 0.13
pair no 1 pair (Die,Cb) gc = 0.2

Linear orders in stability fields
weight = 0.14 :Tca Dih Cb Die Bue Et

Stability field, where
$(X, D_{i*}) \xrightarrow{\phi} (X, Tox)$

Figure 5. *g*-spectrum of the linear combination ϕ of S2 and ncJ2

In terminology of Brüggemann et al. (2008) this corresponds, respectively, to:

$0 < g < 0.13 \rightarrow$ Tca < Dih < Cb < Bue < Die < Et

$0.13 < g < 0.2 \rightarrow$ Tca < Dih < Cb < Die < Bue < Et

$0.2 < g < 0.56 \rightarrow$ Tca < Dih < Die < Cb < Bue < Et

$0.56 < g < 0.86 \rightarrow$ Tca < Die < Dih < Cb < Bue < Et

$0.86 < g < 1 \rightarrow$ Die < Tca < Dih < Cb < Bue < Et.

A comparison of all the orders shows that just the second stability field ($0.13 < g < 0.2$) yields the correct order among the six chemicals. As the weight g refers to the contribution of S2 to the aggregated function ϕ, (whereas $(1 - g)$ to the contribution of nCJ2 to ϕ) this result tells us that only a slight degree of S2 is sufficient to attain coincidence between the order induced by *Tox* and that by ϕ. In other words, a small priority of the molecular complexity over the deficiency in the charge transfer is enough to obtain the total order generated by the toxicity.

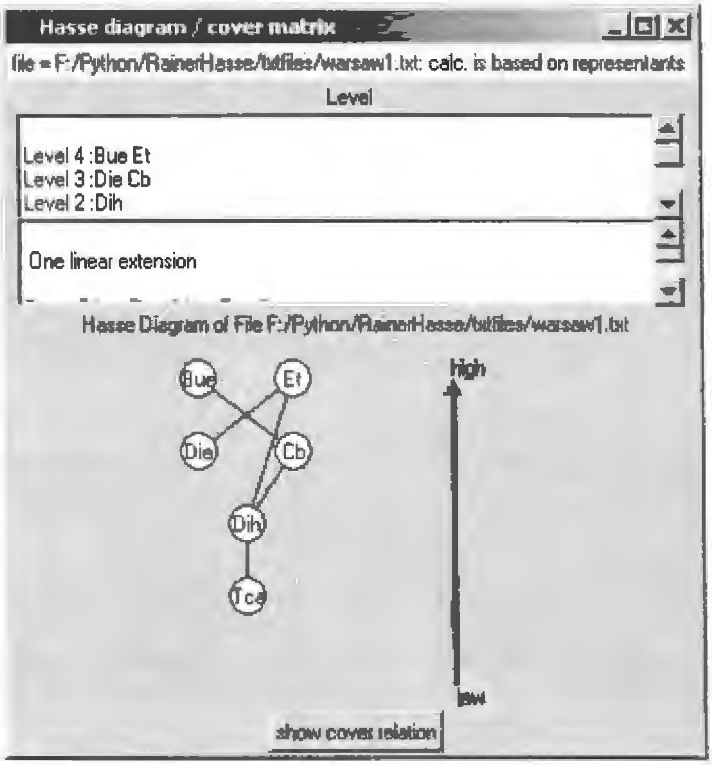
3.3 Chemical structures

We do not claim that now the goal of a good ordinal modelling is reached, as for example one chemical, 2-propanol, had to be eliminated, because no one of the five TIs could differentiate 1,1-Dichlorethane from 2-propanol, albeit the toxicities are strongly differing. However, we think that with the tool 'optimsim4' we may approach that goal in the future.

Nevertheless, we could ask ourselves, how far simple chemical structural information follows the toxicity. Therefore, we selected three classes of chemicals: The Cl-containing nonaromatic compounds, the O-containing ones and the aromatic substances. Just to see the effect of the suboptimal attribute set, we are now referring to the attributes {S2, ncJ2, n6Ch}. The Hasse diagram derived from this attribute set is depicted in Fig. 6. This diagram was obtained by applying the main partial order program (technical name: mainHD7). Owing to the large amount of results delivered by this module, functionality was implemented giving an overview about the results. In Figs. 7 and 8 the graphical user interface and the overview about the main results are shown, respectively.

Note that in PyHasse no attempt was done to program a graphical editor by which the Hasse diagram can be edited. Hence, if it is required for instance to minimise the number of crossings, it is better to use the software WHasse, instead.

By eye inspection one may easily derive that there is indeed a structural relation, as the low toxicity refer to Et and Bue, the medium one to Die and Cb and the high toxicity to the two aromatic substances Dih and Tca. Note that in Figs. 2 and 5 the chemicals of low toxicity are at the top of the Hasse diagram, whereas the highly toxic chemicals are located at its bottom. However, in general, in complex partial orders, optical inspection will be cumbersome and may lead to erroneous results. Therefore, the module subset analysis (technical name 'dds7.py') was developed: Any set of objects may be partitioned and the disjoint subsets may be examined for their dominance degrees (Restrepo et al., 2008a;b).


Hasse diagram / cover matrix
Level
Hasse Diagram of File F:/Python/RainerHasse/txtfiles/warsaw1.txt
Bue
Et
Die
Cb
Dih
Tce
high
low
show cover relation

file = F:/Python/RainerHasse/txtfiles/warsaw1.txt: calc. is based on representants

Level 4 :Bue Et
Level 3 :Die Cb
Level 2 :Dih

One linear extension

Figure 6. One of the windows obtained from the main partial order program (technical name: 'mainHD7')

A partition of the set under study may be: chlorinated nonaromatics, labelled 0, substances containing oxygen, labelled 1, and aromatics, labelled 3. For these subsets a partial order could be established (Fig. 9). Generally, the subsets can be considered as vertices of a directed graph (digraph). Any directed edge of that graph is evaluated by the corresponding dominance degree. The crucial point is how far the transitivity can be established. In that context the contribution of De Loof in this issue is of high interest, as establishing an order among subsets falls in the field of 'cyclic transitivity'. See also De Baets & De Meyer (2008).

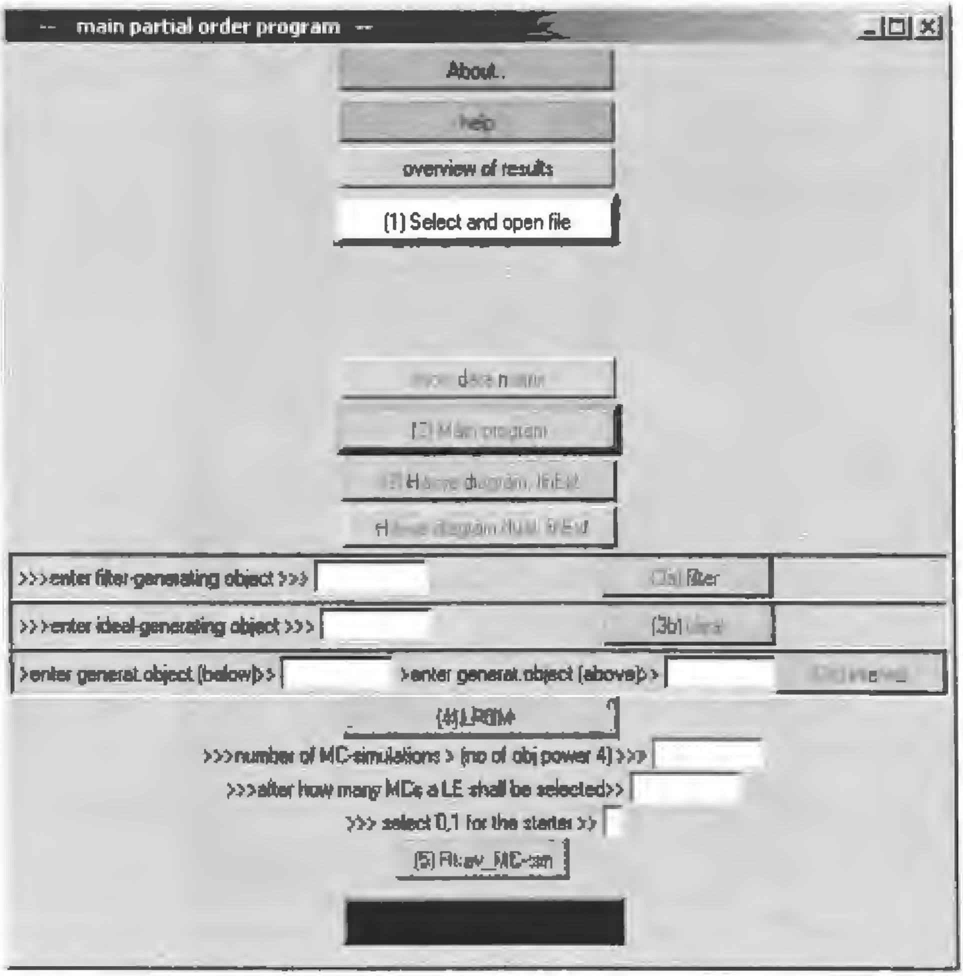

-- main partial order program --
About.
help
overview of results
(1) Select and open file
(2) Main program
>>>enter filter-generating object >>>
(3a) filter
>>>enter ideal-generating object >>>
(3b) ideal
>enter general.object (below)>>
>enter general.object (above)>>
(4) LPOM
>>>number of MC-simulations > (no of obj power 4) >>>
>>>after how many MCs a LE shall be selected>>
>>> select 0,1 for the starter >>
(5) Rkav_MC-sim

Figure 7. Graphical user interface of ‘mainHD7.py’. Note that some buttons are disabled because no file was selected an opened

As low dominance degree values may be considered as ‘noise’, only those directed edges are retained whose dominance degrees are above a certain limiting value (called ‘filter’), which is an input of the user. In Fig. 8 that filter was set up at 0.3, meaning that more than 30 % of all possible order relationships between the elements of a subset A and another B correspond to cases where the elements of A have greater scores than the elements of B.

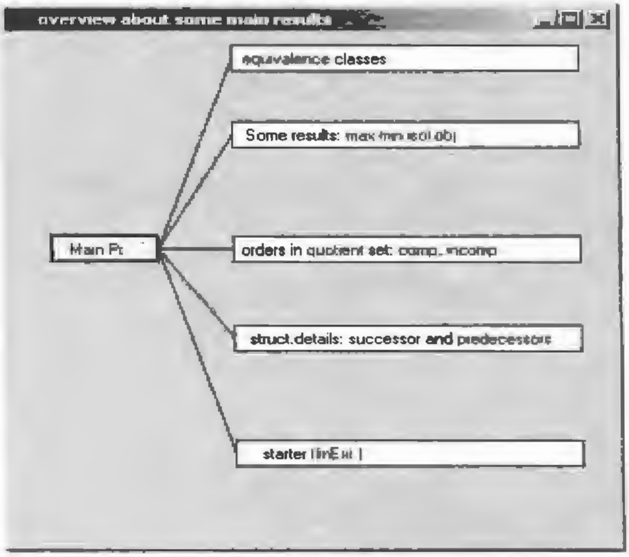
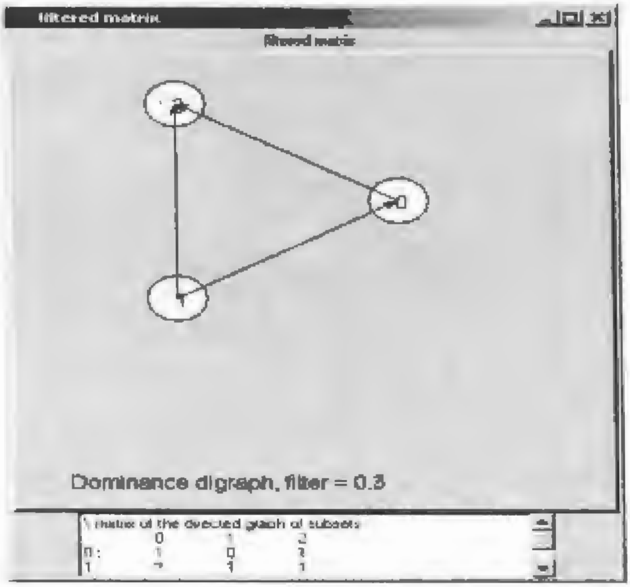

overview about some main results
Main Pr
equvalence classes
Some results: max/min isol.obj
orders in quotient set: comp., incomp
struct.details: successor and predecessors
starter (linExt)

Figure 8. Overview about the main results of ‘mainHD7.py’



```
\ matrix of the directed graph of subsets
        0       1       2
0:      1       0       1
1       1       1       1
```

Figure 9. Digraph of the three sets: 0: Chlorinated nonaromatics, 1: substances containing oxygen, 3: aromatics

4. Pollution in the south-western German state of Baden-Württemberg

One of the main tools of analysing Hasse diagrams is sensitivity. Although WHasse has a very good and convenient facility to perform sensitivity studies, PyHasse has a module (technical name 'sensi4.py') for doing it as well. In the module sensi4.py a sensitivity study is performed based on the distance among the ζ-matrices induced by different attribute subsets. Beyond this, P(IB) (that, is the count of all incomparabilities divided by $n \times (n - 1)/2$, n being the number of objects) as a function of attribute sets (called information basis IB) is shown as a graph. As the attribute subsets are not completely ordered, a canonical sequence is obtained by starting with the most important attribute and merging stepwise the less important ones, thus increasing the content of IB. Therefore, P(IB) can be obtained as an unique function of IB. Finally, one can associate with each object its number of incomparable elements: $U(x) = |\{(x, y): x \parallel y, (x, y) \in X\}|$. In consequence, one can obtain for any object x graphs $U(x)$ as a function of the canonical order, i.e. with increasing number of attributes. The graphical user interface of the sensitivity program is built following the same principles as for all the other modules and it will not be therefore shown here.

The data matrix describing the pollution of Baden-Württemberg consists of 47 regions (isolated regions were in a preprocessing step eliminated) characterised by four attributes describing the concentrations of Pb, Cd, Zn and S in the herb layer as well as Pb, Cd and Zn in the leaves of trees. P(IB) is depicted in Fig. 10 (top) as a function of the canonical order of attribute sets. The majority of the 47 objects will have the same kind of functional dependence as P(IB). However, some regions do not follow that pattern, and these may be of interest: The number of $U(x)$ (x is here region 14) does not increase after aggregating the - in general - next most-important attribute, but does a steep step when the fourth attribute is merged. The conclusion is that the first three attributes do not change much the relation of region 14 to that of the other ones.

5. Problems and future work

PyHasse with actually 18 (including pyhassemenue2) modules was developed using PYTHON and Tkinter. Up to now, even for the calculation of a uniformly selected arbitrary set of linear extensions (following the Bubley-Dyer algorithm) no care was taken for numerical efficiency. Tests with around 30 objects were very satisfying but problems with computational time may arise if the number of objects dramatically increases.

Another problem is that the graphics can, up to now, only be transferred to Office-documents by screenshots. With the library PIL of PYTHON one could easily implement a drawing program, which is useful for text documents. However, an additional download of the users would be necessary. At least, currently, we avoid this inconvenience. A similar problem arises when the library of classes will be available, albeit there may be tools, available in PYTHON to simplify the implementation of complex program systems.

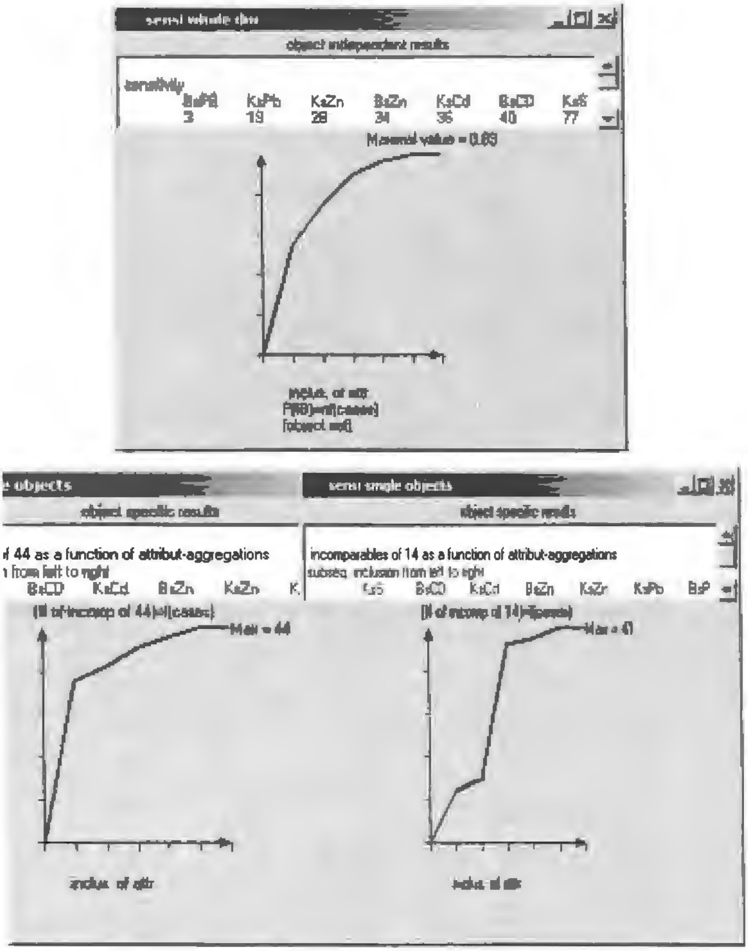

object independent results
sensitivity
BaPb 3
KaPb 19
KaZn 28
BaZn 34
KaCd 36
BaCD 40
KaS 77
Maximal value = 0.89
e objects
object specific results
f 44 as a function of attribut-aggregations
n from left to right
BaCD KaCd BaZn KaZn K
(# of incomp of 44)=f(cases)
Max = 44
index of attr
object specific results
incomparables of 14 as a function of attribut-aggregations
subset, inclusion from left to right
KaS BaCD KaCd BaZn KaZn KaPb BaP
(# of incomp of 14)=f(cases)
Max = 41
index of attr

Figure 10. P(IB) (top) and U(*x*) (bottom) as function of the canonical order of attribute subsets

Many other technical inconveniences must be gradually eliminated; for instance the handling of the graphic windows is still rather inflexible. However, these technical issues will not have the highest priority. Indeed, the main focus in future development will be to include new tools, for example those developed by De Loof et al., 2006 (and this volume) and Rademaker (see this volume). Some other new ideas coming from the 'partial order community' could be implemented in the software. The best what can be happened is that many interested scientists are using PyHasse, so that a stream of ideas concerning the software helps to make PyHasse more and more comfortable, so that new theoretical ideas may arise just from playing with the software.

References

Basak S.C. and Gute B.D. (1997) Characterization of molecular structures using topological indices. *SAR QSAR Environ. Res.*, 7, 1-21.

Basak S.C., Grunwald G.D., Gute, B.D., Balasubramanian K., Opitz D. (2000) Use of statistical and neural net approaches to predicting toxicity of chemicals. *J. Chem. Inf. Comp. Sci.*, 40, 885-890.

Basak S.C., Balasubramanian K., Gute B.D., Mills D., Gorczynska A., Roszak S. (2003) Prediction of cellular toxicity of halocarbons from computed chemodescriptors: A hierarchical QSAR approach. *J. Chem. Inf. Comp. Sci.*, 43, 1103-1109.

Brans J.P. and Vincke P.H. (1985) A preference ranking organisation method (The PROMETHEE method for multiple criteria decision - making). *Manag. Sci.*, 31, 647-656.

Brüggemann R., Bücherl C., Pudenz S., Steinberg C. (1999a) Application of the concept of partial order on comparative evaluation of environmental chemicals. *Acta Hydrochim. Hydrobiol.*, 27, 170-178.

Brüggemann R., Pudenz S., Voigt K., Kaune A., Kreimes K. (1999b). An algebraic/graphical tool to compare ecosystems with respect to their pollution. IV: Comparative regional analysis by Boolean arithmetics. *Chemosphere*, 38, 2263-2279.

Brüggemann R., Sørensen P.B., Lerche D., Carlsen L. (2004) Estimation of averaged ranks by a local partial order model. *J. Chem. Inf. Comp. Sci.*, 44, 618-625.

Brüggemann R., Simon U., Mey S. (2005) Estimation of averaged ranks by extended local partial order models. *MATCH - Commun. Math. Comput. Chem.*, 54, 489-518.

Brüggemann R., Voigt K., Restrepo G., Simon U. (2008) The concept of stability fields and hot spots in ranking of environmental chemicals. *Environ. Modell. Softw.*, 23, 1000-1012.

Bubley R., Dyer M. (1999) Faster random generation of linear extensions. *Discr. Math.* 201: 81-88.

Carlsen L. (2008) Hierarchical partial order ranking. *Environ. Pollution* (in press) doi:10.1016/j.envpol.2007.11.02 3

Char-Tung R. (1971) Application of information theory to select relevant variables. *Math. Biosci.,* 11, 153-161.

Davey B.A. and Priestley H.A. (1990). *Introduction to Lattices and Order.* Cambridge University Press, Cambridge, UK.

De Baets B. and De Meyer H. (2003a) On the existence and construction of *T*-transitive closures. *Inform. Sciences,* 152, 167-179.

De Baets B. and De Meyer H. (2003b) Transitive approximation of fuzzy relations by alternating closures and openings. *Soft Comput.,* 7, 210-219.

De Baets B. and De Meyer H. (2008) On the cycle-transitivity comparison of artificially coupled random variables. *J. Approximate Reas.*, 47, 306-322.

De Loof K., De Meyer H., De Baets B. (2006) Exploiting the lattice of ideals representation of a poset. *Fund. Informaticae,* 71, 309-321.

Denoeux T., Masson M.-H., Hébert P.-A. (2005) Nonparametric rank-based statistics and significance tests for fuzzy data. *Fuzzy Set Syst.*, 153, 1-28.

Gálvez J., García-Domenech R., De Julián-Ortiz J.V., Soler R. (1995) Topological approach to drug design. *J. Chem. Inf. Comp. Sci.*, 35, 272-284.

Halfon E. (2006) Hasse diagrams and software development. In: Brüggemann R. and L. Carlsen (Eds.), *Partial Order in Environmental Sciences and Chemistry*. Springer-Verlag, Berlin, 385-392.

Helm D. (2006) Evaluation of Biomonitoring Data. In: Brüggemann R. and L. Carlsen (Eds.), *Partial Order in Environmental Sciences and Chemistry.* Springer-Verlag, Berlin, 285-307.

Opperhuizen A. and Hutzinger O. (1982) Multi-criteria analysis and risk assessment. *Chemosphere*, 11, 675-678.

Pudenz S. (2005) ProRank - software for partial order ranking. *MATCH - Commun. Math. Comput. Chem.,* 54, 611-622.

Restrepo G., Weckert M., Brüggemann R., Gerstmann S., Frank H. (2008a) Ranking of refrigerants. *Environ. Sci. Technol.*, 42, 2925-2930.

Restrepo G. and Brüggemann R. (2008b). Dominance and Separability in posets, their application to isoelectronic species with equal total charge. *J. Math. Chem.*, 44, 577-602, DOI: 10.1007/s10910-007-9331-x.

Restrepo G., Brüggemann R., Weckert M., Gerstmann S., Frank H. (2008c) Ranking patterns, an application to refrigerants. *MATCH - Commun. Math. Comput. Chem.*, 59, 555-584.

Rücker C., Meringer M., Kerber A. (2004) QSPR using MOLGEN-QSPR: The example of haloalkane boiling points. *J. Chem. Inf. Comp. Sci.*, 44, 2070-2076.

Simon-Hettich B., Rothfuss A., Steger-Hartmann T. (2006) Use of computer-assisted prediction of toxic effects of chemical substances. *Toxicology,* 224, 156-162.

Simon-Hettich B. (2008) Toxizitätsmodellierung mit kommerziellen (Q)SAR-Systemen: praktische Aspekte für die regulatorische Anwendung. *Mitteilungen der Fachgruppe Umweltchemie und Ökotoxikologie*, 14, 45-49.

Stanley R.P. (1986) *Enumerative Combinatorics*. Vol. I. Wadsworth & Brooks/cole, Monterey, USA.

Telles M. (2008) *Python Power! The Comprehensive Guide*. Thomson Course Technology PTR, Boston, USA.

Todeschini R., Consonni V., Maiocchi A. (1999) The K correlation index: Theory development and its application in chemometrics. *Chemometr. Intell. Lab.*, 46, 13-29.

Todeschini R. and Consonni V. (2000) *Handbook of Molecular Descriptors*. Wiley-VCH; Weinheim, Germany.

Trotter W.T. (1992) *Combinatorics and Partially Ordered Sets Dimension Theory.* The Johns Hopkins University Press, Baltimore, Maryland.

Van de Walle B., De Baets B., Kerre E.E. (1995) Fuzzy multi-criteria analysis of cutting techniques in a nuclear reactor dismantling project. *Fuzzy Set Syst.*, 74, 115-126.

Van Huylenbroeck G. (1995) The conflict analysis method: Bridging the gap between ELECTRE, PROMETHEE and ORESTE. *EJOR*, 82, 490-502.

Verhaar H.J.M. (1995) *Predictive Methods in Aquatic Toxicology.* Dissertation. Utrecht, Netherlands.

This book is a collection of papers, prepared in connection with the 8th International Workshop on partial orders, their theoretical and applied developments, which took place in Warsaw, at the Systems Research Institute, in October 2008. The papers deal with software developments (PYHASSE and other existing software), theoretical problems of ranking and ordering under various assumed analytic and decision-making-oriented conditions, as well as experimental studies and down-to-earth pragmatic questions.

ISBN 83-894-7521-9
EAN 9788389475213