MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **65** (2011) 383-414

ISSN 0340 - 6253

An Improved Estimation of Averaged Ranks of Partial Orders

Rainer Brüggemann

Leibniz Institute of Freshwater Ecology and Inland Fisheries, Department Ecohydrology,
D-12587 Berlin-Friedrichshagen, Mueggelseedamm 310, Germany

Lars Carlsen[b,c]

[b] *Awareness Center, Linkøpingvej 35, Trekroner, DK-4000 Roskilde, Denmark*

[c] *Department of Chemical Engineering, Kazakh-British Technical University,*
Tole bi str. 59, Almaty 050000, Kazakhstan

(Received April 9, 2010)

Abstract

Comparison of objects characterized by a multitude of criteria will typically not lead to a linear order, but to a partial order. However, often a linear order is desirable or even required. The present paper presents an improved – extended – approximate local partial order model to estimate a weak or linear order based on averaged ranks of the studied objects originally being partially ordered.

The paper analyses various possible partial order scenarios by means of the new local partial order model, the results being compared to the original approach as well as to exact values (their calculation can be extremely time consuming), demonstrating a distinct improvement of the extended method compared to the original local partial order ranking method. By the approximate methods the values of averaged ranks can be understood in terms of three basic partial order parameters.

The method is applied to current research on human health effects of rocket fuel transformation products.

1. Introduction

Partial order is a very general and simple mathematical structure. Partial order appears almost everywhere in mathematics [1], albeit it is often not explicitly mentioned and sometimes not directly helpful. The generality of partial order allows that this mathematical structure appears in different facets. One of these facets that becomes useful in chemistry and environmental problems is the use of the product order, i.e., component wise order [2-4]. The product order is the order theoretical way to handle multivariate problems, i.e., to study ranking of objects, which are simultaneously characterized by a set of attributes, which may be of different scaling level or are quantifying different non-commensurable aspects [5].

Partial order is simply a consequence of the fact that for two objects x and y the relation $q_i(x) < q_i(y)$ does not hold for all attributes q_i, which are considered as relevant. Hence, the resulting partial order of a set of objects is often not a linear, i.e., complete order. Nevertheless, its analysis allows valuable conclusions about the objects. Some recent papers may be mentioned as examples of the application of partial order in the multivariate context [6-10]. However often, especially in the field of decision making a complete order is preferred or even required. There are many well known methods to obtain from a multivariate data matrix a linear, or with respect to technical aspects also a weak, i.e. including ties, order like PROMETHEE, ELECTRE or REGIME [11-13]. However, all these methods require parameters beyond the data matrix to run them and may even be criticized from a theoretical point of view. For a comprehensive view, see [14]. In that sense, partial order is considered as the method that is based as far as possible on the data matrix alone.

Since the pioneering work of Halfon and Reggiani, 1986 [15] many mathematical tools were developed, which are helpful in analyzing partially ordered sets (posets) [5,16]. Of special interest are methods, which aim at deriving a linear or weak order from a poset, without requiring additional preference parameters. We call this order the “canonical ranking” as it has its origin from the data matrix alone. We do not claim that this canonical order is better than the linear orders obtained from Multi Criteria Decision Aid (MCDA)-methods, but we recommend the use of canonical rankings for comparison purposes or in cases where subjective preferences are too ambiguous.

In the present paper we describe a new method for an approximation of a linear (or weak) order, because the computation of the exact ranks is often computationally intractable or even impossible to achieve if the number of objects to be evaluated exceeds 50 (see also Fig.11).

The paper is organized as follows:

(1) The computational results are based on the software PyHasse. Therefore we give a brief introduction into PyHasse.

(2) Some technical information about partial order to the extent necessary and a description of how linear (or weak) order can be derived is presented and advantages and disadvantages discussed.

(3) The new approximation, extended local partial order model (LPOMext) will be explained and by simple model partial orders approximation errors discussed. A statistical performance analysis is subsequently described.

(4) The new method will be illustrated through application on a specific example from our current research activities on the human health impact of rocket fuel residues.

(5) The paper is finalized by conclusions and outlook.

2. Some more information about PyHasse

The tasks in this paper can be described as follows:

1) Calculate the exact averaged ranks
2) Calculate the averaged ranks after the new LPOMext model
3) Calculate averaged ranks after the original LPOM0 model (see below)
4) Generate randomly data matrices with given range of number of rows and number of columns
5) For most of the model posets shown in this paper and for many others (publication in preparation) the first author could develop closed formulas. So it was possible without spending too much time to calculate the averaged rank as a function of structural parameters, like m and t in the poset of Fig. 3. Nevertheless a calculation by hand is tedious and therefore a module of PyHasse was developed which calculates the averaged rank as function of structural partial order parameters.

The software PyHasse written in the interpreter language Python version 2.6 can be considered as the successor of the software WHASSE the development of which stopped in 2001. PyHasse, as a pretty new software is described in [17]. PyHasse was and will continuously be developed by the first author. PyHasse includes actually (May, 2010) 43 modules. “Hasse” stands for “Hasse diagram technique” and is also thought of as a reference to the German mathematician H. Hasse. Due to the name of the programming language the software got the prefix “Py”. Beside libraries which are freely downloadable, like statistical packages, the first author developed two libraries, rmod2 and raioop2. The library rmod2 contains basic procedures and some formulas of combinatorial type. The library raioop2 contains object – oriented classes, mainly for programming graphics and user interfaces. PyHasse is considered as a software which bridges the gap between professional software and software, typically developed in labs and only understandable by a narrow range of scientists.

All modules have a graphical user interface, a help- and an about-function. The help function informs about the aim of the module, the steps to be done by the user, possible difficulties, recommended test data matrices and –if available- about literature. Most important is mainHD19.py which calculates many features, important within the frame of partial order. In this paper the modules are of special interest which calculate exactly or approximately the averaged ranks: The exact averaged ranks are calculated by means of module avrank4.py. LPOM0 is a part of mainHD19.py. LPOMext-values can be obtained from module hdgt5.py, which also calculates some simple graph-theoretical quantities, which, however, do not play any role in our paper.

As testing of the modules and of new concepts need data matrices, a module randomdm2.py, was written which generates randomly data matrices in certain dimensions.
All the modules can be selected by applying the central platform pyhassemenue7.py, which also provides detailed information about PyHasse.

3. From partial to linear order

3.1 Settings of partial order

Let X be a set of object $\{x_1,\ldots,x_n\}$[1] and IB (information base) a set of attributes $q_1,\ldots,q_m$. We assume finite sets and we symbolize the count of elements of a finite set A by $||A||$. We define the product order as follows:

$(x,y) \in X^2$, $x < y$ iff $q_i(x) \leq q_i(y)$ for all $q_i \in$ IB and there is at least one index i*, with $q_{i^*}(x) < q_{i^*}(y)$. (1)

Obviously not all objects $(x,y) \in X^2$, can be ordered by (1). When an object x is considered, there may be a set of objects, which are incomparable with x. This set is of special interest and is called U(x) [18,19]:

$U(x): = \{y \in X: y||x\}$ (2)

where the sign || denotes y and x being incomparable.

Further there may be objects which cannot be compared with any other object by means of eq. (1). These special objects are called 'isolated'.

In addition order ideals, or down sets O(x) [2],

$O(x):=\{y \in X: y \leq x\}$ (3)

are of importance. Often we want to exclude x, then we are speaking of a successor set $S(x) : = O(x) - \{x\}$.

Similarly important is the concept of order filters, F(x) [2]:

$F(x):=\{y \in X: x \leq y\}$ (4)

[1] In context with the directed graph, obtained from the partial order we also speak of vertices

If we want to exclude x, we are speaking of the predecessors of x.

It is meaningful to introduce the set of all objects, which are comparable with x, C(x), which obviously can be defined as:

$$C(x) = F(x) \cup O(x) \qquad (5)$$

A chain of a poset is a subset of X, such that all objects of this subset are mutually comparable by eq. (1), whereas an antichain of a poset is a subset X' $\subseteq$ X, such that no object of X' can be compared with any other by means of eq. 1.

If x > y and there is no element z such that x>z>y then x “covers” y and the corresponding edge in the graph-theoretical visualization is called a “covering edge”.

An important characteristic of a partial order is the total number of incomparable pairs of objects, U. As U is a measure of relations, it should be normalized by the number of all possible pairs of objects, i.e., by n·(n-1)/2, n being ||X||, the content of object set X. Thus, we introduce the normalized measure of relations P(IB) [20].

$$P(IB) = \frac{U}{\binom{n}{2}} \qquad (6)$$

A further parameter of partial order that plays a role in this study is the ‘length’ of a partial order, which is simply the number of objects in the longest possible chain -1 [2].

A linear extension [21] is a linear order, which respects all order relations of the poset. In general from a poset several linear extensions can be obtained, as an incomparability obviously allows x < y or y < x within a linear extension without hurting the given orders of a poset. For each object x a height H(x,le(k)) can be found giving information about the number of objects ≤ x in le(k), the kth linear extension. The number of all linear extensions of a poset is called LT. LT ≤ n!, where n is the number of objects in the poset. The set of linear extensions of a certain poset can be treated by standard statistics. For example one can derive the mean of H(x,le(k)) over all LT linear extensions:

$$Rkav(x) = \frac{\sum_{k=1}^{n} H(x, le(k))}{LT} \qquad (7)$$

If the mean of H(x,le(k)) is known (called Rkav(x)), the set X can be ordered just by this quantity. Hence, there is a method at hand, which does not require (subjective) preference parameters to get a linear or weak order. We call the order of the objects of X, induced by Rkav the canonical order of a poset. The canonical order is in general a weak order, due to order theoretical symmetries within the poset (in technical terms: graph theoretical automorphisms, see e.g. Harary, [22]). The direct computational realization of averaged ranks of a medium or higher sized object set via calculating the set of all linear extensions, and derive from that statistical explorative quantities is extremely difficult and for any practical purposes not applicable. For example in WHASSE (an often applied software used to analyze product orders [23]) the calculation of the exact canonical order is only possible if $n \leq 15$, $n = ||X||$. Hence, there is a need of operational alternatives for practical applications, e.g. in relation to decision making.

3.2 Alternatives and approximations for the calculation of averaged ranks

3.2.1 Lattice theoretical method

A well known fact, namely that the partially ordered set of order ideals derived from a partially ordered object set is a lattice (especially a distributive lattice [24]) is applied to generate the linear extensions "on the fly" and from them several quantities, like the averaged rank as described in [25-27]. Partially this is used in the software package PyHasse, where a concept of a dynamically stored lattice is applied (module avrank4.py) [28]. Although avrank4.py works well, it has two major disadvantages:

1) When the number of objects or U is large, this procedure is still very time consuming and - in our eyes - more important
2) the mechanism how to get a linear order from a given poset is hardly understandable in terms of O(x), F(x) and U(x) or other simple descriptive details of a partial order.

Therefore there is still a strong need for appropriate approximations. Bruggemann et al. [18] introduced the 'Local Partial Order Model' (LPOM). The main concepts of the LPOM , version 0 (LPOM0), are as follows.

3.2.2 LPOM0

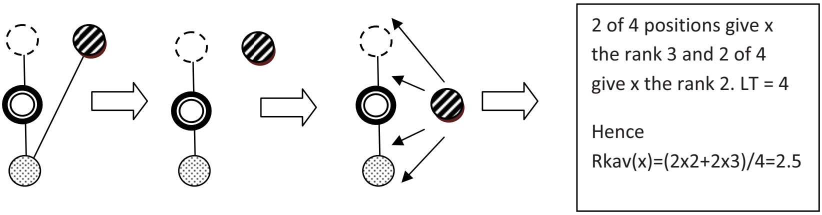
Let x be the object for which the averaged rank is to be determined. Then

1) all objects $y \in X$ with $y||x$ (= U(x)) are considered as isolated
2) the order of all objects comparable to x is extended to obtain a linear order, called the **S-x-P**-chain (Successor - studied object x - Predecessors). The **S-x-P**-chain should not be confused with C(x): C(x) is a partial order, where the objects below or above x must not necessarily be mutually comparable, whereas in **S-x-P**-chain the elements above and below x form specific chains of mutually comparable elements. There are successors, s, for which it is valid that $s \leq x$, and predecessors, p, for which **it is** valid that $x \leq p$
3) the objects of U(x) get positions in the **S-x-P-chai**n due to step 2) and the resulting averaged height of x is calculated in dependence of s and p (step 3) and ||U(x)||.

Although the procedure is explained in more detail elsewhere [18] we exemplify steps 1) to 3) by a simple graphic for the convenience of the reader and show the final formula.

Let x be the object of interest, then –according to the LPOM0-concept:

Rkav(x) = (||S(x)||+1)*(n+1)/(n+1-||U(x)||).



2 of 4 positions give x the rank 3 and 2 of 4 give x the rank 2. LT = 4

Hence Rkav(x)=(2x2+2x3)/4=2.5

Original poset

After steps 1-2)

S-x-P-chain consists of three objects

step 3)

Final calculation

Figure 1: Example for a LPOM0 calculation of object x (symbolized by the vertex ◎) following the three steps explained above. In step 3: the possible heights are 2+2+3+3 = 10. LT = 4, thus Rkav(x) = 10/4 = 2.5

Statistically LPOM0 works astonishingly well (see below) and provides a simple mechanism, to understand how the averaged rank of an object x is obtained. Subsequently Bruggemann et al. [19] introduced some modifications and explained how to improve the approximation. These modifications do improve the result considerably but require a detailed knowledge of the partial order methodology and need advanced programming skills.

3.2.3 Disadvantages of LPOM0

In order to understand the main disadvantages of LPOM0 a simple Hasse diagram should be considered (Fig. 2). Let object e be the object, of which the averaged rank has to be determined. Then the objects u1, u2, u3 are incomparable to object e. In LPOM0 these three objects are considered as isolated and are allowed to get 6 positions. Hence, LPOM0 neglects that the objects u1 and u2 see only three possible positions, instead of 6 and that object u3, sees 5 possible positions including those of u1 and u2. As the probability of being located below object e is reduced in the exact calculation, LPOM0 overestimates the averaged rank and the objects u1, u2 have more influence on the error, compared to the exact method than the object u3.

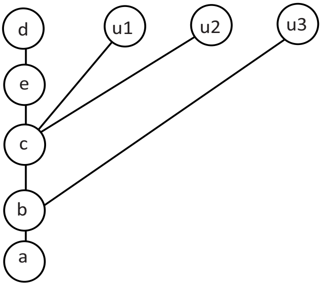


Figure 2: Example of a Hasse diagram (see text)

If we assume u1 and u2 being eliminated, then within the LPOM0 u3 would see 4 positions below and 2 positions above e, whereas in reality u3 would only have 3 positions below and 2 above object e. As in LPOM0 all incomparable objects are considered as equivalent (being supposed as isolated ones) the simple LPOM0 generates many artificial ties in the averaged ranks. Hence, an advanced approximation, which nevertheless is helpful in interpretation of the averaged ranks should take care of:

1) "Tail-effect": Incomparable objects $\in$ U(x) do not necessarily have access to all positions which are possible if predecessors, the interesting object x and its successors are thought of as forming a chain.
2) "Diversity-effect": The incomparable objects do obviously not see the same set of possible accessible positions when merged into the **S-x-P**-chain. There is diversity among the incomparable objects of U(x).
3) Tie-breaking: A reduction of artificial ties of the averaged ranks should be possible.

4. The extended Local Partial Order Model

4.1 Concept

In a given partial order the single objects, x, will be analyzed one after another in order to determine their averaged ranks. Then there is a set U(x) of elements incomparable to x and the total set of objects C(x) comparable to x. Each y $\in$ U(x) sees a range of accessible positions in the inherited partial order of C(x). The merging process by which y is inserted to C(x) depends on those objects, which are incomparable to y and at the same time in C(x). Hence we define two sets:

$P^{<}_{y}$: =O (x) $\cap$U(y), $P^{>}_{y}$: = F(x) $\cap$U(y) with y $\in$ U(x) (8)

The contents of these two sets are of specific importance and we write:

$p^{<}_{y} = || P^{<}_{y}||$ and $p^{>}_{y} = || P^{>}_{y}||$. (9)

By means of (9) the extended LPOM can be formulated as follows:

$$H(x)=\|O(x)\|+\sum_{y\in U(x)}\frac{p_y^<}{p_y^< + p_y^>} \qquad (10)$$

The first part of eq. (10) describes the lowest possible height of x, whereas the sum quantifies the contribution of each object of U(x). The quotient can be interpreted as probability that an object y $\in$ U(x) finds a position below x, which obviously increases the height of x by one unit. So, any object $\in$ U(x) contributes to H(x) according to its probability to be positioned below x. Hence the sum is to be performed over $\frac{p_y^<}{p_y^< + p_y^>}$.

Equation (10) includes LPOM0 as a special case when all objects of all sets U(x), x $\in$ X are thought of as isolated. Clearly, the sum must be considered as an approximation, as it expresses the merging process of objects of U(x) into the **S-x-P**-chain as a step by step process and not as a simultaneous process as it would be if an exact procedure is performed. The difference of H(x) due to equation (10) and an exact value (corresponding to eq. (7)) must therefore be considered as the consequence of the independent performed step-by-step process and can be interpreted as "interaction" among the objects of U(x). To state this more clearly: Merging of one element of U(x) should change the **S-x-P** chain. However, we let it as it is because otherwise the estimation procedure would depend on the sequence.

By eq. (10) we take care of the

- Tail-effect, because indeed only the accessible positions are counted via $p_y^<$
- Diversity-effect, because any object $\in$ U(x) contributes to H(x) by its own individual quotient of the sum in eq. 10.
- Tie-breaking: The main reason to obtain ties in LPOM0 because all elements of U(x) are considered as isolated is now avoided.

Once again, a simple example may help the understanding of the key-equation (10): We take again the Hasse diagram of Fig. 2 as an illustrative example. Let us calculate the averaged rank of object e: There are three elements in U(x), namely u1, u2 and u3. In LPOM0 we would consider them as isolated and there are four positions below e and two positions above object e. According to the three steps of LPOM0 (explained above) we obtain Rkav(e) = 4 + (4/6)*3 = 6. This is, as if we apply eq. 10 but for the special case that all u-elements are considered as isolated. However, u1 "sees" only 1 position below e and 2 positions above e. In Table 1 the "order theoretical" environment of the elements of U(e) is summarized.

U(x)	Below e	Above e
u1	1	2
u2	1	2
u3	3	2

Table 1: Example, see Fig. 2. The number of elements below and above object e are shown for u1, u2 and u3.

Based on Table 1 the evaluation, following the extended LPOM formalism (eq. 10) is performed in the next lines:

O(e) = {a,b,c,e} hence H(e) = 4 + 1/3 (contribution of u1) + 1/3 (contribution of u2) + 3/5 (contribution of u3) (cf. eq. 10), thus H(e) = 5.266. According to PyHasse, module avrank4.py the exact value is 5.19, which should be compared to the result obtained by the simple LPOM0-approximation the value being 6 (vide supra).

4.2 The nature of approximation

4.2.1 Overview

We are aware that the method sketched in section 4.1 is still an approximation. We do not intend to improve LPOMext by the considerations of this section. The main idea of section 4.2 is to find model posets through which we can identify the nature of the approximations implied by LPOMext. The crucial points of LPOMext compared to "reality" are the following:

1. "Combinatorial effect": In Fig. 3 the simplest configuration is shown, where a combinatorial effect appears, which, however, is not taken into account by LPOMext.
2. "One-after-another" in contrast to simultaneous consideration of all linear extensions. This means that in our approach the elements of U(x) contribute to the sum in eq. 10 in an arbitrary order, without considering that elements, already merged are changing the configuration for the next element.
3. "Restrictions": The elements of U(x) may be connected, so that merging one element of U(x) into the **S-x-P**-chain restricts the available positions for other elements of U(x). However, in the LPOMext model a "second" element, u2, will get a position in the **S-x-P**-chain without necessarily preserving its order relation to u1.

4.2.2. Analyzing the crucial point 1 ("combinatorial effect") by model posets (model 1)

In Figure 3 a model poset is shown. The poset is specifically designed to check the combinatorial error because object y can take many positions above object x according to the permutations among Z = {z1, z2,..., zm}, the antichain covering x.

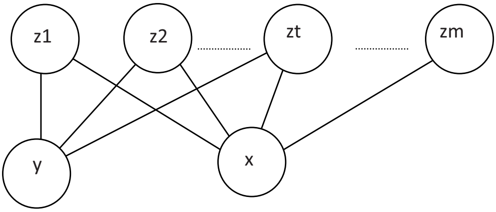


Figure 3: Illustration of combinatorial effects (see text).

The model poset (Fig. 3) is characterized by two structural parameters m and t. The parameter m is the number of elements in the antichain Z={z1,z2,...zt,...,zm} covering the objects x. The parameter t is the number of elements in Z covering y **and** x.

The rank of x is to be determined. According Fig. 3 we assume that object y ∊ U(x) is connected with z1, z2,…,zt, but not with zt+1,…,zm. In order to evaluate this partial order within LPOMext we have to determine those elements of the poset, to which y is incomparable. Accordingly the element y ∊ U(x) sees m-t positions above x. Performing eq. 10 neglects the effect that the number of linear extensions is also affected by the permutations among the elements of Z. This is the reason why the model (Fig. 3) was constructed. The comparison with the exact value will indicate the role of the combinatorial effect (see 4.2.1).

As mentioned above, in the exact evaluation of the poset, shown in Fig. 3, the number of linear extensions with y above x is enhanced by permutations of the m elements covering x and the multiple locations y can take. However, if y is thought of as being below x then only m! different permutations of the elements z1,z2,...,zm are possible. Therefore the expression for the exact averaged rank of x has the following form:

Let LT(y>x) be the number of linear extensions with y above x. This number of linear extensions > m!, because additionally y can take various locations.

Let LT(y<x) be the number of linear extensions if y is located below x. Then LT(y<x) = m!. Therefore, we find:

$$Rkav(x) \sim \frac{1*m!*f+2*m!}{m!*f+m!} \qquad (12)$$

where f is an arbitrary factor, f > 1.

The evaluation of the poset shown in Fig. 3 by LPOMext leads after some steps to the following expression:

$$Rkav(x) = 1 + \frac{1}{2+(m-t)} \qquad (13)$$

As long as the LPOMext-equation does not coincide with the exact equation, we will get an approximation error (see for details below). To get a better feeling for the combinatorial error we simplified the partial order of Fig. 3 (Fig. 4).

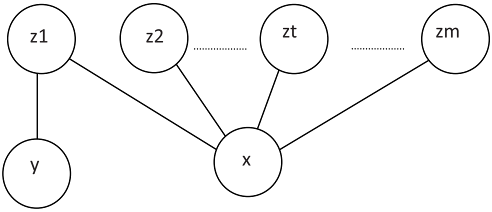


Figure 4: Derivation of a simple formula for an estimation of the combinatorial effect

Assume that the elements z_i are linearly ordered; then there are m! linear orders possible for the antichain Z. If now y is to be located above x then the number of available positions for y depends on the actual location of z1. If z1 has the top position then m positions for y are possible generating m more linear extensions. If z1 is somewhere in the chain of m elements, e.g., it has the height k then the number of permutations, keeping z1 in the height k is :

$$(m-k)!*(k-1)!*\binom{m-1}{k-1}*k \tag{14}$$

The first factor considers the permutations above z1, the second factor those below k, the third factor, expressing k-1 elements out m-1, takes into account that the elements below and above z1 can be exchanged, and the fourth factor considers the positions of y which must be below z1. To get the number of all linear extensions we must sum over k, hence:

$$Rkav(x)=\frac{1*\sum_{k=1}^{m}(m-k)!\cdot(k-1)!\cdot\binom{m-1}{k-1}\cdot k\,+\,2*m!}{\sum_{k=1}^{m}(m-k)!\cdot(k-1)!\cdot\binom{m-1}{k-1}\cdot k\,+\,m!} \tag{15}$$

This equation can be quickly boiled down to the following very simple expression:

$$Rkav(x)=\frac{1*m!*\frac{m+1}{2}+2*m!}{m!*\frac{m+1}{2}+m!}=\frac{m+5}{m+3}=1+\frac{2}{m+3} \tag{16}$$

In the case of t = 1 the unknown factor f is (m+1)/2 (m=1,2,3…) and for the model system of Fig. 4 we can derive an explicit expression for the “combinatorial-type” error:

$$\Delta = Rkav_{exact} - Rkav_{LPOMext} = 1 + \frac{2}{m+2} - (1 + \frac{1}{m+1}) = \frac{m}{m^2 + 3m + 2}$$

As a function of m the error has a maximum around m=4.

If m > 2 and t ≥ 1 (see Fig. 3) then LPOMext underestimates the averaged rank.

Note, if t > 1 analytical expressions can be derived as well, however we suppress the generalized formula here. Instead we summarize the results for some m and t in Table 2.

configuration		t=1	t=2	t=3	t=4	t=5	t=6
m=3	Exact LPOMext LPOM0	1.333 1.25 1.2	1.4286 1.333 1.2	1.5 1.5 1.2			
m=4	Exact LPOMext LPOM0	1.286 1.2 1.167	1.375 1.25 1.167	1.444 1.333 1.167	1.5 1.5 1.167		
m=5	Exact LPOMext LPOM0	1.25 1.167 1.143	1.333 1.2 1.143	1.4 1.25 1.143	1.4545 1.333 1.143	1.5 1.5 1.143	
m=6	Exact LPOMext LPOM0	1.222 1.143 1.125	1.3 1.167 1.125	1.3636 1.2 1.125	1.4167 1.25 1.125	1.4615 1.333 1.125	1.5 1.5 1.125

Table 2: Analysis of model system, Fig.s 3 and 4 (see text).

Each entry of Table 2 contains the exact value obtained from the distributive lattice calculus, the LPOMext-value and the LPOM0-value. The structural parameters m and t are varied.

1. Once again the improvement in LPOMext over LPOM0 is unambiguous. Note that LPOM0 leads to identical values for all the different m, t-configurations.
2. The obtained LPOMext-values are identical for equal values of m-t.
3. LPOMext gets the exact value for t = m. This is understandable, because then x and y have the same order theoretical environment and must get the same value between 1 and 2. As the only position for y is a covering position above x and a covered position below x , the rank of x must be (1+2)/2.
4. The exact values as well as those of LPOMext increase with increasing t values whereas they decrease with increasing m values.
5. LPOMext underestimates the rank in the constellation like that of Fig. 4.

4.2.3 Analyzing the crucial point of type 2 ("one-after-another") by model 2

The basic model partial orders are visualized by the Hasse diagram depicted in Fig. 5.

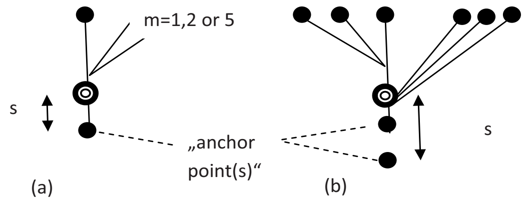


Figure 5: Schematic representations of model partial orders. Object x is represented by: vertex ◎(see text)

Fig. 5a: elements of U(x) have one anchor point [19], s is the number of covering lines between x and the anchor point. We assume one predecessor of x, U(x) with m = 1,2 or 5 elements and 6 successors of x. The elements of U(x) are connected with C(x) at the "anchor point". The anchor point can be one of the six successors of x. Hence varying the anchor point, the measure s (number of covering edges between the anchor point and x) is appropriately varying (cf. Table A, Appendix). It is seen that s does not influence the coincidence of LPOMext with the exact averaged ranks. However, the approximation of LPOM0 is, as expected (*vide supra*) not sensitive to the changes of the partial order as long the number of incomparable elements and the number of successors of x is invariant and the performance of LPOM0 apparently is worsening with increasing |U(x)|. The approximation of LPOM0 is as better as s is increasing, because the quotient Q = accessible positions/all positions of C(x) tends toward 1. "All positions of C(x)" correspond to the interpretation of the elements of U(x) as isolated ones.

Fig. 5b: Here, the elements of U(x) have two anchor points of **S-x-P**. The parameter s is the number of covering lines between x and the lower anchor point. Object x has one predecessor, but now U(x) is split into two groups: One with two elements connected with the element covered by x (U1), and one with three elements whose connection varies in the successor chain of x (U2). The anchor point of U2 is varying and can be one of the six successors of x. The distance of this anchor point to object x is measured by the number of covering edges between the anchor point of U2 and x. In Table B, Appendix the results of Fig. 5b are shown. As above, the LPOM0 model is invariant with respect to the changes in s and the error is decreasing as the quotient Q increases. The data in the second vertical position of each table cell show that LPOMext is no more coincident with the exact values: The elements of U(x) see different order theoretical environments as inserting an element of U(x)

into the **S-x-P**-chain changes the number of accessible positions, so that the next element of U(x) will see a modified **S-x-P**-chain. As already explained, we will not try to improve the approximation of LPOMext by this kind of changes, as the order in which elements are to be merged into the **S-x-P**-chain then becomes important, in contrast to the simultaneous procedure, inherent in the exact calculation. Hence, the name for the error due to “one-after-another” should be changed to “interaction effect” because the sequence how the elements of U(x) are merged into the **S-x-P**-chain becomes important. According to Table B the “interaction effect” becomes as more important as more different the elements of U(x) with respect to their order theoretical environment are.

4.2.4 Analyzing the crucial point of type 3 (“restrictions”), model 3

In Fig. 6 the model-system is shown:

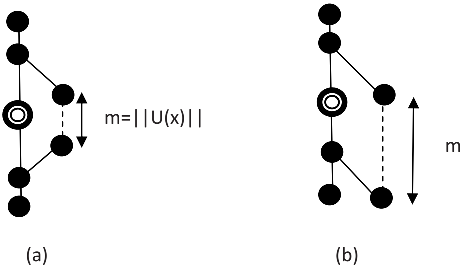


Figure 6: The averaged rank of x (vertex ◎) as a function of the m elements in U(x)

The calculation of the averaged rank of the poset configuration (Fig. 6a) does not pose any difficulty. The averaged rank of x is simply 3+m*3/6 in LPOM0, LPOMext and in the exact calculation (evaluating just eq. 7). Hence, the constraints of this type of partial order is expected to cause little or no error. In the case of configuration b (Fig. 6b) a somewhat different picture develops. Thus, following the three steps, described above for LPOM0 (n=5+m, ||U(x)|| = m-1, ||S(x)|| = 3,or by eq. 10 for LPOMext , we obtain

LPOM0: Rkav(x) = (4*m+24)/7

LPOMext: Rkav(x) = 4 +(m-1)*3/4

The derivation of a closed formula for the exact averaged rank must take into regard that m-1 elements, which are incomparable with x may be arranged above or below x. Therefore the equation for the exact rank becomes somewhat more complex (eq. 17):

$$Rkav_{exact}(x) = \frac{\sum_{k=0}^{m-1}(k+1)\cdot(k+4)^2}{\sum_{k=0}^{m-1}(k+1)\cdot(k+4)} \qquad (17)$$

In Table 3 some values of Rkav for varying m are shown (eq. 17).

m	method	Rkav	m	method	Rkav
2	Exact LPOMext LPOM0	4.7143 4.75 4.571	4	Exact LPOMext LPOM0	6.1667 6.25 5.714
3	Exact LPOMext LPOM0	5.4375 5.5 5.143	5	Exact LPOMext LPOM0	6.9 7 6.286

Table 3. Results of a study, concerning Fig. 6b (model3)

One sees that the LPOMext is closer to the exact values than LPOM0 and that the effect of constraints causes deviations of LPOMext from the exact values in the range of < 0.1 for m ≤ 5. If m →∞ then the slope due to eq. 17 is 3/4 which coincides with that of the LPOMext but disagrees with that of LPOM0. So the study of the model posets, the Hasse diagrams of which are shown in Fig. 6 lead to the interesting conclusion that the approximation error due to the constraint given by the fact that members of U(x) are itself a chain depends on the kind how the elements ∈ U(x) are connected with the **S-x-P**-chain.

4.3 Performance of the extended Local Partial Order Model

Testing the performance of the extended LPOM has two aspects. On the one side it is still of interest to compare LPOMext with the simpler LPOM0, and on the other side it obviously should be compared with the exact values by avrank4.py (module of PyHasse, see sect. 2). The exact values were calculated following the concept of De Loof et al. [25] and the programming realization of Wienand [28] (which was an independent development, see also [29,30]). Twelve data matrices were randomly generated (module randomdm2.py of software PyHasse (see section 5)); some information being summarized in Table 4.

	mean	min	max
n (rows)	19	10	30
m (columns)	3	2	5
P(IB)	0.78	0.43	0.97

Table 4: Characteristics of 12 randomly generated data matrices (abbreviation:. dm)

Hence, the performance analysis can be done for any single randomly generated data matrix. The idea is that the single matrices can be characterized by P(IB) (eq. 6) , number of isolated elements etc. Indeed the Hasse diagrams vary considerably over the 12 matrices so that one can hope to analyze certain deviations of LPOMext from the exact values in terms of characteristics of the 12 matrices. On the other hand the results based on the twelve matrices can be analyzed too as schematically shown in Fig. 7.

4.3.1 Analysis of the pooled data set

In Table 5 the results of a statistical analysis are displayed. The analysis encompasses a linear regression model of the kind

$$\hat{H}(x) = a * H(x)_{model} + b \qquad (18)$$

where $H(x)_{model}$ is the height of object x either by LPOM0 or by the extended model LPOMext (eq. 10).

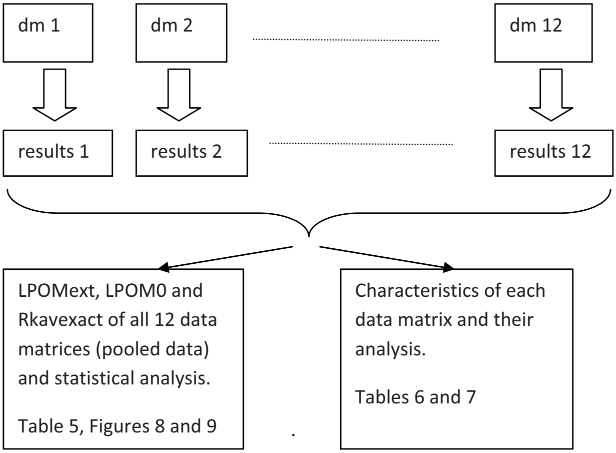

dm 1
dm 2
dm 12
results 1
results 2
results 12
LPOMext, LPOM0 and Rkavexact of all 12 data matrices (pooled data) and statistical analysis.
Table 5, Figures 8 and 9
Characteristics of each data matrix and their analysis.
Tables 6 and 7

Fig. 7: Strategy of section 3: 12 randomly generated data matrices are analyzed and compared with general characteristics of posets like P(IB), length, etc..

The dependent variable, $\hat{H}(x)$, is provided by the exact values (PyHasse module avrank4.py). The regression coefficients a and b be should ideally be 1 and 0, respectively. Beside the regression coefficients a and b the squared correlation coefficient (corrected for freedom) r^2_{DF} is reported.

	a	b	r^2_{DF}
LPOM0	0.818	2.205	0.963
LPOMext (eq.(9))	0.897	1.229	0.988

Table 5: Statistical results of the pooled data set (number of cases: 230)

From the data presented in Table 5 it is obvious that both models still have a bias: Especially the regression coefficient b deviates remarkably from 0 in the case of LPOM0 but also to a lesser extent in the case of LPOMext. The general trend (expressed by the correlation coefficient) is, however, very good in both cases, showing that the essential mechanism in getting the averaged rank is pretty well modeled. Furthermore, we see that the extended model improves the result. In Fig. 8 scatter plots of the exact values versus LPOM0 and LPOMext are depicted.

From Fig. 8 it is clear that one task, i.e., the ‘tie-breaking’ is fulfilled to a high degree. Thus, there are many point clouds which are vertically arranged, i.e., have different exact values, but the same estimated averaged rank due to LPOM0, whereas this is not seen to the same extent in the case of LPOMext.

To further analyze the performance of the two models, LPOM0 and LPOMext, we looked at the absolute errors introduced by the two models, respectively. Thus, in Fig. 9 the absolute error $\delta_1 = |H_{exact}(x) - H_{LPOM0}(x)|$ and $\delta_2 = |H_{exact}(x) - H_{LPOMext}(x)|$, respectively are displayed.

Fig. 9 demonstrates how drastically the method error is reduced if the extended LPOM is applied. For example the mean value of the absolute errors of LPOM (δ_1) is 1.27 whereas that of LPOMext (δ_2) is 0.67, respectively. However, Fig. 9 also visualizes that obviously there are still deviations from the exact values even applying the LPOMext model although they are significantly reduced. These deviations may, after section 4.2 be the combined effect of the three types of errors.

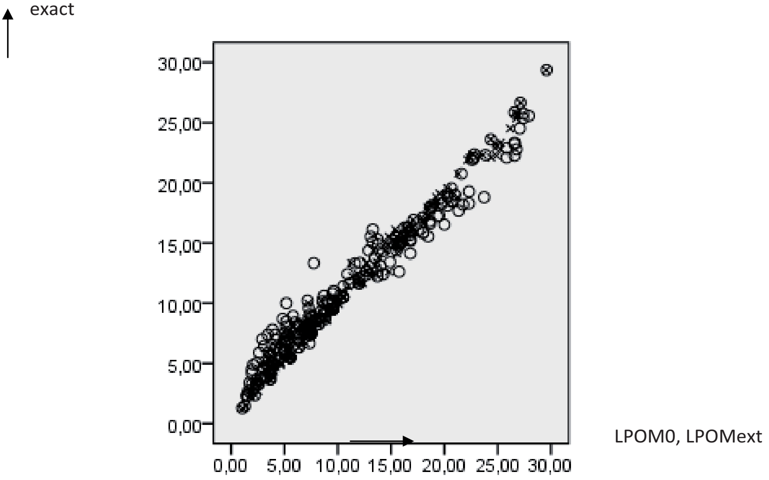


Figure 8: Ordinate: the exact values, abscissa: LPOMext (crosses) and LPOM0 (circles))

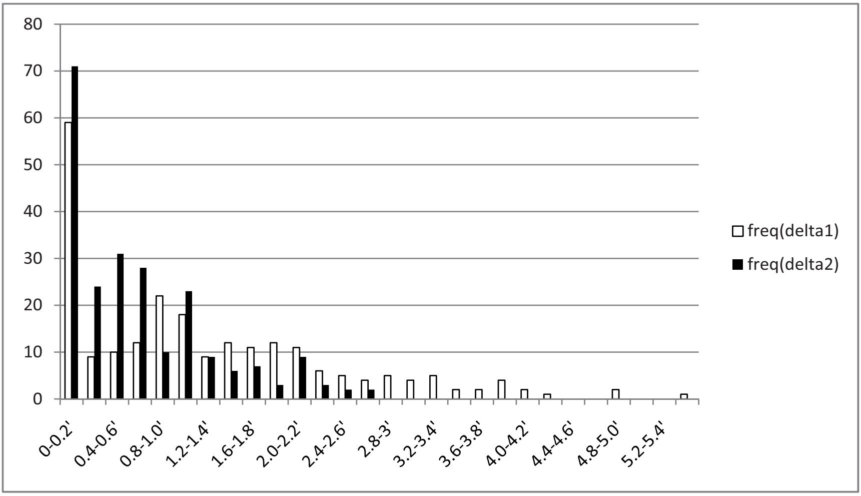


Figure 9: Histogram of the absolute errors δ_1 (white: LPOM0) (<5.6) and δ_2 (black: LPOMext) (<3.0). The ordinate is the frequency and the abscissa represents the error in the corresponding range (for example 1.6≤ δ_I <1.8.

To summarize, the above has unambiguously displayed that the extended local partial order model, LPOMext, does not only reproduce in an overall view (over all 230 cases) the exact values as Fig. 8 and Table 5 show, but also the method’s error is significantly reduced (cf. Fig. 9). In order to see the role of the diversity and tail-effect we now turn to the analysis of the single Hasse diagrams.

4.3.2 Analysis due to 12 randomly generated data matrices

We perform a statistical analysis similar to that reported in sect. 4.3.2, where (1) the aim is to show in which settings of partial order the deviations between the two models are striking and in which settings they are not (Table 6b), and where (2) the results referring some of the 12 data matrices in detail are presented (Table 7).

In Table 6 a) and b) the results based on the twelve single matrices are shown. The parameters n and m are the dimensions of the single data matrix (n: number of rows, m number of columns) , Length is the length of the maximum chain (see above), n(isolated) is the number of objects which are isolated. In Table 6 b) we will not render the correlation coefficients (exact vs LPOM0) and (exact vs LPOMext) as the differences are small and because our main focus is the bias. Instead we analyze the regression equations of type (18). Ideally the slope a and the intercept b should obviously be 1 and 0, respectively.

Case	n	m	P(IB)	length	nisol:= n(isolated)/n
1	18	5	0.96	1	0.56
2	20	4	0.90	2	0.25
3	14	3	0.79	2	0.00
4	30	3	0.75	6	0.13
5	18	4	0.84	3	0.17
6	10	3	0.84	2	0.20
7	10	4	0.96	1	0.60
8	14	5	0.97	1	0.71
9	28	3	0.62	4	0.00
10	23	2	0.43	4	0.00
11	25	2	0.53	7	0.00
12	20	3	0.76	5	0.05

Table 6 a): Quantities of the single-data matrices analysis, n and m are general parameters

Case	a=slope (LPOM0)	b=intercept (LPOM0)	a=slope (LPOMext)	b=intercept (LPOMext)
1	0.78	2.45	0.87	1.52
2	0.78	2.37	0.82	2.07
3	0.84	2.17	0.91	0.91
4	0.78	3.97	0.86	2.47
5	0.83	2.05	0.89	1.32
6	0.89	0.71	0.94	0.41
7	1.00	0.00	1.00	0.00
8	0.82	1.52	0.88	1.02
9	0.70	3.77	0.84	2.26
10	0.83	1.76	0.93	0.72
11	0.90	1.33	0.94	0.60
12	0.75	2.85	0.83	1.92

Table 6 b): Estimates slopes and intercepts

The statistical analysis relating the regression coefficients of both models with simple parameters of any partial order like Length , P(IB) or the fraction of isolated objects (nisol) did not lead to any satisfying relation as seen from the Pearson-correlation that are summarized in Table7:

	a (LPOM0)	b (LPOM0)	a (LPOMext)	b (LPOMext)	P(IB)	length	nisol
a(LPOM0)		-0.898	0.927	-0.889	0.170	-0.239	0.326
b(LPOM0)			-0.825	0.943	-0.229	0.433	-0.418
a(LPOMext)				-0.925	-0.084	-0.145	0.175
b(LPOMext)					-0.023	0.320	-0.254
P(IB)						-0.749	0.789
length							-0.713

Table 7: Pearson correlations of characteristic quantities of the 12 partial orders (P(IB), length and nisol) and of slopes and intercepts of the regression equations (18) for LPOM0 and POMext

Table 7 shows that LPOM0 is the better the more the posets include isolated elements, which, however is not surprising as this is in agreement with the general assumption in the LPOM0 model (*vide supra*). The simple characteristics P(IB) , length and nisol do not have any explanatory potential for the quality of the model LPOMext in terms of slope and intercept. Slope and intercepts are significantly (at level 0.01) negatively correlated.

4.4Critical discussion of the findings in section 4

Whereas the statistical examination shows the pretty good performance, some care was taken to check typical model systems and to see how the approximations due to LPOM0 or LPOMext are performing. It seems as if the "combinatorial error" has the most influence, followed by the "one-after-another" error and finally by the constraints-error. Rank inversions occur mainly due to the combinatorial effect. Clearly this way to argue can only be qualitative as the typical structures discussed in section 4.2 are rarely fulfilled in empirical posets. However we think that a systematical study of model posets supported by a Monte Carlo-simulation of all structural parameters (like m, t) will be of great help to derive new approximations and - given an approximation - to discuss and fix its shortcomings.

5. Real data matrix

In recent years we have been studying the possible influence on the environmental and human health by the rocket fuel heptyl (1,1-dimethyl hydrazine) and its transformation products [31-33]. When heavy rocket carriers are launched from the Baikonur Cosmodrome the burned-out stages are drop over terrestrial areas (fall zones). Thus, a significant physical influence to the environment due to the fall of metal scrap is accompanied by a chemical influence as significant amounts (up to 2 tons) of residual, unburned, fuel is present, part of which is distributed in the environment as a result of the fall [31]. The damaging effects of 1,1-dimethyl hydrazine is well substantiated but a series of studies revealed that a series of transformation products of the fuel may be equally toxic and damaging to the environment. Based on the QSAR approaches PASS (Prediction of Activity Spectra for Substances) and ADME (Absorption, Distribution, Metabolism, Excretion)/Tox Boxes [31-33] we studied the possible influence of the single compounds on the human health [33] and applied partial order ranking to elucidate the more problematic compounds on a cumulative basis. In the present paper we apply these data as a real life example to elucidate the advantageous use of the new LPOMext approximation compared to the original LPOM0 model.

Id	*Chemical*	*CAS No*	*Carcino genic*	*Muta genic*	*Terato genic*	*Embryo toxic*
1	1,1-Dimethyl hydrazine	57-14-7	0.955	0.762	0.689	0.672
2	Trimethyl amine	75-50-3	0.619	0.25	0.25	0.527
3	Dimethyl amine	124-40-3	0.25	0.25	0.563	0.25
4	1,1,4,4-Tetramethyl tetrazene	6130-87-6	0.894	0.792	0.946	0.816
5	N-Nitroso dimethyl amine	62-75-9	0.98	0.969	0.952	0.866
6	N,N-Dimethyl formamide	68-12-2	0.951	0.25	0.614	0.795
7	Tetramethyl hydrazine	6415-12-9	0.827	0.539	0.698	0.604
8	Acetaldehyde dimethyl hydrazone	7422-90-4	0.98	0.25	0.25	0.25
9	Formaldehyde dimethyl hydrazone	2035-89-4	0.683	0.25	0.25	0.25
10	Trimethyl hydrazine	1741-01-1	0.923	0.619	0.811	0.681
11	Acetaldehyde	75-07-0	0.628	0.25	0.25	0.25
12	1-Formyl 2,2-dimethyl hydrazine	3298-49-5	0.897	0.524	0.53	0.25
13	Dimethylamino acetonitrile	926-64-7	0.25	0.25	0.25	0.25
14	Ammonia	7664-41-7	0.25	0.25	0.25	0.25
15	Hydrogen cyanide[b)]	74-90-8	-	-	-	-
16	1,3-Dimethyl-1*H*-1,2,4-triazole	16778-76-0	0.25	0.25	0.25	0.25
17	1-Methyl-1*H*-1,2,4-triazole	6086-21-1	0.25	0.25	0.25	0.25
18	1-Methyl-1*H*-pyrazole	930-36-9	0.25	0.25	0.25	0.25

[a] Values given are the calculated probability for the compounds to exhibit the single effects. Probabilities calculated to be below 0.5 are arbitrarily put to 0.25 for calculation purposes. Note the chemicals 13-18 are equivalent, hence only chemical no 13 appears in Fig. 10.

[b)]Data not available

Table 8: PASS predictions of selected biological activities[a]

No	*Ames*	*Blood*	*Cardiovascular*	*Gastrointestinal*	*Kidney*	*Liver*	*Lungs*
1	0.899	0.57	0.4	0.65	0.28	0.48	0.34
2	0.266	0.44	0.34	0.8	0.2	0.18	0.27
3	0.258	0.2	0.31	0.26	0.11	0.2	0.2
4	1	0.79	0.07	0.92	0.57	0.85	0.74
5	0.999	0.76	0.06	0.97	0.75	0.93	0.71
6	0.367	0.27	0.12	0.65	0.14	0.05	0.4
7	0.631	0.52	0.33	0.83	0.19	0.1	0.17
8	0.864	0.63	0.06	0.84	0.31	0.05	0.75
9	0.848	0.32	0.08	0.9	0.42	0.07	0.72
10	0.757	0.53	0.64	0.66	0.14	0.29	0.29
11	0.327	0.19	0.08	0.25	0.09	0.04	0.04
12	0.84	0.48	0.14	0.71	0.15	0.28	0.42
13	0.219	0.47	0.21	0.89	0.18	0.12	0,47

Table 9: Predicted probabilities for the compounds to exhibit positive Ames test and adverse organ specific health effects by the ToxBoxes module of the ADME/Tox WEB

5.2 Partial order analysis

In Fig. 10 the two Hasse diagrams are shown a) based on data of Table 8 and b) based on Table 9. In both cases with all the attributes.

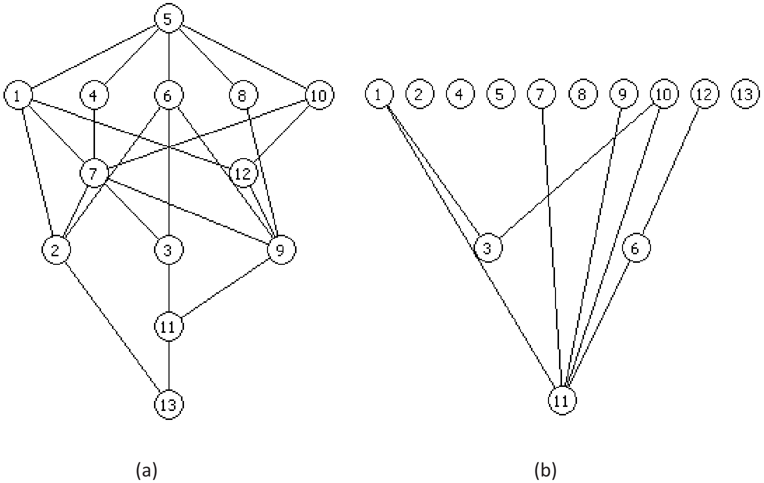


Figure 10: a: Hasse diagram based on Table 10, b: Hasse diagram based on Table 11

Tables 10 and 11 show the results of LPOM0, LPOMext and of the exact calculation via the distributive lattice.

Chemicals	LPOM0	LPOMext	Exact values
1	10.889	10.3	10.005
2	3.111	3.843	3.929
3	3.111	3.843	3.929
4	10.889	10.433	10.202
5	13.0	13	13
6	10.5	9.5	9.186
7	8.4	7.6	7.404
8	9.333	8.5	8.537
9	3.5	3.8	4.075
10	11.2	10.917	10.517
11	2.333	2.4	2.537
12	7.0	6.583	6.677
13	1.0	1	1

Table 10: General information to LPOM0, LPOMext and exact values , based on data of Table 8. Ranks counted from the bottom upwards (in contrast to e.g. Carlsen [34])

The mean value of $|Rkav_{exact} – Rkav_{LPOM0}|$ = 0.623 is clearly worse than that of $|Rkav_{exact} – Rkav_{LPOMext}|$ = 0.166 showing once again the good performance of LPOMext. However, a rank inversion is noted: Exact and LPOM0: ….12 > 9 > 2 ~ 3>11… whereas LPOMext: 12 > 2 ~ 3 > 9 > 11>…. In LPOM0 we find an additional tie, namely 4 ~ 1, which is not justified by the symmetries of the original poset, shown in Fig. 10a. Turning to the second case (cf. Fig. 10b) a somewhat different picture develops as illustrated by the figures given in Table 11. The mean values of the absolute differences $|Rkav_{Exact} – Rkav_{LPOM0}|$ and $|Rkav_{Exact}\text{-}Rkav_{LPOMext}|$ are in this case now closer to each other (0.609 and 0.257) . The reason is evident: The Hasse diagram (Fig. 10b) has significantly more isolated elements so that the basic assumption of LPOM0 is pretty often fulfilled and thus the original LPOM0 model is performing better (cf. discussion above). In Table 12 the ranks obtained from the two approximative models are listed together with the exact values:

Chemicals	LPOM0	LPOMext	Exact	Chemicals	LPOM0	LPOMext	Exact
1	10.50	9.917	9.10	7	9.33	8.50	8.01
2	7.00	7.00	7.00	8	7.00	7.00	7.00
3	3.50	3.75	3.33	9	9.33	8.50	8.01
4	7.00	7.00	7.00	10	10.50	9.92	9.10
5	7.00	7.00	7.00	11	1.75	1.79	2.02
6	7.00	6.33	6.015	12	10.50	10.17	10.01
				13	7.00	7.00	7.00

Table 11: Results of LPOM0, LPOMext and exact calculation based on Table 9. Ranks counted from the bottom upwards (in contrast to e.g. Carlsen [34])

Exact	LPOM0	LPOMext
12	12,1,10	12
1, 10	9,7	1,10
9,7	2,4,5,6,8,13	9,7
2,4,5,8,13	3	2,4,5,8,13
6	11	6
3		3
11		11

Table 12: Ranks of 13 chemicals due to three different methods, tied chemicals are put into the same cell (based on the data of Table 9).

Once again one can see that LPOM0 has a low discrimination power as more ties are found, whereas it is seen that the LPOMext model perfectly reproduces the exact values.

A further application of averaged ranks by the methods demonstrated in this paper is the Hierarchical Partial Order Analysis (HPOR) introduced by Carlsen, [35]. However this is out of the scope of this paper.

6. Conclusions, Outlook

In the present paper we have, in addition to the theoretical background for the LPOMext approach shown an example from our current research. The results clearly demonstrate the superiority of the LPOMext model in comparison to the original approximate LPOM0 model. Are there structures of partial orders which typically cannot described by LPOMext? Certainly yes, as the only parameters of the extended models are $p^{<}_{y}$, $p^{>}_{y}$ and $||U(x)||$ and any configuration which is not describable by these three parameters is a candidate where LPOMext may fail. An example are structures of partial orders, such as those those denoted as “restrictions” in section 4.2.1. Obviously, the errors are not as large as may be expected. However further developments must be extended beyond the three-parameters description.

The averaged ranks derived from the partial order are important tools supporting decisions, as their derivation does not need to find weights by which attributes can be (linearly) combined. We do not claim that the averaged ranks are the “truth”, however they are useful in checking linear orders obtained from decision support tools.

What methods can we offer if a concrete problem is to be solved? The answer may be given in form of a scheme (Fig. 11). The relevant point is that not only the number of objects, n, is the determining quantity by which a selection of the methods can be performed, but also the number of all incomparabilities U. The larger U, the more permutations must be checked as more the computing time goes with n!.

The methods which are available are:

- Distributive lattice method (DLM), the exact calculation of the averaged ranks
- Bubley-Dyer Monte Carlo Markov chain method (BD) [36]
- Local Partial Order Model 0 (LPOM0) and
- Local Partial Order model, extended (LPOMext).

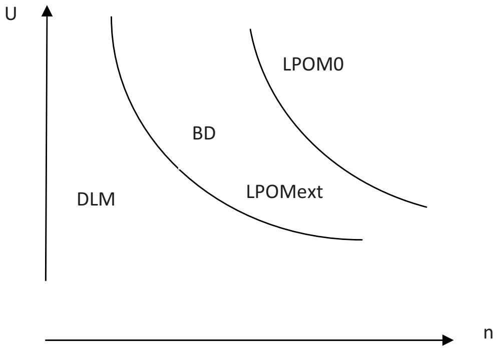


Figure 11: Application range for four methods. n: number of objects, U number of incomparabilities

If DLM can be applied then this exact method is clearly the best one. Its limits are sketched by hyperbolic like curve, and to give it a more concrete meaning: A calculation applying the module avrank4.py of PyHasse needs less than 1 minute for 35 objects and U = 131. When, however a data matrix with 50 rows and U = 516 is considered then the computation needs around 16 minutes (and analyzes more than 10^{30} paths in the lattice) , hence the critical P(IB)-value may be 0.4. If P(IB) of an actual problem is below 0.4 then DLM can be used instantly. The same data matrix analyzed with BD (10^6 MonteCarlo-simulations) took around 10 minutes. LPOM0 and LPOMext do the calculation immediately. However both methods are approximations. So why should we use LPOM0 and why LPOMext? We think that for screening purposes i.e. for getting a quick overview both approximations are recommendable. One main advantage of LPOM0 and furthermore of LPOMext is that the factors determining the final averaged rank of an object are well defined and understandable. The advantage of LPOMext over LPOM0 is its reduction of ties.

So, how we will proceed in the future? One main idea is, what was already discussed in section 4, to introduce model posets in order to analyse the main factors leading to deviations from the exact rank. In order to do this, several simple posets were selected, whose averaged rank can be determined by an explicit formula. It is hoped that graph-theoretical structures in the directed graph which represents the partially ordered set can be identified, in order to derive sharper limits for the applicability of local partial order models.

Another idea was already formulated in the paper of Patil and Taillie [37], who derive the linear order not by the detour of first calculating averaged ranks and from them the linear (or weak) order, but directly. Hence, an important task in the next future will be to derive computationally tractable concepts for a direct derivation of linear or weak orders.

References

[1] P. Basieux, *Die Architektur der Mathematik - Denken in Strukturen,* Rowohlt Taschenbuch Verlag, Hamburg, 2000.

[2] B. A. Davey, H. A. Priestley, *Introduction to Lattices and Order,* Cambridge Univ. Press, Cambridge, 1990.

[3] P. B. Sørensen, R. Bruggemann, M. Thomsen, How to guide and assess risk reduction using risk characterization indicators. *Am. J. Appl. Sci.* **6** (2009) 1255–1263.

[4] K. Voigt, J. Gasteiger, R. Bruggemann, Comparative evaluation of chemical and environmental online and CD–ROM databases. *J. Chem. Inf. Comp. Sci*. **40** (2000) 44–49.

[5] R. Bruggemann. K. Voigt, Basic principles of Hasse diagram technique in chemistry, *Comb. Chem. High Throughput Screen.* **11** (2008) 756–769.

[6] S. Kardaetz, T. Strube, R. Bruggemann, G. Nützmann, Ecological scenarios analyzed and evaluated by a shallow lake model, *J. Environ. Manag.* **88** (2008) 120–135.

[7] P. B. Sørensen, B. B. Mogensen, S. Gyldenkaerne, A. G. Rasmussen, Pesticide leaching assessment method for ranking both single substances and scenarios of multiple substance use, *Chemosphere* **36** (1998) 2251–2276.

[8] P. Annoni, Different ranking methods: potentialities and pitfalls for the case of European opinion poll, Environ. Ecol. Stat. **14** (2007) 453–471.

[9] M. Pavan, A. Mauri, R. Todeschini, Total ranking models by the genetic algorithm variable subset selection (GA–VSS) approach for environmental priority setting, *Anal. Bioanal. Chem*. **350** (2004) 430–444.

[10] M. Pavan, V. Consonni, R. Todeschini, Partial ranking models by genetic algorithm variable selection (GAVSS) approach for environmental priority settings, *MATCH Commun. Math. Comput. Chem*. **54** (2005) 583–609.

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

[12] B. Roy, The outranking approach and the foundations of the ELECTRE methods. in: C. A. Bana e Costa, (Ed.), *Readings in Multiple Criteria Decision Aid*, Springer–Verlag, Berlin, 1990, pp. 155–183.

[13] M. Roubens, Preference relations on actions and criteria in multicriteria decision making, *Eur. J. Oper. Res.* **10** (1982) 51–55.

[14] G. Munda, *Social Multi–Criteria Evaluation for a Sustainable Economy*, Springer–Verlag, Berlin, 2008.

[15] E. Halfon, M. G. Reggiani, On ranking chemicals for environmental hazard, *Environ. Sci. Technol*. **20** (1986) 1173–1179.

[16] R. Bruggemann, R., E. Halfon, G. Welzl, K. Voigt, C. Steinberg, Applying the concept of partially ordered sets on the ranking of near–shore sediments by a battery of tests, *J. Chem. Inf. Comp. Sci.* **41** (2001) 918–925.

[17] R. Bruggemann, K. Voigt, Analysis of partial orders in environmental systems applying the new software PyHasse, in: J. Wittmann, M. Flechsig (Eds.), *Simulation in Umwelt- und Geowissenschaften,* Potsdam 2009. Shaker–Verlag, Aachen, 2009, pp. 43–55.

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

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

[20] R. Bruggemann, K. Voigt, Stability of comparative evaluation, – example: Environmental databases, *Chemosphere* **33** (1996) 1997–2006.

[21] W. T. Trotter, *Combinatorics and Partially Ordered Sets Dimension Theory*, The Johns Hopkins Univ. Press, Baltimore, Maryland, 1992.

[22] F. Harary, *Graphentheorie*, R. Oldenbourg-Verlag, München, 1974.

[23] R. Bruggemann, C. Bücherl, S. Pudenz, C. Steinberg, Application of the concept of partial order on comparative evaluation of environmental chemicals, *Acta Hydrochim. Hydrobiol*. **27** (1999) 170–178.

[24] R. P. Stanley, *Enumerative Combinatorics Volume I.* Wadsworth & Brooks/cole, Monterey, 1986.

[25] K. De Loof, B. De Baets, H. De Meyer, R. Bruggemann, A Hitchhiker's guide to poset ranking, *Combin. Chem. High Throughput Screen.* **11** (2008) 734–744.

[26] K. De Loof, H. De Meyer, B. De Baets, Exploiting the lattice of ideals representation of a poset, *Fund. Informat.* **71** (2006) 309–321.

[27] K. De Loof, *Efficient Computation of Rank Probabilities in Posets*, Dissertation, Univ. Gent. 2009.

[28] O. Wienand, 2003. http://bio.math.berkeley.edu/ranktests/lcell

[29] J. Morton, L. Pachter, A. Shiu, B. Sturmfels, O. Wienand, Convex rank tests and semigraphoids, *SIAM J. Discr. Math*. **23** (2009) 1117–1134.

[30] J. Morton, L. Pachter, A. Shiu, B. Sturmfels, O. Wienand, 2006. Geometry of Rank Tests. http://arvix.org/abs/math/0605173 -:1-9.

[31] L. Carlsen, O. A. Kenesova, S. E. Batyrbekov, A preliminary assessment of the potential environmental and human health impact of unsymmetrical dimethyl-hydrazine as a result of space activities, *Chemosphere* **67** (2007) 1108–1116.

[32] L. Carlsen, O. A, Kenesova, S. E. Batyrbekova, A QSAR/QSTR study on the environmental health impact by the rocket fuel 1,1–dimethyl hydrazine and its transformation products, *Environ. Health Insights* **1** (2008) 11–20.

[33] L. Carlsen, O. A. Kenesova, S. E. Batyrbekova, A QSAR/QSTR study on the human health impact by the rocket fuel 1,1–dimethyl hydrazine and its transformation products, *Environ. Toxicol. Pharmacol.* **27** (2009) 415–423.

[34] L. Carlsen, Partial order ranking of organophosphates with special emphasis on nerve agents, *MATCH Commun. Math. Comput. Chem*. **54** (2005) 519–534.

[35] L. Carlsen, Hierarchical partial order ranking, *Environ. Poll.* **155** (2008) 247–253.

[36] R. Bubley, M. Dyer, Faster random generation of linear extensions, *Discr. Math.* **201** (1999) 81–88.

[37] G. P. Patil, C. Taillie, Multiple indicators, partially ordered sets, and linear extensions: Multi–criterion ranking and prioritization, *Environ. Ecol. Stat.* **11** (2004) 199–228.

Appendix

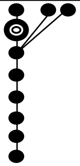
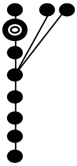
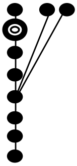
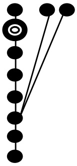
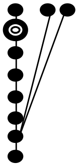
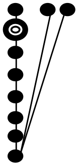
configuration	scheme (m=2)	m=1	m=2	m=5
s=1 exact LPOMext LPOM0		7.33 7.33 7.78	7.67 7.67 8.56	8.67 8.67 10.89
s=2 exact LPOMext LPOM0		7.5 7.5 7.78	8.00 8.00 8.56	9.50 9.50 10.89
s=3 exact LPOMext LPOM0		7.6 7.6 7.78	8.20 8.20 8.56	10.00 10.00 10.87
s=4 exact LPOMext LPOM0		7.67 7.67 7.78	8.33 8.33 8.56	10.33 10.33 10.89
s=5 exact LPOMext LPOM0		7.71 7.71 7.78	8.43 8.43 8.56	10.57 10.57 10.89
s=6 exact LPOMext LPOM0		7.75 7.75 7.78	8.50 8.50 8.56	10.75 10.75 10.89

Table A: Results referring to Fig. 5a. Each entry of the table contains the exact, the LPOM0 and the LPOMext-value, arranged in three vertical positions) (see text).

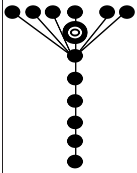
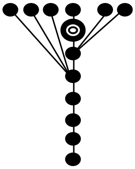
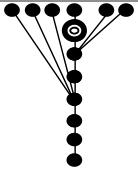
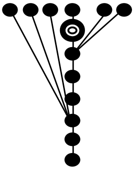
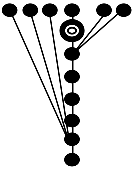
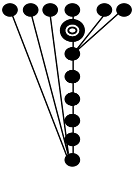
configuration	scheme	approach	averaged ranks
s=1		Exact LPOMext LPOM0	8.67 8.67 10.89
s=2		Exact LPOMext LPOM0	9.00 9.17 10.89
s=3		Exact LPOMext LPOM0	9.23 9.47 10.89
s=4		Exact LPOMext LPOM0	9.42 9.67 10.89
s=5		Exact LPOMext LPOM0	9.56 9.81 10.89
s=6		Exact LPOMext LPOM0	9.67 9.92 10.89

Table B: Results related to Fig. 5b (m=|U(x)|=5) (see text).