The building blocks of complexity: a unified criterion and selected applications in economics and finance[1]

Seese, D.[*] Schlottmann, F.[* and **]
[*]Institute AIFB [**]GILLARDON AG financial software[2]
University Karlsruhe Alte Wilhelmstraße 4
D-76128 Karlsruhe D-75015 Bretten
Germany Germany
Tel: +49 (0) 721 608 – {6037 | 6557}
E-mail: {seese | schlottmann}@aifb.uni-karlsruhe.de

ABSTRACT

The analysis of many complex problems and complex dynamic systems shows that there are dependencies between high complexity and the existence of large grids in the underlying structures containing inhomogeneous elements and including an irregular flow of information. This complexity criterion is formulated in a precise way and analysed in different areas of application, particularly for selected problems from economics and finance. We discuss how our criterion can be applied to make complex problems more tractable by exploring structural parameters to control the complexity of problems and systems in complexity engineering. In some areas the criterion is provable in a strong mathematical sense, whereas in others it is confirmed by numerous examples, without finding a counterexample. The areas of application cover: complexity theory, design of efficient algorithms, dynamic systems, chaos theory, neural networks, auctions, capital markets, portfolio credit risk and operational risk management.

1. INTRODUCTION

The growing complexity of many real world problems is one of the biggest challenges of our time. Besides the high complexity even of single products or systems and the corresponding complexity of the technologies to develop these components, the problems usually get more difficult e. g. by the globalisation of companies and competition, the quick and ubiquitous flow of information through the growing World Wide Web (transport systems, communication systems, etc.), the appearance of electronic commerce, the speed up of time to market (i. e. from the first ideas to the final products) as a necessity to survive in business, the pure size and the number of components and development steps for new products.

---

[1] This paper is supplementary material for a series of talks which the first author gave in May 2002 at the School of Finance and Economics of the University of Technology in Sydney, at the School of Mathematics and Statistics of the University of Sydney and at the Sydney Financial Mathematics Workshop. The material is based on the two original articles Seese & Schlottmann (2002) and Schlottmann & Seese (2002).

[2] Partial support from GILLARDON AG financial software is gratefully acknowledged. The views expressed in this paper are results of independent research and do not necessarily reflect the views of GILLARDON AG financial software.

Moreover, it seems that managers and politicians confronted with real life situations and/or systems of growing complexity fail too often, and many problems are too difficult to expect correct or at least adequate decisions, see e. g. Tschoegl (2000) and King (2001) for an overview and an analysis of well-known historic debacles in the financial industry. The problem is that the reason for this is not a lack of intelligence – usually managers are highly educated and are supported by sophisticated tools and qualified staff.

One of the main problems in complexity research is quite fundamental – it is the lack of a precise definition or a useful criterion that helps us to decide whether a problem, situation, or system has to be called complex. Usually such complex problems, situations, or systems are defined by giving a collection of properties characterising them as complex. The typically used properties for describing complex entities are:

> *inestimable, incalculable, confusing, highly connected, having their own momentum, non-transparent, devious, depending on probabilities, unstable, depend on non-linear processes, having a very large number of parts connected together in a particular pattern, being difficult to understand or to explain because there are many different aspects or people involved*.

For instance, Carkhuff et al. (2000) define complexity as

> *the inability to relate to or represent the interdependent process and processing relationships within and between systems.*

It is not the goal of this paper to discuss all possible definitions of different aspects of complexity - especially for computer science many formal definitions that are trying to cover the aspect of the interaction between machines or agents with bounded resources in a problem solving process can be found in the literature (see e. g. Papadimitriou (1994)). Instead, we will present a structural criterion which seems to be responsible for high complexity of many theoretical and practical problems. The basic question is whether there exists a reason for high complexity of problems and the behaviour of systems which can be explained by an examination of the input structures to the problem, the internal structure of the regarded system or the necessary communication structure of the methods that solve the regarded problems algorithmically. To get a possible answer to this question, we have identified a uniform criterion within the huge diversity of theoretical and real life complex problems. The goal of this paper is to present our ideas, arguments and examples together with some proposals how to make the criterion applicable to complexity engineering, particularly for economic and financial problems.

The paper is organised as follows. In the next section we will develop the basic idea of the criterion by discussing NP-hard problems and tiling problems. The theoretical foundation for the structural ideas is presented in section 2, where the necessary definitions of graphs, grids and graph minors together with related main results are presented. In section 3 we build a bridge to the control of dynamic systems. Section 4 presents a collection of some examples from different areas of application: auctions, capital markets, credit risk and operational risk management. The paper ends with concluding remarks and references.

1. COMPUTATIONAL COMPLEXITY AND LARGE GRIDS

When we are thinking about computational complexity, NP-hardness is one of the most famous notions. Many algorithmic problems in real world applications are NP-hard (see Garey & Johnson (1979)) and their NP-hardness is the preferred argument that is given if we are not able to find an efficient solution to a problem and propose a heuristic approach instead of an exact algorithm. By definition, a problem is NP-hard if each problem in the complexity class NP, i. e. all problems solvable by a non-deterministic algorithm in polynomial time, are reducible to it in polynomial time. A problem is NP-complete if it is NP-hard and inside the class NP. Hence, the NP-complete problems are the hardest problems among all those that can be solved by a non-deterministic algorithm in polynomial time and until now there has not been any solution for such a problem by a deterministic polynomial time algorithm (polynomial in the size of the input).

To get a feeling how structure influences the complexity of a problem we first look at computational complexity of decision problems for graphs. Usually a property P of graphs is given here and the question is for an arbitrarily given graph G, whether G has the property P or not. Examples of such properties are:

PLANARITY
instance: graph G
question: Does G have an embedding without edge-crossings in the Euclidean plane?

HAMILTONIAN CIRCUIT
instance: graph G
question: Does G contain a Hamiltonian circuit, i. e. a subgraph of G which is a circuit containing each vertex of G?

While the left problem can be solved in linear time for all graphs, the right one is a standard example of a NP-complete problem, for which no polynomial time solution has been found until now. In Papadimitriou (1994) a profound introduction to computational complexity of decision problems is given. Many problems are investigated with respect to their complexity, e. g. in Garey & Johnson (1979), Downey & Fellows (1999) and Brandstaedt et al. (2000), and many of them are interesting for real world business applications. There are different attempts to make a complex decision problem tractable by restricting the class of problem instances to graphs with a special structure, e. g. to planar graphs or trees instead of the class of all graphs (see section 3 for definitions). It is a surprising observation that almost all NP-complete problems remain NP-complete for almost all restrictions of the input, with the exception of structures closely related to trees (see figure 1 below).

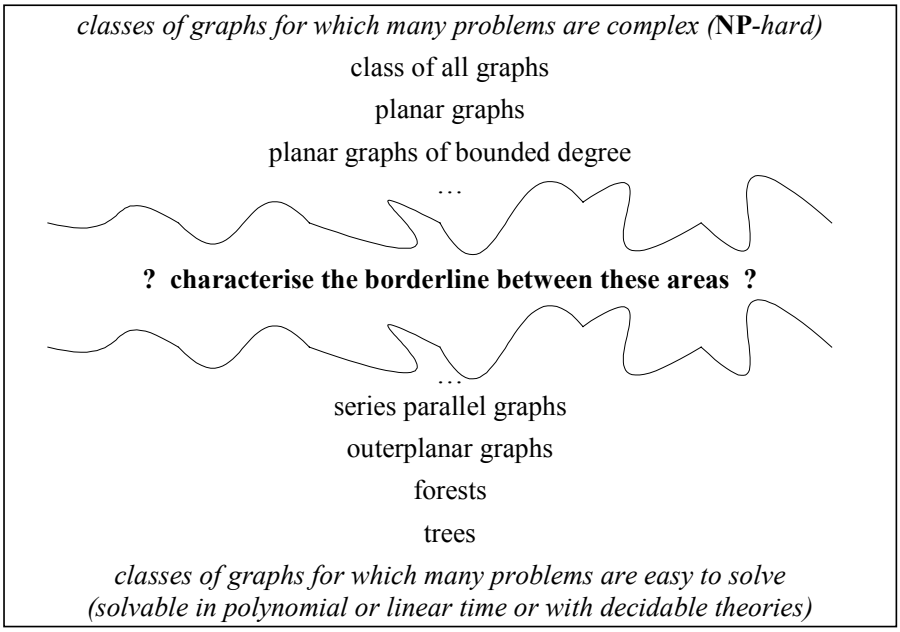

Figure 1: borderline between complex and non-complex graph problems

For trees and graphs with a structure closely related to trees (up to a certain parameter - see e. g. Downey & Fellows (1999)) most algorithmic problems are solvable in polynomial or even linear time. So the natural question is to search for a characterisation of the borderline between these different kinds of behaviour by finding a structural reason for high and for low complexity. Surprisingly, the attempts to achieve this were partially successful (see section 2) and led to the criterion presented in this paper.

To shape the idea of our criterion, it is necessary to look at the proof that a given problem, say P, is NP-hard. The usual way to prove this is to choose a problem Q, which is known to be NP-hard and show that Q can be reduced to the given problem P. The following problem often serves as 'master' reduction problem from the problem class NP.

TILING
instance: $D:=\{t_0,\ldots,t_k\}$ set of square tile types together with two relations $H,V\subseteq DxD$ (the horizontal and vertical compatibility relations, respectively) and a natural number n
question: Is there a nxn tiling, i. e. a function $f:\{1,\ldots,n\}x\{1,\ldots,n\}\rightarrow D$ such that
(a) $f(1,1)=t_0$, and
(b) for all i,j: $(f(i,j),f(i+1,j))\in H$, and $(f(i,j),f(i,j+1))\in V$?

There are many variants and applications of the tiling problem in complexity, decidability, picture recognition and physics, also for other shapes and general covering problems. Figure 2 shows a visual representation of a set of tiles and a tiling for n=3.

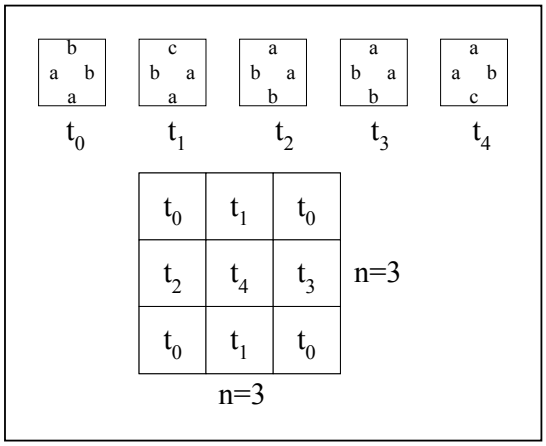

Figure 2: tiling problem example for n=3

We will call a decision problem *undecidable* if there is no algorithm to solve it, otherwise it is *decidable*. For our applications the following result is important.

(1) Theorem (see e. g. Papadimitriou (1994)): TILING is NP-complete if n is given in unary representation, it becomes NEXP-complete if n is given in binary representation, the problem becomes undecidable if it is asked whether there exists an nxn-tiling for all n>0, even when the origin constraint, condition (a), is omitted.

Assume now that we are regarding a decision problem P for a class K of input structures for which we are not able to find a polynomial time solution. In this case it is often conjectured that the problem is NP-hard. If we need a proof for this conjecture the only thing to do is to find a polynomial time reduction of the TILING problem to the original problem, i. e. we have to find an algorithm F which transforms each tiling problem (D,V,H,n) in polynomial time into an input structure $F((D,V,H,n))=G\in K$ such that there exists an nxn-tiling of (D,V,H) if and only if F((D,V,H,n)) has property P. Usually, this is accomplished by showing that some elements $G\in K$ contain (in a definable way) a large grid structure representing the positions of the tiles in the nxn-square, that the local structure of these elements G permits the coding of the tiles and permits a 'flow of information' along the edges of the grid in such a way that it can be verified whether two neighbouring tiles fit together (horizontally or vertically).

The analysis of this proof leads us to the possibilities to reduce the complexity of a problem P on a class K of structures either by trying to avoid the possibility to find or define large grids inside the input structures, or by avoiding the possibility to code the tiles, or by avoiding the flow of information between parts of the structure coding different tiles. This can be achieved by not allowing input structures which contain large grids (in a definable way – specified later in section 3) or by restricting the local structure of the input in such a way that the structures look locally the same (i. e. are very regular or locally isomorphic), or simply by restricting the flow of information. Of course the coding has to be done in such a way that the 'decoding' is reflected in a certain way by the regarded property P.

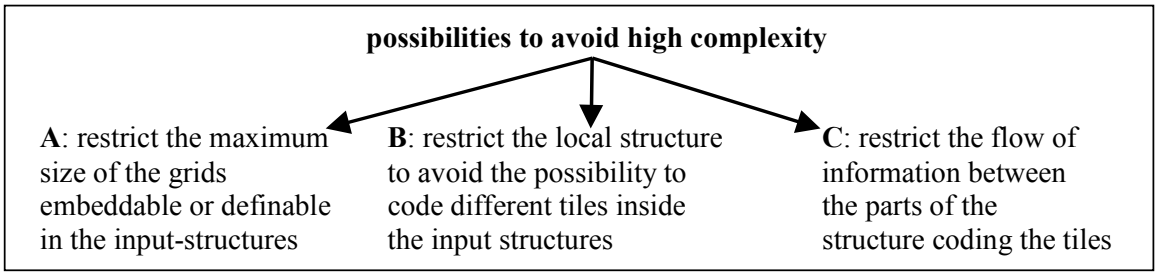

Figure 3: possibilities to avoid high complexity

In the following sections we will make this notions more precise and we will show that the criterion underlying figure 3 is not only of relevance in computational complexity, but it is also important for complexity investigations of dynamic systems and in several other areas of real world applications.

2. SOME IMPORTANT RESULTS FROM TOPOLOGICAL GRAPH THEORY

To give a formal definition of our criterion we use standard mathematical and graph theoretic terminology that can be found in any standard text book. Usually N denotes the set of natural numbers and i, j, k, l, m, n are used for elements of N. The cardinality of a set X, i. e. the number of its elements, is denoted as $|X|$. To make the article also readable for those not familiar with graph theoretic concepts we present some of the necessary terminology here and refer the reader to the details in the literature, e. g. West (1996). A *simple graph* G:=(V, E) with n *vertices* and m *edges* consists of a vertex set V:=V(G)= $\{v_1, \ldots, v_n\}$ and an edge set E:=E(G)=$\{e_1, \ldots, e_m\}$, where each edge is an unordered pair of vertices. We write uv for the edge (u,v) and say that u and v are *adjacent* when uv$\in$E(G). Remember that in this interpretation uv=vu. In this case the vertices u and v are denoted the *endpoints* of the edge e=uv. Define n(G)=:n and m(G)=:m. The *degree* of a vertex a in a graph G is the number of vertices adjacent to it. A *path* is a finite graph with exactly two vertices of degree 1 and without vertices of degree $\geq$3. A graph G is said to be *connected* if for each pair of vertices a and b there is a subgraph H of G which is a path and contains a and b. Usually, the graphs in this paper are assumed to be finite. Exceptions of this rule will be clear from the context or will be explicitly mentioned. Two graphs H and G are said to be *isomorphic* if there is a 1-1-function f from V(H) onto V(G) such that for all pairs of vertices (u,v) of V(H) uv is adjacent in H if and only if f(u)f(v) are adjacent in G. A graph H is said to be a *subgraph* of a graph G, denoted as H$\subseteq$G, if V(H)$\subseteq$V(G) and E(H)$\subseteq$E(G). If e:=uv is an edge of G, then *contracting* e means replacing both endpoints of e by a new single vertex a, and choosing a new graph G.e:=(V(G.e),E(G.e)), where V(G.e):=(V(G)\{u,v})$\cup${a} and E(G.e):=E(G)\({ux: x$\in$V(G)}$\cup${xv: x$\in$V(G)))$\cup${ax: x$\in$V(G)}$\cup${xa: x$\in$V(G)}. A graph H is a *minor* of a graph G if H results from a subgraph of G by contracting some of its edges. We do not distinguish graphs from their isomorphic copies here.

The following structures are essential ingredients for the high complexity of many problems. Let n>0 be an arbitrary natural number. The n-*grid* Q_n is the graph defined by V(Q_n):={(i, j): 0$\leq$i<n and 0$\leq$j<n} and E(Q_n):={{(i,j), (k,l)}: $|i-k|+|j-l|=1$, 0$\leq$i,j,k,l<n}. Define the *infinite grid* Q_∞:=(V(Q_∞), E(Q_∞)) by V(Q_∞):={(i, j): i$\in$N, j$\in$N} and

$E(Q_\infty):=\{\{(i,j),(k,l)\}: |i-k|+|j-l|=1 \text{ and } i,j,k,l\in N\}$. It is easy to see that for each planar graph H (for a definition of planarity see section 2), there is an n such that H is a minor of the n-grid Q_n (see Robertson & Seymour 1984). A surprising observation is that graphs which avoid n-grids as minors (for a fixed n) have a structure which is up to a certain parameter – the tree width – closely related to trees. The result is one of the structural components of our criterion discussed in the previous section. This structural result is the essential reason that a certain set of algorithmic problems can be solved efficiently.

To make the ideas more precise we need the notion of tree decomposition. A *forest* is a graph which does not contain a cycle. A *tree* is a connected forest. A *tree decomposition* of a graph G is a pair (T,X), where T is a tree and $X:= (X_t: t\in V(T))$ is a family of subsets of V(G) with the following properties:
$(X_t: t\in V(T))=V(G)$, for every edge e of G there exists $t\in V(T)$ such that e has both ends in X_t, and for $t,t',t''\in V(T)$: if t' is on a path of T between t and t'', then $X_t\cap X_{t''}\subseteq X_{t'}$.

The *width* w((T,X)) of the tree decomposition (T,X) is $\max(|X_t|-1: t\in V(T))$. The graph G has *tree width* tw(G):=m if m is the minimum k such that G has a tree-decomposition of width k.

The following landmark result concerning grids, minors and tree width shows that graphs without large grid minors are essentially 'tree structured'.

(2) Theorem (see Robertson & Seymour (1996)): For every planar graph H there is a natural number n_H such that every (even possibly infinite) graph G without H as minor has tree width $\leq n_H$.

A class K of graphs has *universally bounded tree width* if there is a natural number m such that the tree width of each graph in K is bounded by m, i. e. $tw(G)\leq m$ for each $G\in K$. There is a world of results proving that many algorithmic problems for many classes of graphs of universally bounded tree width can be solved in a time being polynomial or even linear in their input size. Moreover, there are many uniform approaches defining an algebraic, a logic or a hybrid calculus which allow us to describe algorithmic problems and to construct efficient algorithmic solutions for such problems, sometimes even in an automatic way (see e. g. Arnborg et al. (1991), Courcelle et al. (2000)).

Together these results give a good theoretical explanation why so many problems for input structures without large grids can be solved efficiently:

- Graphs without large grids have a close similarity to trees.
- For such graphs, many global problems can be solved by algorithms working locally in a way similar to dynamic programming, hence they can be solved in linear time.

As a consequence, we can conclude that almost all algorithmic problems for graphs without large grids (i. e. of bounded tree width) can be solved in linear time. These problems contain almost all NP-hard problems for graphs and networks.

Now assume that grids of arbitrary size can be contained in the class of input structures. Is it possible to give a good characterisation of the structures and problems for which we can

expect efficient algorithms in this case? The results in this area are not as elaborate as in the preceding considerations. Hence, we will show an example from classic efficient algorithms underlining our arguments about the connections between structure and complexity.

A standard problem of computer science is the problem of sorting a set of n input elements corresponding to a given key. Such a general sorting problem can be solved in O(n log n) time. The sorting problem asks for the existence of a linear ordering which sorts the elements according to the same order defined by the given ordering of their keys. In this setting the problem is not obviously a question about graphs of bounded tree width since the graph underlying a linear ordering is a complete[3] graph (simply transform each directed edge of the ordering into an undirected edge). A complete graph of n vertices has the tree width n-1 and contains the $\lfloor\sqrt{n}\rfloor$ x $\lfloor\sqrt{n}\rfloor$ grid as subgraph.

Looking more carefully at the structures underlying the ordering problem, it is observable that the input structures can be represented as ordered trees, i. e. trees whose branches are linear orderings[4], and the output structures are closely related to graphs of path width[5] 2, i. e. they result from a graph of path width 2 by substituting two of its paths by a linear ordering. But in an ordered tree the successor relation, whose transitive closure is the ordering of the tree, can be used to define the partial ordering of the ordered tree. This can be used as basis of a transformation of the ordering problem into a tree problem. It can be shown in a related way that many algorithmic problems with known efficient solutions, e. g. breadth-first search, depth-first search, planarity, minimum spanning tree, topological sort, max-flow, can be solved by algorithms using a flow of information whose structure is closely related to trees or paths (see Seese & Schlottmann (2001) for more details).

3. STRUCTURE OF DYNAMIC SYSTEMS AND COMPLEXITY

Dynamic systems are very important for many real world applications, particularly in market-driven economic dynamics and many everyday business life contexts. During the last 10–15 years there has been a lot of success in the study of dynamic systems. The literature on these subjects is very large and growing, and it is impossible for us to give an exhaustive presentation. In this section, we will have a look at dynamic systems with respect to the question how structure influences complexity.

On a first look it seems that dynamic systems are typically much easier than complex decision problems, since for dynamic systems we are usually interested in the iteration of a 'simple' function f, i. e. we start from a value x_0 and define $x_{t+1}:=f(x_t)$ for all t=0,1,2,…, so we are interested in the trajectory generated by the iteration of the function f starting from a given value. Possibly, f can have additional parameters. Typical problems in this area are:

---

[3] A graph is *complete* if each pair of its vertices is connected by an edge. A complete graph with n vertices is denoted (up to isomorphism) as K_n.

[4] In this special case only one branch is used – the given linear ordering.

[5] *Path width* is defined in the same way as tree width, the only difference is that the underlying graph of the tree decomposition has to be a path.

STATE CAN BE DRIVEN TO THE ORIGIN, where a system $x_{t+1}:= f(x_t,u_t)$ and a fixed initial state x_0 are given and the question is whether there exist some $T\geq 1$ and controls u_t, $t=0,\ldots,T-1$, such that $x_T=0$,
NULLCONTROLLABILITY, asking whether all states x_0 can be driven to the origin,
TRAJECTORY GOES TO THE ORIGIN, asking for a system $x_{t+1}:=f(x_t,u_t)$ and a fixed initial state x_0 whether there exist some $T\geq 1$ such that $x_T=0$, and
ALL TRAJECTORIES GO TO THE ORIGIN, asking whether for every x_0 there exists a T such that $x_T=0$.

For linear systems all these problems can be solved in polynomial time (see Sontag (1995)). For arbitrary systems this question is stated too general and usually too difficult to answer, e. g. the null-controllability problem for general non-linear systems includes the problem of deciding whether a given arbitrary non-linear equation $\Phi(u) = 0$ has a solution (see Blondel & Tsitsiklis (2000) for more details). Even if the system inhibits only a single scalar non-linearity the problem to decide about stability is difficult, as illustrated by the example of Neural Networks whose activation function is a saturated linear function. For this case, Siegelmann and Sontag (1995) showed that TRAJECTORY GOES TO THE ORIGIN is undecidable by simulating Turing machines. This is relevant to many finance applications since Neural Networks are a common method for prediction of economic time series, e. g. asset prices over time.

What is it that makes such systems complex in case of non-linear functions and why they remain easy for linear functions? Is there a connection to our criterion on grids? These problems are of arithmetic kind in a certain sense and graphs do not appear on a first look. Of course, these problems are proved to be NP-hard or undecidable via the usual proof techniques, hence tilings and Turing Machines are reduced to this problems, and so grids occur on a second view. But the grids can also be found by a more direct consideration. To demonstrate this, let us focus now on a slightly different, but nevertheless famous problem, the decision problem for the Mandelbrot set, firstly asked by R. Penrose. It is defined by the recursion $x=x^2+c$ over the complex numbers. The loop of the recursion terminates with output 1 if $|x|>2$. Blum & Smale (1993) used a special computation model over real numbers from Blum et al. (1989) to show that this set is undecidable.

Another example of an iteration leading to undecidable sets is the Newton's method used to solve mathematical problems in many applications. It is a method to search for approximate zeros ζ of functions f, i. e. values ζ with $f(\zeta)=0$. The corresponding decision problem is whether the Newton method converges for a given starting point. Using the BSS-model of computation over R, the following theorem was proved:

(3) Theorem (see e. g. Blum et al. (1998)): The Mandelbrot set and the set of points that converge under Newton's method are undecidable over R.

Algorithmically, the decision problems from (3) are quite simple, the algorithms are just realisations of one loop. So what is so difficult about computing one loop containing only very simple arithmetic operations? In case of the Mandelbrot set it is just one multiplication and one addition of complex numbers. By analysing a single iteration step we can see that if x and c have the form $x=a+bi$ and $c=d+ei$, then the result of the next step of the iteration is of the form $x=x^2+c=(2ab+e)i+a^2-b^2+d$, where a, b, d and e are real

numbers and i is the complex unit. It is obvious that the arithmetic properties of this operation are quite simple. But where is the hidden complexity?

The first reason is the size of the objects handled in this algorithm – the numbers. Let us assume that we work with real numbers having finite binary representations (see below) as inputs and start with an input c of size n. More exactly, we could assume that the length of the binary representation of both components d and e have a length of at most n. After the first step, the length of the representation of the result has already doubled (exactly it is at least 2n –1; the size of the represented number grows nearly to $2^{(2^n)}$ after n steps). If we use an arithmetic of an arbitrary high precision (i. e. we allow registers of arbitrary large finite size) the representation of the intermediate result has at least a size of $2^t(n–1)+1$ after t steps of running the loop. Hence, even writing down the intermediate results is difficult since they are growing exponentially large. However, it is not necessary to know the complete intermediate result for the final decision, since it is sufficient to check whether the equivalent condition $a^2+b^2>4$ is true for the actual value of x=a+bi. To see more details, let us represent the real part and the imaginary part of a complex number by a sequence of bits, respectively, where we separate the bits representing the integer parts from the bits representing the fractional parts by a dot, e. g. the real part by $(u_s\ u_{s-1}...u_1 \bullet u_{-1}\ u_{-2}...u_{-(r-1)}\ u_{-r}...)$ and the imaginary by $(v_t\ v_{t-1}...v_1 \bullet v_{-1}\ v_{-2}...v_{-(r-1)}\ v_{-r}...)$. Moreover, let us assume that we compute using finite complex numbers only, i. e. numbers whose representation of the real as well as of the imaginary part can assumed to be finite. To analyse the complexity from a structural point of view, assume that we are interested in the subproblem to decide about the outcome of the underlying recursion after T steps for an arbitrarily given natural number T and an arbitrary finite input x. To decide about the result, it is sufficient to know whether there exists an integer t with $1\le t\le T$ for which there are more than two bits set to 1 in the representation on the left side of the point in the binary representation of a^2+b^2, where x=a+bi is the result of the computation at step t.

Now define a *communication graph* for an algorithm. The idea is just to look exactly what possibly happens with the content of the variables on a bit level at each time step. So a communication graph consists of vertices that represent all possible contents of the variables on a bit level at each step of time, and there are edges between such 'bit positions' if they possibly influence each other (see Seese & Schlottmann (2001) for details and Hromkovic (1997) for related ideas). We are particularly interested in the size of embeddable grids, i. e. the tree width, and in regularities of the communication graphs.

A simple observation is that the Mandelbrot set problem becomes trivial if the size of the input and the size of the intermediate results are bounded by a fixed n. In this case, the tree width of the corresponding communication graph is bounded and the problem becomes obviously decidable. Particularly, the decidability is trivial since for each fixed bound of the number of bits in the variables the algorithm degenerates to a finite automaton.

(4) Lemma: The tree width of the communication graph of the above algorithm for the Mandelbrot set cannot be bounded.

This is caused by the growing number of necessary bits for the representation of the intermediate results. It is known that there is no uniform bound for the length of the

recursion for arbitrary inputs. The rest follows from an analysis of the communication necessary to multiply numbers having large representations. It is not difficult to see that the communication graph defined above contains a bipartite graph $K_{m,m}$ whose size m is growing with the size of the calculated numbers. Since it is a well known fact that the tree width of such graphs is growing with m, the tree width is not bounded.

Hence, we have an indication for well known dynamic systems that complex problems are connected with large grid sizes. With respect to our philosophy, it is useful to observe that the communication graphs are quite regular. However, the irregularity that seems to be necessary in a certain sense for high complexity comes from the different inputs here which cause an irregular flow of information along the regular communication graph. This again gives evidence of a possible connection between high complexity of dynamical systems on one side and large grids, irregularities and an unbounded information flow on the other. Nevertheless, this is not a strict proof that high complexity for dynamic systems and large grids are causally related, it is only an indication that there could be a connection. Related questions can be asked for other dynamic systems, which are of special interest for economic and business applications (see e. g. Sterman (2000)). The complexity of many real world systems and problems is mainly caused by an interaction of structural complexity and the dynamics of the systems and processes. The above mentioned Mandelbrot set is an example for an algorithmic problem based on a simple loop but the prediction of its behaviour is nevertheless very complex. Therefore, it is obviously more complex to try to predict the behaviour of more complicated systems in many real world application contexts, e. g. in prediction of financial markets or other economic applications. We will provide some consequences and solutions for further applications in the next section.

4. SELECTED APPLICATIONS IN ECONOMICS AND FINANCE

4.1. Auctions and capital markets

Contrary to fixed pricing, where a market participant endogenously sets a price for a product or service that has to be accepted or refused by the responding market participant, in dynamic pricing the price is dynamically determined by bids of the market participants. We concentrate on multi item auctions where interdependent auctioned items are regarded and where bidders have got preferences over combinations of items. There are several applications, e. g. allocation of bandwidth, transportation or manufacturing tasks, as well as contracts between construction companies or electricity markets.

A multi item auction can be modelled as a combinatorial auction, i. e. we start with a universe O of traded objects and the buyers supply the auctioneer with a set A of bids. The i-th bid is a subset A_i of O and a price p_i that the buyer i is willing to pay for all the objects in A_i. Our auctioneer has to choose a collection of bids $B \subseteq A$ that yields the best possible total price while being consistent, in the sense that no two sets A_i and A_j from A overlap. The auctioneer's decision which of the conflicting bids to accept is related to graph theoretic concepts. A *bid graph* G consists of a set of vertices of G, which are the bids and a set of edges, where an edge is placed between any two bids if they share an object. Each vertex gets a weight equal to the value of the bid it represents. In this

notation the auctioneer's goal is to find the most valuable consistent set of bids. But this problem is equivalent to the maximum weight independent set problem, which is NP-hard and cannot be approximated to within a ratio $O(n^{1-\varepsilon})$ for an n-node graph and any positive ε, unless P=NP (see Akcoglu et al. (2001)). The problem remains NP-complete for cubic planar graphs and becomes solvable in polynomial time for many classes of structures of bounded tree width (see Garey & Johnson (1979)). Hence, we have again a problem caused by large grids and an example for a solution by restriction of the grid size.

Moreover, Aspnes et. al. (2000) showed that in a double auction capital market, i. e. a market where stock buyers and stock sellers compete in a two-sided auction, the possibility of predicting future prices depends heavily on the computational complexity of market participants' trading strategies. If there are a large number of traders but they employ a relatively small number of simple strategies, then there is an algorithm for predicting future price movements with high accuracy in polynomial time. However, if the number of trading strategies is large, market prediction becomes computationally more complex than NP-complete problems (Aspnes et al. (2000) define a corresponding strong complexity class). Grids can be found here because of the common reduction techniques used in the proofs of these facts.

4.2 Risk management

The management of risk from financial transactions, e. g. lending money or holding positions in the financial markets, and the management of aggregated risk in whole institutions, e. g. banks, or in world-wide financial systems has become more important than ever during the last years. This is due to the large extent of financial disasters that has been observable (Long-Term Capital Management hedge fund crash, Barings Brothers bank debacle, etc.). Since the world is getting more and more dependent from its fragile financial system, the management of aggregated risk will be one of the big challenges for the financial industry (and e. g. for politicians and the financial supervision authorities) in the future. It must be pointed out here that most aspects of risk management are also relevant to other business industries like large manufacturing companies.

At first, it is important to distinguish between uncertainty and risk concerning decision situations e. g. for a manager. Basically, the latter situations are characterised by a non-empty set S, $|S| > 1$ of possible outcomes for a variable X (e. g. describing a certain state of a system or an amount of loss) at a fixed future time horizon t and probabilities $Prob(X = s) \in [0,1]$ for the occurrence of each $s \in S$ at time t. In contrast to this, if we face a decision under uncertainty over X we are not provided with possible outcomes and/or the probabilities of the outcomes are not known. Of course, such uncertain situations and events are usually not manageable unless we get more knowledge about them, whereas decisions under risk can be managed to some extent (i. e. they can not be managed perfectly by 100 percent confidence).

The probabilities $Prob(X = s)$ can either be objective (true probabilities) or subjective (a priori probabilities estimated by the risk manager). In practical applications, the true probabilities are not known, so they have to be estimated from observations. This often leads to some problems due to the lack of data and/or the quality of available data sources.

Typically, when we talk about risk management we are mainly interested in negative consequences from outcomes of the variable X, e. g. if X represents a yield which we expect to obtain from an investment we are only interested in situations where X falls below a fixed target value s': $X < s'$ or equivalently, $X - s' < 0$.

Now we will create an illustrative example to show how risk management decisions in an economy are related to our criterion. Consider a bank manager who knows that the profit of his bank at the end of the next year depends completely on the same year's profits of different companies, e. g. they are obligors of the bank and the bank has no other sources of profits. The companies operate in a certain common economic environment. The bank manager wants to be sure that the bank's profit will not fall below a certain minimum level since otherwise he will get no payment from the bank's owners or the bank will even go bankrupt if the bank's losses (loss = negative profit) exceed a critical level. How will he try to manage this situation?

Figure 4 shows the main focus of current risk management methodology from our view of complexity in the context of the above example.

At first, the bank's manager will try to transform the uncertain decision situation described above into a risky one. This means he will build (or select) a model of the whole system containing the companies from which the bank's profit depend. Using current risk management methodology, he will e. g. choose a random variable X_i having a certain probability distribution as a representative for each relevant company i in his model and use correlations between the variables to express the dependence between the different companies. Then he will estimate the parameters for his model from his own observations of the behaviour of the real-world system. Finally, he will use the model to estimate the distribution of the bank's profits and derive his own management decisions to influence the real-world system towards his own goals.

It is important to keep in mind that each random variable used for modelling a risk driver (e. g. a company) in this decision process is only a simple representative for a complex dynamic system that is usually too difficult to analyse from the risk manager's view, e. g. because of bounded resources, time restrictions, asymmetric information, principal-agent problems etc.

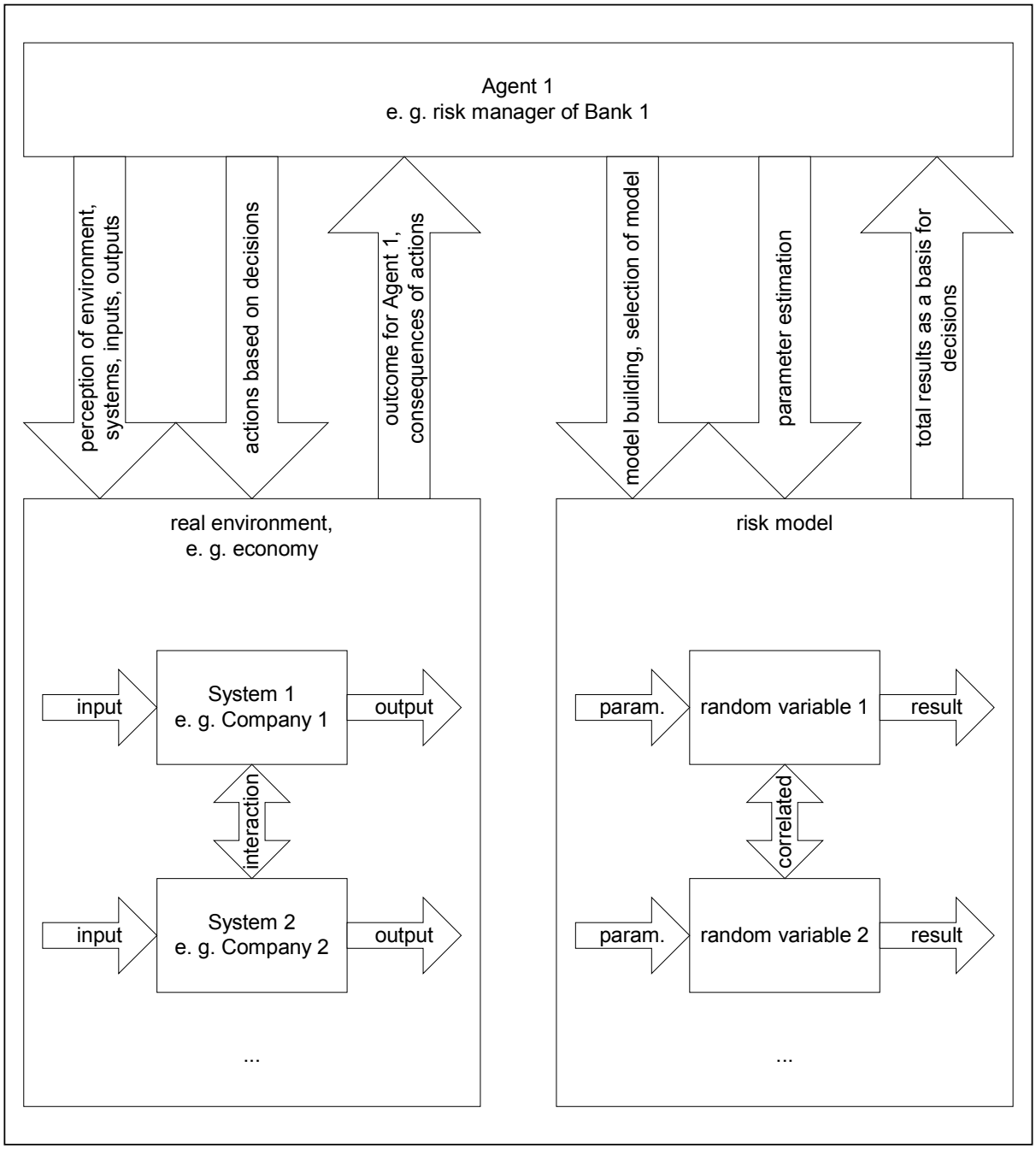

Figure 4: current methods of risk management

Since there are usually other banks and institutions in an economy that have also lent money to the regarded companies they will also follow the approach from figure 4 but use their own models, parameters and observations. This leads to an interesting structure containing large grids which connect the elements of the regarded economy – obligors, lenders and other important institutions that have a major influence on the overall stability of the system.

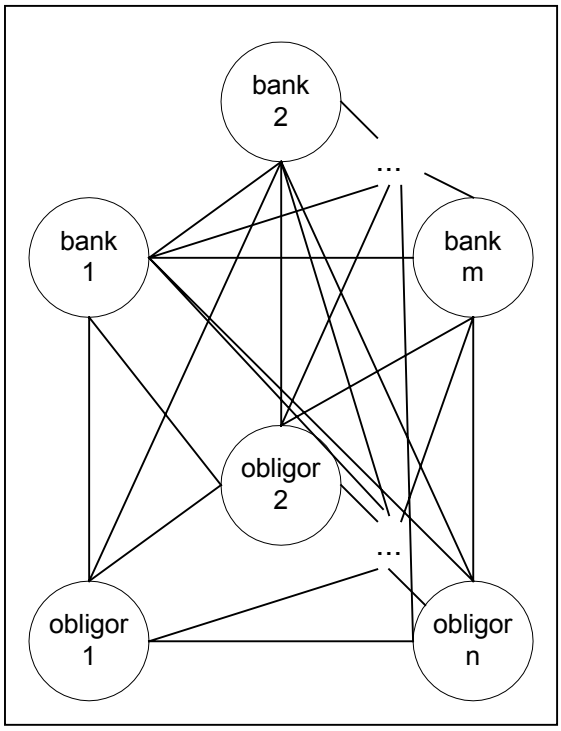

Figure 5: large grids in risk management situations

Furthermore, because of the interaction between the elements of the economy as well as time differences between perception of changes, reactions and consequences of these reactions this is a highly dynamic system.

So if we are concerned about the stability of the financial system on a large scale, our theoretical complexity considerations are interesting for the analysis of structural reasons for aggregated risks.

The main problem e. g. for the financial supervision authorities which try to establish rules to stabilise the financial systems of large economies is that huge grids cannot be avoided in a liberal economy where nearly all agents like banks etc. are allowed to make contracts with each other. As a consequence, the financial supervision authorities have to concentrate on our paths B and C from figure 3: Either the local structure inside the large grids has to be very regular or the level of interaction between the elements inside the grids has to be restricted. Indeed, the financial supervision authorities move further into this direction as the development of supervisory regulations for financial companies have shown during the last ten years (cf. e. g. Basel Committee of Banking Supervision (2001) and the subsequent publications of the Bank of International Settlements).

We think that the discussion of our complexity considerations in this very recent field of research particularly provides an interesting insight into the structural reasons for the necessity of supervisory and internal risk management rules that matches the usual arguments based on a pure probabilistic and correlation-oriented point of view. It must be pointed out here that our complexity considerations can be used as an additional tool together with the usual risk management techniques based on probabilities and risk models to improve the understanding of risky decision situations by an explicit structural analysis of the underlying sources of risk. In addition to that, the analysis of the

computational complexity in risk management decision situations is highly relevant for the development of adequate tools. To underline these arguments, we will focus on two special areas of risk management in the next sections.

4.3. Credit risk

Currently, one main focus in the management of credit risk in financial institutions is on the measurement and management of aggregated losses from lending money to a given set of n obligors (credit portfolio). The focus on aggregated losses comes from the fact that small losses are not threatening the bank's existence (although they have a bad influence on the bank's profits) whereas large losses can result in an insolvency of the bank itself.

For the calculation of the probability distribution of aggregated losses from the portfolio, each obligor $i \in \{1,\ldots,n\}$ can modelled by a two-state variable X_i (so-called Bernoulli variable) that takes the value 1 if obligor i defaults at a fixed time horizon t, e. g. t = 1 year (which means the bank loses the outstanding amount L_i of money from i) and the value 0 if no default of obligor i occurs at the time horizon. Alternatively, one can also use an intensity-based approach (see e. g. Lando (1998), Duffie & Singleton (1999)) where X_i takes the value 1 if obligor i defaults within a fixed time horizon t. The probability of default p_i describes the probability that $X_i = 1$. Of course, Prob $(X_i = 0) = 1 - p_i$. The potential loss amount for the bank L_i from a default of obligor i is called Loss Given Default (LGD). The LGD is the part of the total amount M_i lent to obligor i which is not covered by collateral agreements etc.

We now consider the bank from our example in section 4.2. which is holding a credit portfolio of n obligors and has no further risky investments. For the following considerations we assume that all L_i are positive integer values. This is not critical from a real-world view, as we can transform a set of given real numbers into integer values by scaling the former using an appropriate factor (e. g. multiply amounts in dollars/cents by 100 to obtain integer dollar values). To avoid trivial solutions, we assume $L_i \neq L_j$ for $i \neq j$ and $p_i > 0$ for all $i \in \{1,\ldots,n\}$.

If we assume a given integer value K expressing the bank's ability to absorb the accumulated losses $X = \sum_i X_i L_i$ arising from defaults of obligors in the portfolio at the time horizon t, it is interesting to calculate the probability that the bank survives the time horizon t, denoted by Prob $(X \leq K)$. K is the default threshold of the bank, e. g. K = amount of bank's equity (cf. the main idea of the Black-Scholes-Merton firm value model Black & Scholes (1973), Merton (1973)). This is a crucial question, particularly for the bank's management, the bank's employees and the banking supervision authorities. The following lemma shows that this problem is computationally intractable.

(5) Lemma: The problem of calculating the bank's survival probability given the data from its n obligors p_i, L_i for all $i \in \{1,\ldots,n\}$ and the bank's default threshold K is NP-hard.

Proof: see appendix.

(6) Corollary: The problem of calculating the bank's default probability Prob $(X > K)$ given its n obligors' data p_i, L_i and the bank's default threshold K is NP-hard.

Proof: Prob (X > K) = 1 - Prob (X ≤ K). The rest follows from Lemma (5).

These two lemmata are interesting results, as we do not need any assumptions on the default probabilities or the dependence structure (e. g. correlations) between the Bernoulli variables to obtain a computationally difficult problem. Furthermore, we did not assume any specific portfolio credit risk model. Therefore the results are model-independent.

We will now analyse a general problem of portfolio credit risk modelling using our complexity criterion to provide an insight into the fine structure of these problems. To achieve this, we will construct an input structure containing a general description of a credit portfolio.

(7) Definition: A credit portfolio can be represented by an undirected graph G = (V,E) with vertex and edge evaluations where the vertex set V = {1,…,n} represents all obligors, and the edge set E is complete, i. e. contains all possible edges between each pair (i,j) of obligors i, j. There are functions p:V->R and L:V->N that map the associated default probability p_i and the loss given default L_i to each vertex (obligor). Furthermore, there is a function r:E->R mapping the strength of the dependency between the adjacent two vertices to each edge.

An example for a small portfolio containing four obligors is shown in figure 6.

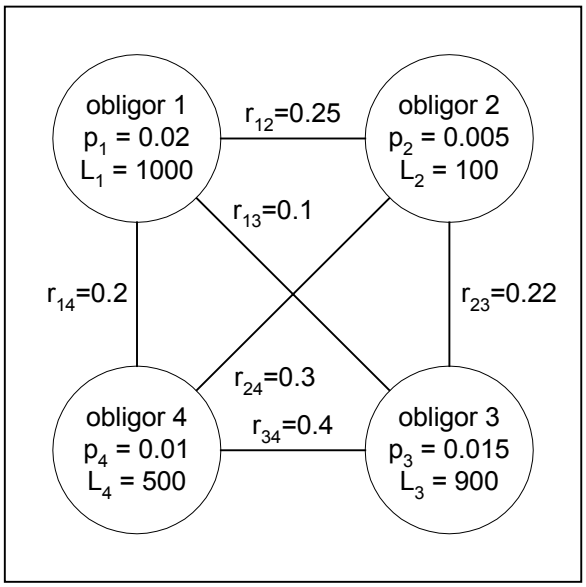

Figure 6: credit portfolio modelled by graph elements

If we are interested in portfolio credit risk measurement or management based on quantitative methods using all the given default probabilities, dependencies between obligors and loss amounts, we have to construct algorithms based on the graph G from definition (7). This is particularly true for all measurement (and management) methods that base on the full probability distribution of aggregated losses (cf. e. g. Lemma (5) above).

(8) Proposition: According to our complexity criterion, any problem of portfolio credit risk measurement that requires the computation of the full probability distribution of

aggregated losses from input data which is based on the graph structure specified by definition (7) is a complex problem.

Some observations supporting this proposition:

The tree width of the input graph G for the portfolio credit risk management algorithm A is n-1 since G is a complete graph. So we observe a large grid structure as the first ingredient of complexity since the tree width of G cannot be bounded for $n \to \infty$.

The second observation is that we usually have local irregularities concerning the vertices of the graph. Each vertex $i \in V$ can have a distinct default probability $p(i) = p_i$ and an arbitrary (but natural number) potential loss amount $L(i) = L_i$. This is quite realistic since banks lend different amounts of money to many obligors of different quality (concerning their default probability). Of course, in real-world applications the cardinality of the regarded set of different default probabilities used in calculations based on graph G is typically small (e. g. ≤ 18) due to the fact that the probabilities have to be estimated from empirical observation of defaults in a small number of (so-called rating) classes containing obligors of similar quality. But this is not limiting our observation of local irregularities inside our graph as long as there are different LGD values L_i and at least two different default probabilities p_i. This assumption should compatible with nearly all real-world credit portfolios.

Looking at the flow of information along the edges of the graph G we observe that the flow of information between two obligors representing their dependency causes computational difficulties unless r is a constant function satisfying $r(e) = 0$ for all edges $e \in E$. In this case, we can remove all edges from our input graph and obtain a less difficult problem, since we do not only restrict the flow of information between different obligors but we also reduce the tree width of the input graph to 0 since there are no grids left. Thus, there are only independent obligors. Many portfolio credit risk problems are easier to handle in this setting.

To illustrate this fact, imagine we construct an algorithm C which simulates the state of all obligors in the portfolio at the end of the next year using the given input graph G by assigning the value 0 to each vertex if the corresponding obligor defaults in a simulation run, and the value 1 otherwise. In the trivial case $r(e) = 0$ for all edges e the algorithm C can handle the decision of assigning the value 0 or 1 to a fixed vertex $i \in V$ just by using the associated default probability so there is no additional computational cost beyond simulating each obligor in each simulation run. But if there is at least one $e \in E$ having $r(e) > 0$ then there will be an additional computational requirement to handle this dependency. And in the case of a large grid structure containing such non-trivial edge valuations in conjunction with non-trivial vertices (i. e. non-constant default probabilities and non-constant LGD values) we expect computationally difficult problems.

Note that we explicitly do not restrict the dependence structure between two obligors to be measured by correlations as proposed in most current real-world applications. The dependence structure can also be modelled by copula functions or other concepts. For our complexity criterion, it is sufficient that r has a non-trivial structure, i. e. r is not constantly zero.

Summarising all observations about the graph used for modelling the credit portfolio problem structure, we have found that all three elements of complexity from our criterion are present inside the graph G. Therefore, we cannot expect to find an efficient (i. e. polynomial time) algorithm A for portfolio credit risk measurement which produces exact results based on the full probability distribution of aggregated losses which are calculated using given the input structure defined by the above graph G unless the problem class P = problem class NP.

We want to point out that this is not a rigorous mathematical proof for (8). But it provides a valuable insight into the fine structure of all portfolio credit risk problems concerning measurement and management. The existing literature usually focuses on data problems and specific model assumptions, e. g. for the obligor's quality from the bank's perspective, default probabilities or dependencies, to justify a certain calculation procedure or algorithm. Many studies focus on economic or mathematical modelling of defaults, but not on computational aspects which are mission-critical for effective and accurate real-world portfolio credit risk management. We have shown above that even without assuming a specific model for obligors' dependencies, obligor data etc. we cannot expect to find a polynomial time (in the number of the obligors) algorithm for measuring portfolio credit risk unless the problem class P = problem class NP. As a consequence, portfolio credit risk management based on quantitative portfolio credit risk measurement is a complex task.

Our proposition (8) is also supported by the fact that until now there is no known polynomial time algorithm for quantitative portfolio credit risk measurement based on the full probability distribution of losses calculated using the data modelled by our graph. All existing methods use approximation schemes or randomised algorithms to avoid the complexity. We will now apply the complexity management guidelines derived from our criterion to see how we can reduce the complexity of the problem by restricting the inputs. By considering some details from selected portfolio credit risk measurement models we will see that these guidelines are implicitly used when applying these models.

The first method derived from our criterion for complexity reduction of portfolio credit risk problems is to break up the large dependence structure between defaults of different obligors i,j. If we consider pairwise independent obligors, many problems will become easier to solve since we do not allow a large grid structure in our input graph G for a certain (hypothetical) portfolio credit risk measurement algorithm A.

Of course, for most real-world portfolios we cannot assume the obligors to be independent from each other. But the idea of breaking up the dependence structure can even be valuable in settings where the obligors are dependent. For example, if we assume the default probabilities p_i to be conditionally independent given a certain state of the obligors' common economy which is reflected e. g. by the state of a fixed number of macro-economic indicators influencing all default probabilities the same way, we obtain computationally (and statistically) easier calculations. To illustrate this fact, consider e. g. that all default probabilities p_i are linked to a single macro-economic factor according to the equation $p_i = m_i * Z$ where m_i is the mean default probability of obligor i and Z is a random variable having a mean (or expectation) equal to 1 that scales all default probabilities into the same direction depending on the current state of the economy. Z describes a common macro-economic driver of default probabilities because they rise in a

recession and drop in a boom of the modelled economy in which the obligors operate. If we assume this kind of obligor dependency modelling we will at least obtain computationally easier calculations given a fixed state of the economy since the input graph G for the calculation algorithm A does not explicitly contain large grids. Instead, the respective algorithm A has to be called many times using different parameters (i. e. p_i) which are determined by the states of the economic indicators to incorporate the default dependencies between obligors. Afterwards, the output from each run of the regarded algorithm A has to be processed to obtain a final result that includes the necessary dependencies. In current real-world models for portfolio credit risk measurement this is a common method of complexity reduction. For the CreditRisk+ model (cf .CreditSuisse Financial Products (1997)) this method is the key aspect of modelling default dependencies. Due to this aspect (and further concepts that reduce the complexity, see below), the analytical approximation of the loss probability distribution can be performed very efficiently. In Wilson's model (cf. e. g. Wilson (1997a, 1997b)) the state of the economy is simulated by a Monte-Carlo-Algorithm in "outer simulations". The default probabilities and other probabilities describing the development of the obligors' credit quality (so-called migration probabilities) are adapted according to the results of these outer simulation runs, and afterwards each single obligor can be simulated by a second Monte-Carlo-Algorithm in an "inner simulation" using the adapted (conditional) probabilities. There are no explicit dependencies to be handled in each inner simulation run since the dependencies between defaults of different obligors are covered by the adaptation of the obligors' individual probabilities according to the outer simulations.

If we look at the dependence structure modelled by an input graph G' after introducing a common risk factor which is used as a linking element between obligors, we observe that the tree width of the resulting graph G' is significantly lower than the tree width of our original graph G. Figure 7 shows an example.

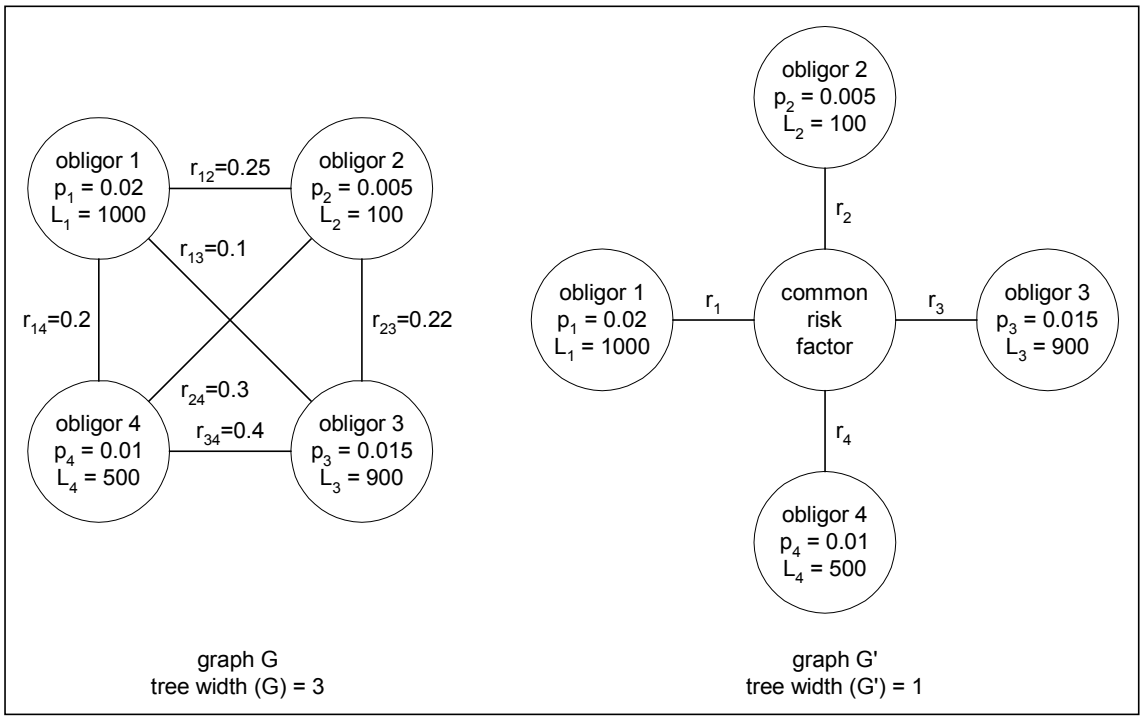

Figure 7: complexity reduction by introduction of a common risk factor

It is important to see that the tree width of the resulting graph G' is bounded by 1 whereas the tree width n-1 of the original graph G is not bounded for $n \to \infty$.

Of course, the underlying risk measurement algorithm A has to be changed according to the new input structure G', but we can at least expect lower computational complexity due to the reduced size of the embedded grids inside the input structure.

This observation is not only relevant for portfolio credit risk measurement but also for many other methods of asset valuation or risk management, consider e. g. the Capital Asset Pricing Model (CAPM) as a one-factor valuation approach (see e. g. Sharpe (1963, 1964) for details).

The second way of complexity reduction proposed by our criterion is to enforce a homogeneous structure inside the given graph G. In the context of portfolio credit risk measurement this is e. g. the case if all obligors share common properties, i. e. if there is a constant LGD value $L = L_i = \text{const}$ and/or a fixed default probability $p = p_i = \text{const}$ for each obligor i. We expect less computational complexity in this case since we do not have to respect certain individual properties of the obligors when processing the graph G by an algorithm. An example for such a complexity reduction principle used in a portfolio credit risk model can be found in CreditRisk+. The obligors are clustered into homogenous classes according to their LGD values. The classes are generated by choosing an appropriate norm exposure L (the CreditRisk+ model proposes $L = \max_i \{L_i\} / 100$). Afterwards, dividing each individual LGD value L_i by the norm exposure L and rounding up the resulting value to the nearest next integer number j yields the corresponding class number j for each obligor. The resulting classes contain obligors having quite similar or even equal LGD values. The advantage of this classification procedure is that the Poisson approximation used to calculate the loss probability distribution in the CreditRisk+ algorithm can be performed on a significantly smaller number of homogenous classes instead on each obligor. Together with a conditional independency between obligors (cf. our above remarks) this yields a very fast approximation scheme.

Hopefully, these insights into the complexity reasons and complexity reduction principles can support the invention of "second generation" portfolio credit risk measurement models yielding a better computational performance than current simulation models and avoid other problems of analytical approaches, e. g. the instability of the CreditRisk+ recursion in some parameter settings when working with fixed arithmetic precision (i. e. usual floating point data types on a standard computer).

Now we will return to our bank from our illustrative example from section 4.2. Assume that the bank uses a portfolio credit risk model for the calculation or approximation of the full portfolio loss distribution. A common downside risk measure used for quantification of portfolio credit risk given a portfolio's aggregated loss distribution is the Credit-Value-at-Risk.

(9) Definition: Given a probability level $\alpha \in (0,1)$ and a cumulative discrete probability distribution F describing the cumulative probability of aggregated losses in the credit portfolio the Credit-Value-at-Risk at the level α is given by $F^{-1}(\alpha) - \mu(F)$, where $F^{-1}(\alpha)$ is the α-percentile of F and $\mu(F)$ is the mean/expectation of F.

Now if a portfolio manager in our sample bank wants to optimise the bank's risk-return profile using the Credit-Value-at-Risk and the expected return from the portfolio, he will build an optimisation problem of the following type:

(10) Problem:

$\max \Sigma_i r_i y_i$

with respect to $\Sigma_i w_i y_i \leq K$

$y_i \in [0, L_i]$ rational numbers

$r_i, w_i > 0$ rational numbers

The r_i values characterise the rate of return for obligor i from the bank's perspective and the w_i values are the "capital weights" that have to be respected for each unit of net risk (LGD) held by the bank in its portfolio. Summing up all capital weights multiplied by the amount of risky LGD we obtain the aggregated amount of risk which can be compared with a given bound K, consider e. g. the bank's given equity K (cf. our derivation of the problem stated in lemma (5)). Note that this is the way banking supervision authorities restrict the lending capacities of banks by imposing capital weights w_i on credit granting activities and setting an individual supervisory capital limit K for each bank. Of course, a bank can use internal risk weights based on Credit-Value-at-Risk calculations for its own risk-return management purposes. Usually, internal risk management figures are assumed to describe the bank's economic situation much better than the supervisory calculations. However, for our considerations, it is sufficient to know capital weights w_i for each obligor and a limit K.

The y_i are the decision variables for the portfolio manager. In our example, the value of y_i is determined by the total amount of money L_i that was lent to obligor i, i. e. $y_i \in [0,L_i]$. If $y_i < L_i$ then the part $L_i - y_i$ of the amount lent to obligor i has to be transferred from our bank to a third party risk buyer, e. g. an insurance company.

We require all variables to be rational numbers in problem (10) because the usual Turing machine computing model does not allow real-value variables using arbitrary precision. But it is not limiting our results since this is a usual assumption for proofs of complexity in all areas of application.

The corresponding decision problem for (10) is:

(11) Problem: Are there numbers $y_i \in [0, L_i]$ for given natural numbers R and K such that

$\Sigma_i r_i y_i \geq R$ and $\Sigma_i w_i y_i \leq K$

$y_i \in [0, L_i]$ rational numbers

$r_i, w_i > 0$ rational numbers

The economic interpretation of problem (11) is that we search for a portfolio structure satisfying a minimum return R and a risk limit K. An interesting result concerning this problem and the related optimisation problem (10) is the following:

(12) Lemma: If the $y_i \in [0, L_i]$ are integer variables then problem (11) is NP-hard.

Proof: see appendix.

Since we did not assume any specific risk measure for the w_i variables in lemma (12) and its proof, we know that integer optimisation of credit portfolios with respect to supervisory capital weights and limits is NP-hard. The same holds for Credit-Value-at-Risk based capital weights and limits.

We know that dynamic programming yields solutions for knapsack-like problems in pseudo-polynomial time. This will work if the returns and the capital weights in a portfolio do not influence each other like the sizes and the values of the items in the original INTEGER KNAPSACK problem.

However, if the capital weights w_i themselves depend from each element y_i in a solution, i. e. $w_i = f(y_1, y_2,\ldots, y_n)$ for each i and f is a non-linear, non-convex function, which is the case if Credit-Value-at-Risk based capital weights are used (for an overview of the difficult structure of Value-at-Risk functions see e. g. Pflug (2000)), then the usual dynamic programming approach which constructs a solution by evaluating intermediate results will not work at least for some instances of the problems (10) and (11). This is due to the fact that we cannot construct an optimal partial solution without knowing the final value of all y_i variables since all y_i variables influence all capital weights.

The usual dynamic programming approach constructing an optimal solution using "step by step" principles cannot be used to solve every possible instance of problems (10) and (11) in pseudo-polynomial time if the capital weights are determined by Credit-Value-at-Risk calculations.

The same holds if we use risk-adjusted return figures instead of constant coefficients r_i.

However, for real-world portfolios there are heuristic approaches to solve problems (10) or (11) approximately within reasonable (i. e. non-exponential) time. See e. g. Schlottmann & Seese (2001, 2002) for related computational results and further details.

We have shown that some credit risk problems of high relevance for real-world portfolio management are difficult to solve, i. e. NP-hard. By an analysis of credit portfolio modelling using our complexity criterion we have identified structural reasons for high complexity in this area of application. The complexity management guidelines derived from our complexity criterion are compatible with the methods applied within existing portfolio credit risk models to achieve lower computational complexity bounds.

In the field of stock portfolio management which is related to our considerations about credit portfolio optimisation in problem (11) there are some interesting results concerning our complexity criterion in the study of Kao, Nolte & Tate (2000). The goal of their study is to analyse the computational limits that occur in the case of two or more investment alternatives (stocks) $S_1, S_2, \ldots, S_k$ if we ask for the optimal allocation vector $x = (x_1, .., x_k)$

of these assets given a certain type of utility function for the regarded investor. Each stock has an associated discrete probability distribution describing the probability of stock returns using the following definition:

(13) Definition: For a given real number $\mu > 0$ and integers m_1, m_2 satisfying $m_1 < m_2$ we define $m = m_2 - m_1 + 1$ and $\Delta = \{j\ \mu : j = m_{1,\ldots,}\ m_2\}$. There are $k \in N$ investment alternatives (stocks) $S_1,\ldots,S_k$. The returns of each stock i can only take values from Δ, and $S_i(\beta)$ denotes the probability that the (ex ante) return of stock i is equal to $\beta \in \Delta$. For each stock i we know the complete probability function of its returns.

For the description of the investor's utility function or investment style, the following definition is proposed:

(14) Definition: Given is a desired target return α. Using the portfolio data from definition (13),

$RA_b(\alpha, x)$ denotes the smallest probability that the return of the asset allocation vector $x = (x_1, .., x_m)$ is at most α% over all joint distributions for $S_1,\ldots,S_k$

$RA_w(\alpha, x)$ denotes the largest probability that the return of the asset allocation vector $x = (x_1, .., x_m)$ is at most α% over all joint distributions for $S_1,\ldots,S_k$

$RA_a(\alpha, x)$ denotes the average probability that the return of the asset allocation vector $x = (x_1, .., x_m)$ is at most α% over all joint distributions for $S_1,\ldots,S_k$

$AG_b(\alpha, x)$ denotes the largest probability that the return of the asset allocation vector $x = (x_1, .., x_m)$ is at least α% over all joint distributions for $S_1,\ldots,S_k$

$AG_w(\alpha, x)$ denotes the smallest probability that the return of the asset allocation vector $x = (x_1, .., x_m)$ is at least α% over all joint distributions for $S_1,\ldots,S_k$

$AG_a(\alpha, x)$ denotes the average probability that the return of the asset allocation vector $x = (x_1, .., x_m)$ is at least α% over all joint distributions for $S_1,\ldots,S_k$.

Due to Kao et al. (2000), RA describes a risk-averse investor who prefers minimisation of loss over maximisation of maximising profits while AG describes an aggressive investor who has the opposite preferences. In this study, there are three subtypes for each of these two investment styles depending on the investor's choice which probability is to determine the investment decision: A best-case risk-averse investor minimises RA_b by choosing an appropriate x, while a worst-case risk-averse investor minimises RA_w, and an average-case risk-averse investor minimises RA_a. In contrast, an aggressive investor will choose x such that RA_b is maximised if he is best-case, such that RA_w is maximised if he is worst-case or such that RA_a is maximised otherwise.

Based on the above definitions the following result could be proved:

(15) Theorem: Given S_1, S_2, and α, an optimal asset allocation x for a worst-case risk-averse investor minimizing $RA_w(\alpha,x)$ can be computed in $O(m^2 \log m)$ arithmetic operations.

In the proof of theorem (15) Kao et al. use a binary tree that describes the contents of contingency tables for the joint probability distributions of the returns from the two stocks. Each node of the constructed binary tree is labelled with the contents of two variables that represent an intermediate result during the search for the optimal x that minimises $RA_w(\alpha,x)$. This tree structure is essential for the lower computational complexity bound compared to a trivial algorithm requiring $O(m^3)$ arithmetic operations.

Due to mathematical properties (dualities) concerning the relationships between the different investor types, the computational complexity stated in theorem (15) also holds for the best-case risk-averse, the worst-case aggressive and the best-case aggressive investor. However, this is not the case for the average risk-averse or aggressive investor, since these average cases ask for the complete subset of contingency tables describing the joint distribution of the two stocks' returns that satisfies the minimum (maximum) conditions for $RA_a(\alpha, x)$ ($AG_a(\alpha, x)$). There is no known algorithm solving this problem in polynomial time (measured by m). See Kao et al. (2000) for a formal proof and a randomised algorithm for an approximation of the average cases' optimal solution.

We have to remind here that all these results belong to the two stock (k = 2) case and all the above complexity bounds are related to m, the dimension of the probability functions of possible returns. Remind that even in the two stock case, we know no polynomial time algorithm for the average investor's problem.

For k > 2 stocks, the following theorem was proved in Kao et al. (2000).

(16) Theorem: The existence of a greedy algorithm for finding the optimal asset allocation x for k > 2 stocks and a worst-case investor implies P = NP.

The proof of theorem (16) uses a reduction of a variant of the 3-DIM-MATCHING problem, which is known to be NP-complete (cf. e. g. "SP1" in Garey & Johnson (1979), p. 221).

Theorem (16) implies that unless P = NP we cannot expect to find a deterministic algorithm which exactly solves the k stock portfolio problem for k > 2 in polynomial time (still measured by m).

The results by Kao, Nolte & Tate show that even in a quite simple economic setting, where only a small number of stocks and simple investment strategies are considered and parameter estimation and prediction problems are absent, the resulting optimisation problems are very difficult from the perspective of computational complexity. From the viewpoint of our complexity criterion, the absence of grids plays an essential role in obtaining lower computational bounds for some problems since Kao et al. explicitly use tree structures to reduce the complexity of some calculations.

4.4. Operational risk

Operational risk (abbreviated by OpRisk) for financial or industrial companies (or other organisations) is a synonym for unexpected losses of profits or cash flows caused by failures of management or internal controls, changing markets, products, services, technologies, human error and/or fraud, failures of information systems or by unmanageable events and complex operations (see e. g. Marshall (2001) or King (2001)

for details). It is currently one of the most important and most difficult management problems for many companies throughout the world.

The management of OpRisk requires solutions for many NP-hard problems, e. g. capital market prediction, financial risk management, optimisation problems, scheduling, operations management, facility management, total quality management, reliability engineering, statistical process control, control of non-linear dynamical systems and complex organisations. Large grids and irregular flow of information cause high complexity in many of these problems as indicated in the above sections, so our criterion is of implicit relevance concerning this point.

Furthermore, there are strong dependencies between staff, process elements, information systems, external markets and other organisational elements. These structures, their connections and their dynamics lead to a more complex overall management problem. In a highly connected organisation a failure of a single element or a wrong decision can already result in huge losses and often in other unpredictable consequences, cf. the consequences of September 11, 2001.[6] Therefore, we want to point out that our complexity criterion is explicitly applicable here. An operational risk manager has to keep an eye on the grid size of the organisation (smaller embedded grids are preferable), on the regularities of the organisational elements (more regular patterns are more manageable) and on the dependencies between organisational elements (less pairwise dependencies lead to lower overall risk). Of course, these general considerations cannot be perfectly applied to each real world organisational element. However, they provide an explicit modelling and understanding of complexity in operational risk management without assuming too much data requirements. This is very important since in real world operational risk management the data that could be used as a basis for choosing distributions of risk factor variables X_i and for estimation of parameters of these distributions is usually missing or at least the history of collected data is too short and cannot be extended because the events that are to be analysed occur very seldom.

Due to these problems concerning the necessary databases for the usage of sophisticated stochastic models, simple non-structural, macro-level numbers like the company's total profit, turnover etc. are currently used as proxies for quantitative operational risk measures. But since there is a strong need for explicit measures of organisational complexity to allow better quantitative evaluation of operational risk, our grid criterion, particularly the well-defined tree width of embedded grids in the organisational structure, is obviously of high relevance in this context. It can be used to evaluate different organisations from a micro-level structural approach. So our considerations about the tiling problem and the three derived paths to manage complexity are also very useful here.

We think that it is one of the areas where a synergy from parametric and dynamic complexity together with other disciplines could be helpful to understand the appearing problems and to find possible solutions.

[6] E. g. whole companies were destroyed by the terrorist attacks that hit two single, but very important buildings.

CONCLUSION

We have provided a structural criterion of complexity in this paper which is important in different theoretical and application-oriented contexts. Starting from computational complexity we have analysed topological graph theory problems to observe how the presence of large grids inside the input structures influences the complexity. A main conclusion is that graphs without large grids are closely related to trees, and restricting the input for computational problems to such tree-like structures leads to non-complex problems. Therefore, we have found a first way to make complex problems more tractable by restricting their structure. For problems where the grid sizes cannot be bounded we have identified two other possibilities to avoid high complexity: Either the local structure inside the large grids has to be very regular, or the flow of information between the grid elements has to be restricted. Otherwise, if a problem is not restricted in at least one of these three structural criteria the problem will be complex (i. e. NP-complete).

After our considerations about graphs and related problems we have analysed the relationship between the structure of dynamic systems and their complexity. It has been shown that dynamic systems containing recursive iterations of a single non-linear function can lead to complex problems, even if the recursive iterations are based on quite simple loops. This is important for many market-driven dynamic systems like capital markets. We found indications that our complexity criterion is useful to manage complexity in these areas, too, since our analysis of the iteration steps in simple dynamic systems has led to the discovery of large grids inside the communication graphs of the regarded variables' contents. This is a basis for analysing more complicated dynamic systems by examining the interaction between their components.

We have considered the presence of large grids and the resulting complexity of problems in financial applications like dynamic pricing in auctions or capital markets, and risk management. For portfolio credit risk management, we have shown that selected problems in portfolio credit risk management are NP-hard or NP-complete and used our complexity criterion to provide structural reasons for the complexity of credit portfolio problems. Moreover, we have pointed out that our criterion is implicitly used in many real-world credit risk applications to make complex problems tractable. In the very recent field of Operational risk management there is a need for structural complexity measures and complexity management guidelines, so our criterion is of high relevance in this context, too.

Besides the results and indications that we have discussed, there are still many open questions. For instance, there is a need for further empirical validation of our concepts in real-world applications. The proposed grid-oriented criterion can lead to tools which support complexity engineering of real life problems if the corresponding parameters influencing the complexity can be controlled, e. g. by regarding designs of smaller grid sizes or with more regular patterns, or by using organisational structures which avoid high complexity in our sense. Finally, we think that we have contributed an idea that makes a difference in understanding of complexity.

APPENDIX

Proof of Lemma (5):

We are interested in the probability Prob ($X \leq K$) that the bank does not default at the time horizon. Since $X = \Sigma_i X_i L_i$ this means that we have to determine all possible subsets $A' \subseteq \{L_1, L_2,\ldots,L_n\}$ satisfying $\Sigma_{a\in A'} L_a \leq K$ since Prob ($X \leq K$) is the sum of the probabilities of all possible states for the Bernoulli variables leading to possible aggregated losses $\leq K$.

From a decision problem view this is at least as hard as solving the following problem at most $\Sigma_i L_i$ times for each $k \in \{1,\ldots, \Sigma_i L_i\}$:

(*) Problem: Given is the set $\{L_1, L_2, \ldots, L_n\}$ and a fixed integer k. Is there a subset $A' \subseteq \{L_1, L_2, \ldots, L_n\}$ satisfying $\Sigma_{a\in A'} L_a = k$?

But this decision problem is NP-complete since it we can construct an equivalent instance of this problem from any given instance of the SUBSET SUM problem ("SP13", cf. Garey & Johnson (1979), p. 223):

SUBSET SUM Problem: Given is a finite set A, an integer weight $s(a) > 0$ for each element $a \in A$ and an integer b. Is there a subset $A' \subseteq A$ satisfying $\Sigma_{a\in A'} s(a) = b$?

We can transform an instance of SUBSET SUM into an equivalent instance of problem (*) by using a polynomial time calculable 1-1 function f: A -> N that maps a subsequent natural number to each element $a \in A$ starting from $f(a) = 1$ for the first element $a \in A$. We set $n = |A|$. Choosing $L_{f(a)} = s(a)$ for each $a \in A$ and setting $k = b$ yields an equivalent instance of (*). By construction of f and the variables $L_{f(a)}$ and k, A' is a solution for SUBSET SUM if and only if A' a solution of problem (*) and vice versa.

Since our decision problem (*) is a restricted version of our default probability calculation problem, the latter problem is also NP-hard.

q. e. d.

Proof of Lemma (12):

Obviously for integer variables y_i, problem (11) is in NP since a non-deterministic algorithm can guess a solution for the y_i variables and verify it in polynomial time.

Now consider an instance of the following problem ("MP10", cf. Garey & Johnson (1979), p. 247):

INTEGER KNAPSACK Problem: Given is a finite set U, an integer size $s(u) > 0$, an integer value $v(u) > 0$ for each element $u \in U$ and positive integers S,V. Is there an assignment of a non-negative integer c(u) to each $u \in U$ such that $\Sigma_{u\in U} c(u)\, s(u) \leq S$ and $\Sigma_{u\in U} c(u)\, v(u) \geq V$?

We can construct an equivalent instance of the decision problem (11) for a given INTEGER KNAPSACK problem instance by using a polynomial time calculable 1-1 function f: U -> N that assigns a subsequent natural number to each element $u \in U$ starting from f(u) = 1 for the first element $u \in U$ and by setting n = |U|, $r_i = v(f^{-1}(i))$, $w_i = s(f^{-1}(i))$, R = V, K = S.

The only problem is to determine an upper bound L_i for each obligor $i \in \{1,...,n\}$, and therefore an upper bound for y_i such that $c(f^{-1}(i))$ is a solution of the INTEGER KNAPSACK problem instance if and only if y_i is a solution of problem (11).

To obtain an upper bound for L_i (or equivalently c(u)) we look at the inequality $\Sigma_{u \in U}\, c(u)\, s(u) \leq S$ and consider the smallest possible value for s(u) on the left hand side. In the relevant case s(u) = 1 this inequality can only be satisfied if c(u) is bounded by S, i. e. $c(u) \leq S$ for all $u \in U$. For s(u) > 1, c(u) must be < S to satisfy this inequality.

Therefore, if we choose $L_i = S$ for all i and consider the above transformation of the variables, we have found a polynomial time reduction from INTEGER KNAPSACK to our problem (11). Since INTEGER KNAPSACK is known to be NP-complete, our problem (11) is NP-hard. Moreover, problem (11) is in NP, therefore it is NP-complete.

q. e. d.

REFERENCES

1. Arnborg, S., Lagergren, J. & Seese, D. 1991, 'Easy problems for tree-decomposable graphs', *Journal of Algorithms* 12, pp. 308-340.
2. Akcoglu, K., Aspnes, J., DasGupta, B., Kao, M. 2001, 'Opportunity cost algorithms for combinatorial auctions', http://www.cs.yale.edu/homes/aspnes/auctions-abstract.html.
3. Aspnes, J., Fischer, D., Fischer, M., Kao, M. & Kumar, A. 2000, 'Towards understanding the predictability of stock markets from the perspective of computational complexity', preprint to appear in *Proc. of 12th annual ACM SIAM Symposium of Discrete Algorithms*.
4. Basel Committee for Banking Supervision, 2001, The New Basel Capital Accord, www.bis.org/publ/bcbsca03.pdf.
5. Black, F. & Scholes M., 'The pricing of options and corporate liabilities', *Journal of Political Economy* 81 (1973), pp. 637-659.
6. Blondel V. & Tsitsiklis, J. 2000, 'The boundedness of all products of a pair of matrices is undecidable', *Systems & Control Letters* 41, pp. 135-140.
7. Blum, L., Shub M. & Smale, S. 1989, 'On the theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines', *Bulletin of the Amer. Math. Soc.* 21, pp. 1-46.
8. Blum, L. & Smale, S. 1993, 'The Goedel incompleteness theorem and decidability over a ring', in *From Topology to Computation: Proc. of the Smalfest*, eds M. Hirsch, J. Marsden, M. Shub, Springer, New York, pp. 321-339.
9. Blum, L., Cucker, F., Shub, M. & Smale, S. 1998, *Complexity and real computation*, Springer, New York.

10. Brandstaedt, A., Bank Le, V. & Spinrad. J. 2000, 'Graph classes: A survey', *SIAM Monographs on Discrete Mathematics and Applications*.
11. Carkhuff, R., Carkhuff, C., Cook, A., O'Brien, W., Tisdale, D. & Rayson, P. 2000, 'Complexity, Simplicity and Collaboration', in *Complexity and Complex Systems in Industry*, Conference Proceedings, eds I. Mc Carthy & T. Rakotobe-Joel, University of Warwick, pp. 86-91.
12. Courcelle, B., Makowsky, J. & Rotics, U. 2000, 'Linear time solvable optimization problems on graphs of bounded clique width', *Theory Comput. Systems 33*, pp. 125-150.
13. CreditSuisse Financial Products (1997), CreditRisk+TM - a credit risk management framework, available via http://www.csfp.co.uk/creditrisk/assets/creditrisk.pdf.
14. Downey, R. & Fellows, M. 1999, *Parameterized Complexity Theory*, Springer, Berlin.
15. Duffie, D. & Singleton, K 1999, 'Modelling term structures of defaultable bonds', *Review of Financial Studies* 12 (1999) 4, 687-720.
16. Garey, M. & Johnson, D. 1979, *Computers and Intractability*, W. H. Freeman and Company, New York.
17. Hromkovic, J. 1997, *Communication complexity and parallel computing*, Springer, Berlin.
18. Kao, M., Nolte, A. & Tate, S. 2000: 'The risk profile problem for stock portfolio optimization' (extended abstract) STOC 2000, Portland Oregon USA, pp. 228-234.
19. King, J. 2001, *Operational Risk*, John Wiley & Sons, New York, 2001.
20. Lando, D. 1998, 'On Cox processes and credit risky securities', *Review of Derivatives Research* (1999) 2, pp. 99-120.
21. Lengauer, T. 1990, *Combinatorial algorithms for integrated circuit layout*, John Wiley & Sons, Chicester.
22. Marshall, C. 2001, *Measuring and Managing Operational Risk in Financial Institutions*, John Wiley & Sons, Singapore.
23. Merton, R. 1973, A rational theory of option pricing, *Bell Journal of Economics and Management Science* (1973) 4, pp. 141-183.
24. Papamiditriou, C. 1994, *Computational Complexity*, Addison-Wesley, Reading.
25. Pflug, G. 2000, 'Some remarks on the Value-at-Risk and the Conditional Value-at-Risk', in: Uryasev, S. (ed.), *Probabilistic constrained optimization*, Kluwer, Dordrecht pp. 272-281.
26. Reason, J. 1987, 'The Chernobyl Errors', *Bulletin of the British Psychological Society* 40, pp. 201-206.
27. Robertson, N. & Seymour, P. 1984, 'Graph minors III. Planar tree-width.', *J. Combin. Theory* Ser. B. 36, pp. 49-64.
28. Robertson, N. & Seymour, P. 1986, 'Graph minors V. Excluding a planar graph.', *J. Combin. Theory* Ser. B. 41, pp. 92-114.
29. Schlottmann, F. & Seese, D. 2001, 'A hybrid genetic-quantitative method for risk-return optimisation of credit portfolios', Quantitative Methods in Finance 2001 conference abstracts, University of Technology, Sydney, Australia, p. 55, full paper available via ww.business.uts.edu.au/finance/resources/qmf2001/Schlottmann_F.pdf.
30. Schlottmann, F. & Seese, D. 2002, 'Hybrid multi-objective evolutionary computation of constrained downside risk-return efficient sets for credit portfolios', accepted for Computing in Economics and Finance 2002 conference, Aix-en-Provence, France, to appear.
31. Seese, D. & Schlottmann, F. 2001, 'Structural reasons for high complexity: A survey on results and problems', working paper, University of Karlsruhe.

32. Seese, D. & Schlottmann, F. 2002, 'Large grids and local information flow as a reason for high complexity', in: Frizelle, G. & Richards, H., *Tackling industrial complexity: the ideas that make a difference*, Proceedings of the 2002 conference of the Manufacturing Complexity Network, University of Cambridge, pp. 193-207.
33. Siegelmann, H. & Sontag, E. 1995, 'On the computational power of neural nets', *Journal of Computer and System Sciences* 50, pp. 132-150.
34. Sontag, E. 1995, 'From linear to non-linear: some complexity comparisons', *Proc. IEEE Conf. Decision and Control*, New Orleans, pp. 2916-2920.
35. Sterman, J. 2000, Business dynamics, McGraw-Hill, Boston.
36. Tschoegl, A. 2000, 'The Key to Risk Management: Management', in *Risk Management*, eds M. Frenkel, U. Hommel, Springer, Heidelberg, pp. 103-120.
37. West, D. 1996, *Introduction to Graph Theory*, Prentice-Hall, Englewood Cliffs.
38. Wilson, T. 1997a, Portfolio Credit Risk (I), in: Risk (1997) 9, pp. 111-119.
39. Wilson, T. 1997b, Portfolio Credit Risk (II), in: Risk (1997) 10, pp. 56-61.