Proceedings of The 21st

Czech-Japan Seminar on Data Analysis and Decision Making

Kamakura, Japan
November 23rd - 26th, 2018

Edited by Shao-Chin Sung & Milan Vlach

Organizing institution: Aoyama Gakuin University, Japan

© Shao-Chin Sung, Milan Vlach, 2018

Committees

Programme Chairs:

- Shao-Chin Sung (Aoyama Gakuin University)
- Milan Vlach (Charles University in Prague)

Programme Committee:

- Čepek, Ondřej (Charles University in Prague)
- Martin Gavalec (University of Hradec Králové)
- Masahiro Inuiguchi (Osaka University)
- Hiroaki Ishii (Kwansei Gakuin University)
- Takeshi Itoh (Tohoku University)
- Radim Jiroušek (University of Economics, FM Jindrichuv Hradec)
- Vilém Novák (University of Ostrava)
- Irina Perfilieva (University of Ostrava)
- Jaroslav Ramík (Silesian University in Opava)
- Shogo Shiode (Kobe Gakuin University)
- Jiřina Vejnarová (Czech Academy of Sciences, Prague)

Organizing Committee

- Shao-Chin Sung (Aoyama Gakuin University)
- Natsumi Takahashi (Aoyama Gakuin University)

Organizing Staffs

- Naoya Makiishi (Aoyama Gakuin University)
- Natsuki Niwa (Aoyama Gakuin University)
- Masaya Tanaka (Aoyama Gakuin University)

Foreword

Dear Colleagues,

Czech-Japan Seminars on Data Analysis and Decision Making originated by the two meetings that were held in Ishikawa and Celadna in 1999. These meetings were financially supported by an international collaborative research project between Czech Republic and Japan, and were successful in achieving new results in the area of data analysis and decision making under uncertainty.

For initiating further mutual research cooperation, the seminars were alternatively organized in the Czech Republic and Japan every year. This year, with the support by Center of Advanced Information Technology at Aoyama Gakuin University, the 21st meeting will be taken place at Kamakura in November 2018.

Shao-Chin Sung
Aoyama Gakuin University
Kamakura, November 2018

Table of Contents

Optimal Design of Production Systems

Milan VLACH[1] and Helena BROŽOVÁ[2]

[1] *Charles University* and *Czech Academy of Sciences*
Malostranské náměstí 25, 118 00 Praha 1
Czech Republic
vlach@ktiml.mff.cuni.cz

[2] *Czech University of Life Sciences*
Kamýcká 129, 165 00 Praha 6
Czech Republic
brozova@pef.czu.cz

1 ABSTRACT

Consider the standard linear programming model for allocating resources to activities so as to attain a given economic objective; that is, the problem

$$\text{maximize } cx \text{ subject to } Ax \leq b \text{ and } x \geq 0,$$

where x is a real $(n, 1)$-vector of variables, c is a given real $(1, n)$-vector, b a given real $(m, 1)$-vector, and A is a given real (m, n)-matrix. To be more explicit regarding interpretation, we assume that:

- For each activity j, x_j is a decision variable the values of which represent the levels of activity j, and c_j denotes the payoff (profit) per unit of activity j.
- For each resource i, b_i is the available amount of resource i.
- For each resource i and each activity j, a_{ij} is the amount of resource i required to perform a unit of activity j.

Milan Zeleny, in a number of papers (see, for example, [3, 4, 5]), proposed and developed a specially structured linear programming model called (De Novo) optimal system design. In Zeleny's model, the components of b are considered, ceteris paribus, to be variables the values of which are restricted by the condition

$$pb \leq B, \; b \geq 0$$

where p is a real nonnegative $(1, m)$-vector whose components are interpreted as the unit prices of resources, and B is a given positive number representing the total available budget.

To indicate that the components of b are now real variables we change the notation and use the letter y instead of b. Then the problem of optimal system design can be formulated as that of

$$\begin{array}{ll} \text{maximizing} & cx + 0y \\ \text{subject to} & Ax - Ey \leq 0, py \leq B, x \geq 0, y \geq 0. \end{array}$$

It turns out that this specially structured linear programming problem can be solved with the help of the following continuous knapsack problem:

$$\begin{array}{ll} \text{maximize} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{subject to} & (pA)_1x_1 + \cdots + (pA)_nx_n \leq B, \\ & x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0, \end{array}$$

provided all c_j and all $(pA)_j$ are positive.

The situation is more complicated when we have to deal with multiple criteria; that is, with the problem

$$\begin{array}{ll} \text{maximizing} & Cx + 0y \\ \text{subject to} & Ax - Ey \leq 0, py \leq B, x \geq 0, y \geq 0, \end{array}$$

where C is a real (q, n)-matrix of coefficients of q objective functions. The reason is that it is not clear what is meant by maximization of q functions.

In this paper we follow our previous results on De Novo approach presented in [2]. First we briefly recall De Novo methodology both for the single criterion and multiple criteria problems as proposed by Zeleny, rectify some errors in an alternative solution proposed by Shi [1], and show how the problem with upper bounds on levels of activities can also be solved by means of continuous knapsack problem. Then we present examples of adaptation of De Novo approach for models with both capacity and requirement constraints, where the transformation to continuous knapsack problem is not possible. However, the so called optimum-path ratio for achieving the best performance for a given budget B is again applicable.

Acknowledgement
This research was supported by the grant of the Czech Science Foundation No. P402/12/G097 DYME - Dynamic Models in Economics.

References

[1] Y. Shi. Studies on optimum-path ratios in multi-criteria De-Novo programming problems. *Computers Math. Applic.*, 29:43–50, 1995.

[2] M. Vlach, H. Brožová. Remarks on De Novo approach to multiple criteria optimization. In *36th International Conference Mathematical Methods in Economics*, pages 618–623, MatfyzPress, 2018.

[3] M. Zeleny. Optimal system design with multiple criteria: De Novo programming approach. *Engineering Costs and Production Economics*, 10:89–94, 1986.

[4] M. Zeleny. Optimal Given System vs. Designing Optimal System: The De Novo Programming Approach. *International Journal of General System*, 17:295307.

[5] M. Zeleny. Multiobjective Optimization, Systems Design and De Novo Programming. In *Handbook of Multicriteria Analysis*, pages 243–262, Springer, 2010.

A Dynamic Programming Approach and Parallel Computing Approach for Computing All-Terminal Reliability of Small Networks with Uniform Probability

Naoya MAKIISHI[1], Natsumi TAKAHASHI[2] and Shao-Chin SUNG[3]

[1,2,3] *Aoyama Gakuin University*
5-10-1 Fuchinobe, Chuo-ku, Sagamihara Kanagawa, 252-5258, Japan
[1] c5617165@aoyama.jp, [2] takahashi@ise.aoyama.ac.jp, [3] son@ise.aoyama.ac.jp

Abstract

All-terminal reliability of a given network is a probability where all nodes in the network are connected. All-terminal reliability is computed for a network where all edges have a probability. Some algorithms which compute all-terminal reliability are proposed, e.g. enumerating all connected spanning subgraphs, factoring algorithm etc. In this paper, we are concerned with the problem of computing all-terminal reliability of network with uniform probability. This problem is known to be $\#P$-complete. When all edges have the same reliability, all-terminal reliability of a given network can be obtained based on the number of connected spanning subgraphs in terms of the number of edges in those subgraphs. Based on this fact, we propose a dynamic programming approach for computing all-terminal reliability. Moreover, we have implemented the proposed approach, and obtained the number of connected spanning subgraphs in terms of number of edges for all networks with up to 9 vertices. According to this result, the network with the largest all-terminal reliability among all networks consists of given numbers of vertices and edges are identified. To reduce computational time, we propose parallel computing approach for computing all-terminal reliability which is based on the proposed dynamic programming approach. As well as dynamic programming approach, we computed all-terminal reliability for all networks with up to 9 vertices by proposed parallel computing approach. We have implemented this approach with two computers, and reduced computational time compared to dynamic programming approach.

Keywords : All-Terminal Reliability, Dynamic Programming, Parallel Computing

Succinctness of Switch-List Representations of Boolean Functions

Ondřej Čepek and Miloš Chromý

Department of Theoretical Computer Science and Mathematical Logic,
Faculty of Mathematics and Physics, Charles University,, Prague, Czech Republic
cepek@ktiml.mff.cuni.cz, chromy@ktiml.mff.cuni.cz

Abstract

In this paper we focus on a slightly unusual way how to represent Boolean functions, namely on representations by switch-lists. Given a truth table representation of a Boolean function f the switch-list representation of f is a list of Boolean vectors from the truth table which have a different function value than the preceding Boolean vector in the truth table. The main aim of this paper is to compare switch-list representations with a number of standard representations (such as CNF, DNF, and OBDD) with respect to their relative succinctness, and hence to include switch-list representations in the Knowledge Compilation Map [7].

Keywords: Boolean functions, Knowledge compilation

1 Introduction

A Boolean function on n variables is a mapping from $\{0,1\}^n$ to $\{0,1\}$. This concept naturally appears and is extensively used in many areas of mathematics and computer science. There are several different ways in which a Boolean function may be represented. Common representations include truth tables (TT – with 2^n rows where a function value is explicitly given for every binary vector), list of models (MODS – list of binary vectors on which the function evaluates to 1), various types of Boolean formulas (including CNF and DNF representations), various types of binary decision diagrams (BDDs, FBDDs, OBDDs), and Boolean circuits.

The task of transforming one of the representations of a given function f into another representation of f (e.g transforming a MODS representation into an OBDD or a circuit into a DNF) is called knowledge compilation. For a comprehensive review paper on knowledge compilation see [7]. One of the parameters important in knowledge compilation is the *succinctness* of representations, which roughly speaking describes, how large the output representation is with respect to the size of the input representation. Precise definition of this notion will be given later in this text.

In this paper we shall study the succinctness of quite interesting but less usual representations of Boolean functions, namely the representation by intervals of truepoints and a the closely related representation by switch-lists. Let f be a Boolean function and let us fix some order of its n variables. The input binary vectors can be now thought of as binary numbers (with bits in the prescribed order) ranging form 0 to $2^n - 1$. An interval representation is then an abbreviated TT or MODS representation, where instead of writing out all the input vectors (binary numbers) with their function values, we write out only those binary numbers x for which $f(x) = 1$ (x is a truepoint of f) and simultaneously $f(x-1) = 0$ ($x-1$ is a falsepoint of f) and those binary numbers y for which $f(y) = 1$ (y is a truepoint of f) and simultaneously $f(y+1) = 0$ ($y+1$ is a falsepoint of f). Thus the function is represented by an ordered list of such pairs $[x, y]$ of integers, each pair specifying one interval of truepoints. Note that $x = y$ for those pairs which represent an interval with a single truepoint. In this paper we shall use a related representation by switch-lists. A switch is a vector (binary number) x such that $f(x-1) \neq f(x)$. A switch-list is an ordered list of all switches of a given function. A switch-list of f together with the function value $f(0,0,\ldots,0)$ forms a switch-list representation of f.

Note here, however, that changing the order of variables may dramatically change the length of switch-list and interval representations. It is easy to construct functions with a single switch (and a single truepoint interval) under one permutation of variables and $\Omega(2^n)$ switches (and truepoint intervals) under another permutation.[1] Hence the length of the switch-list (and interval) representation may be $O(n)$ for one permutation of variables and $\Omega(n2^n)$ for another permutation. On the other hand, there exist Boolean functions (e.g. a parity function), where listing all switches (truepoint intervals) is asymptotically as space consuming as writing out the full truth table regardless of the chosen variable order.

Interval representation of Boolean functions was introduced in [13], where the input was considered to be a function represented by a single interval (two n-bit numbers x, y) and the output was a DNF representing the same Boolean function f on n variables, i.e. a function which is true exactly on binary vectors (numbers) from the interval $[x, y]$. This knowledge compilation task originated from the field of automatic generation of test patterns for hardware verification [12, 10]. In fact, the paper [13] achieves more than just finding some DNF representation of the input 1-interval function – it finds in polynomial time the shortest such DNF, where "shortest" means a DNF with the least number of terms. Thus [13] combines a knowledge compilation problem (transforming an interval representation into a DNF representation) with a knowledge compression problem (finding the shortest DNF representation).

In [6] the reverse knowledge compilation problem was considered. Given a DNF, decide whether it can be represented by a single interval of truepoints with respect to some permutation of variables (and in the affirmative case output the permutation and the two n-bit numbers defining the interval). This problem can be easily shown to be co-NP hard in general (it contains tautology testing for DNFs as a subproblem), but was shown in [6] to be polynomially solvable for "tractable" classes of DNFs (where "tractable" means that DNF falsifiability can be decided in polynomial time for the inputs from the given class). The algorithm presented in [6] runs in $O(n\ell f(n, \ell))$ time, where n is the number of variables and ℓ the total number of literals in the input DNF, while $f(n, \ell)$ is the time complexity of falsifiability testing on a DNF on at most n variables with at most ℓ total literals. This algorithm serves as a recognition algorithm for 1-interval functions given by tractable DNFs. This result was later extended in [11] to monotone 2-interval functions, where an $O(\ell)$ recognition algorithm for the mentioned class was designed. Recently, these results were further extended to k-interval functions for arbitrary k, where the recognition algorithm runs in polynomial time in the length of the input formula for any constant k (the complexity is of course exponential in k) [5]

Switch list representations have an added advantage over the truepoint intervals representations. Given a DNF, its logical negation can be represented by a CNF of the same length (and vice versa), and the transformation is purely mechanical (replace disjunctions by conjunctions, conjunctions by disjunctions, and negate all literals). Clearly, both the function and its negation have the same switch-lists and the representations differ only by opposite values of $f(0, 0, \dots, 0)$. Thus the results valid for the DNF language can be easily rewritten into results for the CNF language. This is not true for interval representations because an interval of truepoints turns into an interval of falsepoints by negation, and hence the truepoint intervals representations of a function and its negation differ, and even the number of intervals may be different (although the difference is at most one).

For this reason, we shall use the switch-list representations throughout this paper. It is not a limiting assumption in any way: clearly, the list of intervals can be easily compiled from the list of switches and the function value $f(0, 0, \dots, 0)$, and vice versa.

As a final remark let us note that the combination of results from [6] and [13] gives a polynomial time minimization (optimal compression) algorithm for the class of 1-interval functions given by tractable DNFs, or in other words, for the 1-interval subclass of functions inside any tractable class of functions. DNF minimization (optimal compression) is a notoriously hard problem. It was shown to be Σ_2^p-complete [14] when there is no restriction on the input DNF (see also the review paper [15] for related results). It is also long known that this problem is NP-hard already for some tractable classes of DNFs – maybe the best known example is the class of Horn DNFs (a DNF is Horn if every term in it contains at most one negative literal) for which the NP-hardness was proved in [1, 8] and the same result for cubic Horn DNFs in [3]. There exists a hierarchy of subclasses of Horn DNFs for which there are polynomial time minimization algorithms, namely acyclic and quasi-acyclic Horn DNFs [9], and CQ Horn DNFs [2]. There are also few heuristic

[1] For instance $f(x_1, \dots, x_n) = x_1$ gives a single interval if x_1 is the first variable in the considered permutation and 2^{n-1} intervals when x_1 is the last variable (in the latter case every truepoint interval is a singleton as truepoints and falsepoints alternate regularly).

minimization algorithms for Horn DNFs [4]. Suppose we are given a Horn DNF. We can test in polynomial time using the algorithm from [6] whether it represents a 1-interval function and then (in the affirmative case) use the algorithm from [13] to construct a minimum DNF representing the same function as the input DNF. Thus we have a minimization algorithm for 1-interval Horn DNFs. It is an interesting research question in what relation (with respect to inclusion) is this class with respect to the already known hierarchy of polynomial time compressible subclasses of Horn DNFs (acyclic Horn, quasi-acyclic Horn, and CQ-Horn DNFs).

The paper is organized as follows. In the next section we shall introduce the necessary terminology and notation, and define the propositional languages studied in this paper. The main section that follows contains the succinctness results for the defined languages. We finish the paper with few concluding remarks.

2 Definitions and notation

A *Boolean function*, or a *function* in short, in n propositional variables is a mapping $f : \{0,1\}^n \to \{0,1\}$, where $x \in \{0,1\}^n$ is called a *Boolean vector* (a *vector* in short). A function f in n variables can be represented by a truth table, which is a list of all 2^n vectors together with their function values. Rather then listing all vectors, one can list only models of f (all vectors x for which $f(x) = 1$). The language of all such representations of all Boolean functions is called **MODS**, and each list of models for a particular function is called a *sentence* of the **MODS** language. Similarly we can consider sentences of non-models (all vectors x for which $f(x) = 0$) which define the language $\neg$**MODS** used in this paper for symmetry purposes.

Propositional variables $x_1, x_2, \ldots$ and their negations $\overline{x}_1, \overline{x}_2, \ldots$ are called *literals* (*positive* and *negative literals* respectively). An elementary conjunction of literals

$$T = \bigwedge_{i \in I} x_i \wedge \bigwedge_{j \in J} \overline{x}_j \tag{1}$$

is called a *term*, if every propositional variable appears in it at most once, i.e. if $I \cap J = \emptyset$. A *disjunctive normal form* (or DNF) is a disjunction of terms. Similarly, an elementary disjunction of literals

$$C = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \overline{x}_j \tag{2}$$

is called a *clause*, if every propositional variable appears in it at most once, i.e. if $I \cap J = \emptyset$. A *conjunctive normal form* (or CNF) is a conjunction of clauses.

It is a well known fact, that every Boolean function can be represented by DNFs and CNFs (typically by many different ones). By **DNF** we shall denote the propositional language of all DNFs (of all functions), and similarly **CNF** shall denote the language of all CNFs. Each individual DNF (or CNF respectively) is then called a *sentence* of the **DNF** (or **CNF** respectively) language.

A *Binary Decision Diagram (BDD)* is a rooted directed graph with two terminals labeled 0 and 1. Each non-terminal node is a decision node with exactly two outgoing edges. Each decision node corresponds to a propositional variable and the two outgoing edges correspond to the assignments of 0 and 1 to this variable. Each directed path from the root to a terminal thus corresponds to a (possibly partial) assignment of truth values to variables and the terminal specifies the function value for such an assignment. Let $<$ be a total order on the set PS of all propositional variables (we assume this set to be denumerable). An *Ordered Binary Decision Diagram (OBDD)* with respect to $<$ is a BDD such that on every path from the root to a terminal no two decision nodes correspond to the same variable and moreover every such path respects the prescribed order $<$. The second condition means that there does not exist a directed path p from the root to a terminal and two variables $x < y$, such that the decision node corresponding to y precedes the decision node corresponding to x on path p. All OBDDs with respect to $<$ form the language $\mathbf{OBDD}_<$ and the language **OBDD** is defined as the union of $\mathbf{OBDD}_<$ languages over all total orders on the set PS.

Now we are ready to define the two principal languages of this paper. Again, let $<$ be a total order on the set PS of all propositional variables, let X be a subset of PS of size n, and let f be a function on variables from X. Consider vector $x \in \{0,1\}^n$ where the bits of x correspond to the variables of X in the prescribed order $<$. Each such vector x can be in natural way identified with a binary number from $[0, 2^n - 1]$, so for every $x > 0$ the vector $x - 1$ is well defined. We call $x \in \{0,1\}^n$ a *switch* of f with respect to order $<$, if $f(x-1) \neq f(x)$. The list of all switches of f with respect to $<$ is called the *switch-list* of f with respect to $<$. The switch-list of f with respect to $<$ together with the function value $f(0)$ is called the *switch-list representation* of f with respect to $<$. The set of switch-list representations with respect to $<$ (of all functions) forms the propositional language $\mathbf{SL}_<$. Finally, the language $\mathbf{SL}$ is the union of $\mathbf{SL}_<$ languages over all total orders on the set PS.

Two sentences (possibly from two different propositional languages) are called *logically equivalent* if they represent the same function. Now we are ready to define the most important concept of this paper.

Definition 1 *A propositional language* $\mathbf{L}$ *is* at least as succinct *as another propositional language* $\mathbf{K}$*, denoted* $\mathbf{L} \leq \mathbf{K}$*, if and only if there exists a polynomial* p *such that for every sentence* $\alpha \in \mathbf{L}$ *there exists a logically equivalent sentence* $\beta \in \mathbf{K}$ *such that* $|\beta| \leq p(|\alpha|)$ *(where the size of the sentence is the number of bits necessary to encode it). If* $\mathbf{L} \leq \mathbf{K}$ *holds and* $\mathbf{K} \leq \mathbf{L}$ *does not (denoted* $\mathbf{K} \not\leq \mathbf{L}$*), we write* $\mathbf{L} < \mathbf{K}$.

The following diagram summarizing the succinctness relations of many commonly used propositional languages appeared in [7]).

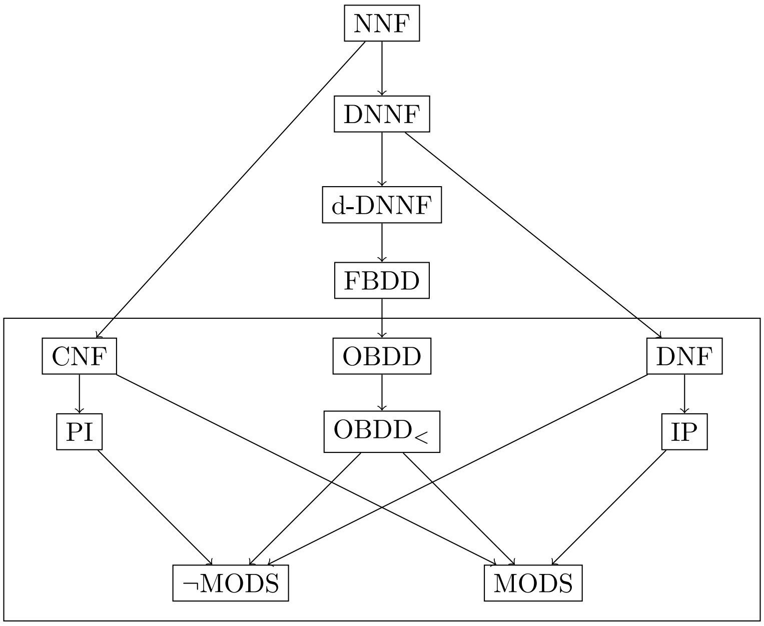


Figure 1: The diagram of succinctness relations taken from [7]. For symmetry reasons the language $\neg$**MODS** was added into the diagram. Each directed arc $A \longrightarrow B$ means that A is strictly more succinct than B, i.e. $A < B$.

The main aim of this paper is to add the languages $\mathbf{SL}_<$ and $\mathbf{SL}$ into the framed part of the diagram in Figure 1 and establish the succinctness relations to the languages already presented there. This goal is achieved in the next section.

3 Succinctness of switch list representations

In this section we prove the succinctness relations for $\mathbf{SL}_<$ and $\mathbf{SL}$ languages described in Figure 2. We will use one subsection for each relation, the numbering of the subsections corresponds to the arrow numbers in Figure 2.

3.1 $\mathbf{SL} < \mathbf{SL}_<$

This is the most obvious of the relations in Figure 2. The inequality $\mathbf{SL} \leq \mathbf{SL}_<$ follows from the fact that the language $\mathbf{SL}_<$ is a subset of the language $\mathbf{SL}$. To show that $\mathbf{SL}_< \not\leq \mathbf{SL}$ let us consider

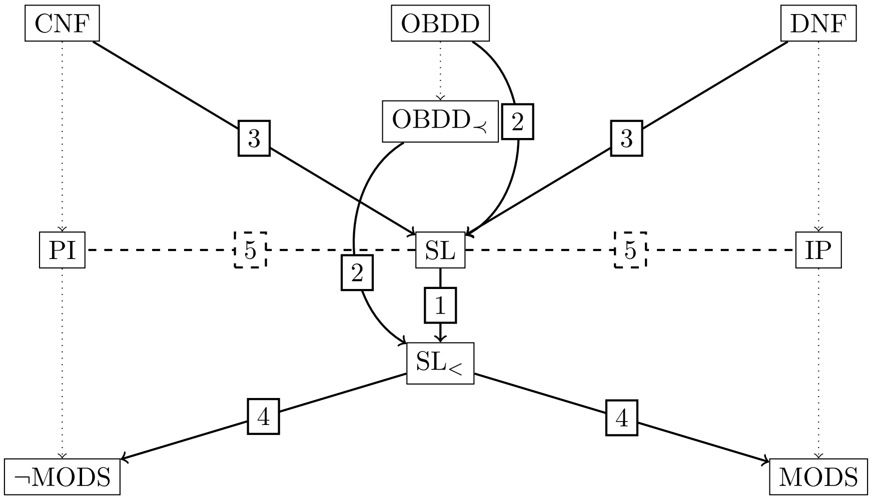


Figure 2: Solid arrows correspond to strict succinctness results, dashed lines to incomparability results, and dotted lines to known strict succinctness relations from Figure 1 which do not follow from transitivity using the solid arrows.

function $f(x_1, \dots, x_n, y) = y$ and two orders of its variables $<_1$ and $<_2$ where $y <_1 x_1 <_1 \dots <_1 x_n$ and $x_1 <_2 \dots <_2 x_n <_2 y$. Clearly, the SL representation of f with respect to $<_1$ has a single switch (all falsepoints precede all truepoints) while the SL representation of f with respect to $<_2$ has 2^{n+1} switches (falsepoints and truepoints alternate, so every vector is a switch). Thus the SL representation of f in the language $\mathbf{SL}_{<_2}$ is exponentially larger than the SL representation of f with respect to $<_1$ in the language **SL**. This simple example is not fully satisfactory, because f does not depend on variables $x_1, \dots, x_n$ and so $f(x_1, \dots, x_n, y) = f(y)$ is in fact a function in just one variable (and of course the exponential blowup disappears). This can be easily fixed by switching one falsepoint of f into a truepoint and considering $f(x_1, \dots, x_n, y) = y \vee (\bigwedge_{i=1}^{n} x_1, \dots, x_n)$. This function now depends on all $n+1$ variables, still has a single switch with respect to $<_1$ (we switched the last falsepoint with respect to $<_1$ to a truepoint, so all falsepoints still precede all truepoints), and still has exponential many switches with respect to $<_2$ (only the last two switches with respect to $<_2$ disappeared).

3.2 $\mathbf{OBDD}_< < \mathbf{SL}_<$ and $\mathbf{OBDD} < \mathbf{SL}$

It is easy to see that $\mathbf{SL}_< \not\leq \mathbf{OBDD}_<$ and $\mathbf{SL} \not\leq \mathbf{OBDD}$. A good example is the parity function on n variables which is symmetric, and thus changing the order imposed on the set of propositional variables changes neither the minimum size OBDD nor the minimum size SL representation (and therefore this example works both for the ordered and unordered case). The parity function is well known to have an OBDD of linear size in n, while the number of switches in any SL representation is exponential in n. The last fact follows e.g. from an easy observation that every second vector in the truth table that corresponds to an even number changes its parity when the last bit is flipped from 0 to 1 to get the next odd number. Thus every vector which corresponds to an odd number is a switch, and therefore the parity function has $\Omega(2^n)$ switches.

Now we shall prove $\mathbf{OBDD}_< \leq \mathbf{SL}_<$ and $\mathbf{OBDD} \leq \mathbf{SL}$ by taking an arbitrary SL representation of an arbitrary function f and constructing from it an equivalent OBDD representation with respect to the same order of variables. The idea behind the construction is quite simple. Fix the order of variables of the given function f, say $x_1, x_2, \dots, x_n$. Consider the complete binary decision tree of f which branches in the prescribed order, i.e. a tree with n levels of decision nodes and 2^n function values on level $n+1$. Replace all function values with just two terminals 0 and 1, that is, connect the outgoing edges from the decision nodes that branch on the value of x_n directly to these two terminals. Then start a bottom-up process of eliminating redundant decision nodes. In this process, every decision node with both outgoing edges leading to the same terminal t is deleted and replaced by an edge from its parent node to the terminal t. This process obviously stops with an OBDD that represents f and all decision nodes in this OBDD form a binary tree. What is the size of this OBDD? Consider the leaf decision nodes, that is decision nodes with both outgoing edges going to terminals. Obviously, for every leaf node these two edges necessarily go to different terminals (otherwise the node would have been eliminated) and so the path from the root to any leaf node encodes a prefix of some switch of f. Moreover, by the definition of a leaf node, no two leaf nodes can encode a prefix of the same switch, so the number of leaf nodes is upper bounded by the number of switches. It follows that the number of decision nodes in the constructed OBDD is at most n times the number of leaf nodes, which is at most n times the number of switches, which is exactly the size of the input switch-list representation (each switch is a vector of length n). Therefore the constructed OBDD has a linear size with respect to the size of the input SL representation.

The above considerations suffice for a proof of an existence of a linear size OBDD equivalent to the input SL representation. However, if we want to obtain also a polynomial time compilation procedure that constructs the output OBDD, we have to avoid building the exponentially large decision tree to start with. This can be easily avoided by building the output OBDD from top to bottom rather than bottom-up. We start by creating both terminal nodes, the root node, assigning the interval $[0, 2^n - 1]$ and variable x_1 to the root node, and inserting the root node into a queue. Then we start processing the nodes from the queue in the following manner. Extract the first node v with an assigned interval $[a, b]$ and a variable x_k from the queue. Scan the input switch-list until one of the following two situations occurs:

1. (Non-constatnt interval) Switch x in the switch-list is found, such that, if interpreted as a binary number, $a < x \leq b$ holds. In this case construct two children nodes v_L, v_R of v, assign variable x_{k+1} to both v_L and v_R, and assign interval $[a, (a + b + 1)/2 - 1]$ (the left half of $[a, b]$) to v_L and interval $[(a + b + 1)/2, b]$ (the right half of $[a, b]$) to v_R (a is always even, b is always odd, and the length of $[a, b]$ is always a power of 2, so there are no issues with rounding). Insert v_L and v_R to the end of the queue.

2. (Constatnt interval) Two consecutive switches x, y in the switch-list are found, such that, if interpreted as binary numbers, $x \leq a$ and $b < y$ hold. This means that there is no switch in the interval $[a + 1, b]$ and hence all vectors in the interval $[a, b]$ share the function value of x. So node v can be deleted from the tree of decision nodes and replaced by an edge from the parent node of v to the terminal $f(x)$.

The procedure stops when the queue is empty and constructs exactly the same OBDD as the bottom-up procedure described in the previous paragraph. The work per decision node is linear in the size of the input switch list (we scan the switch-list once per node), so the overall complexity of the compilation procedure is at most quadratic in the size of the input switch list[2].

3.3 CNF < SL and DNF < SL

Let us start by proving **DNF ≤ SL**. It was shown in [13] that any 1-interval function (i.e. any function with at most two switches) on n variables can be represented by a DNF with at most $2n - 4$ terms. Obviously, the output DNF has a size which is at most quadratic in n and hence also at most quadratic in the size of the input SL representation (two vectors of length n). Now assume we have SL representation of function f with k switches on the input, which means that f has $\lfloor k/2 \rfloor$ or $\lfloor k/2 \rfloor + 1$ truepoint intervals (depending on the parity of k and the function value f(0)). Let us construct a DNF representation for each interval using the algorithm from [13]. Obviously, the disjunction of these DNFs represents f and the size of this aggregated DNF is $O(kn^2)$ while the size of the input is $O(kn)$.

Now let us show **CNF ≤ SL**. Starting with SL representation of a function f with k switches we can turn it in a constant time into SL representation of $\neg f$ by negating the value of $f(0)$ and keeping the switch-list unchanged. Both SL representations have size $O(kn)$. Using the construction from the previous paragraph, we get a DNF of $\neg f$ of size $O(kn^2)$. This DNF can be switched in $O(kn^2)$ time into a CNF of the same size which represents f (by mechanically applying de Morgan rules

[2] In fact, since the tree is built in a BFS manner level by level, the procedure can be modified to restart the scan of the switch list from the beginning only once per level, improving the complexity upper bound to n times the size of the input switch list. Using a smarter data structures which for each decision node define not only the relevant interval of binary numbers but also the relevant interval in the switch-list, the overall complexity can be brought further down to linear time complexity by eliminating the factor n.

to propagatie the negation on the outside of the DNF formula towards the literals, which changes the DNF into a CNF). This finishes the construction.

In order to prove **SL** $\not\leq$ **DNF** we shall assume by contradiction that **SL** $\leq$ **DNF**. Note, that by the transitivity of succinctness relations and the above proved relation **CNF** $\leq$ **SL** we obtain from the assumption that **CNF** $\leq$ **DNF**, which is known not to be true (see e.g. [7] for a counterexample). The proof of **SL** $\not\leq$ **CNF** proceeds in a completely similar way, only the languages **CNF** and **DNF** exchange their roles.

3.4 SL$_<$ < MODS and SL$_<$ < ¬MODS

Consider the function $f(x_1,\dots,x_n,y) = y \vee (\bigwedge_{i=1}^{n} x_1,\dots,x_n)$ from Section 3.1. It depends on all $n+1$ variables, has a single switch with respect to order $y < x_1 < \ldots < x_n$ of the variables, and has $2^{n+1}+1$ truepoints (models) and $2^{n+1}-1$ falsepoints (non-models). This proves **MODS** $\not\leq$ **SL** and $\neg$**MODS** $\not\leq$ **SL**.

The relations **SL**$_<$ $\leq$ **MODS** and **SL**$_<$ $\leq$ $\neg$**MODS** are more or less obvious. Indeed, there can be at most twice as many switches as truepoints (models) and symmetrically also at most twice as many switches as falsepoints (non-models) as any switch can be identified with a pair of consecutive vectors with opposite function values.

3.5 SL is incomparable with both PI and IP

First we prove **SL** $\not\leq$ **IP** and **SL** $\not\leq$ **PI**. Let us proceed by contradiction assuming **SL** $\leq$ **IP** (**SL** $\leq$ **PI** respectively). This assumption together with the relation **OBDD** $\leq$ **SL** proved in Section 3.2 implies **OBDD** $\leq$ **IP** (**OBDD** $\leq$ **PI** respectively). However, both these relations are known to be false, see e.g. [7] for counterexamples.

Now we shall show **PI** $\not\leq$ **SL**. Let us consider function f on $2n$ variables $x_1,...,x_n,y_1,...,y_n$ defined as follows. The truepoints (models) of f are the vectors $\{v_i | i = 1,...,n\}$ where v_i assigns only variables x_i and y_i to 1, and all other variables to 0. Thus f has exactly n models, and the size of its SL representation is $O(n^2)$ ($2n$ switches each of size $2n$). Now for an arbitrary subset of indices $S \subseteq 1,...,n$, let us define a clause $C_S = (\bigvee_{i\in S} x_i \vee \bigvee_{i\notin S} y_i)$. We shall show that for every S, C_S is a prime implicate of f, and therefore f has at least 2^n prime implicates, showing the claim.

Take an arbitrary model v_i of f which by definition satisfies both x_i and y_i. No matter how S was selected, either x_i or y_i appears in C_S satisfying it. Thus C_S is an implicate of f. Now take an arbitrary proper subclause C of C_S. By the definition of C there is an index i such that neither x_i nor y_i appear in C. That means that the model v_i of f falsifies C, which implies that C is not an implicate of f. Thus C_S is prime.

Note, that the above constructed function f also shows **PI** $\not\leq$ **MODS** answering the long standing open problem from [7] (stated as a question mark in Table 3 on page 237).

It remains to show **IP** $\not\leq$ **SL**. This relation is more or less a consequence of **PI** $\not\leq$ **SL** due to the duality between CNFs and prime implicates on one hand and DNFs and prime implicants on the other hand. Consider the negation of the function from the previous proof. Obviously, SL representation of $\neg f$ has exactly the same size as the SL representation of f. Vectors $\{v_i | i = 1,...,n\}$ are now the only falsepoints (non-models) of $\neg f$ and for every $S \subseteq 1,...,n$, we can show that the term $T_S = (\bigwedge_{i\in S} \neg x_i \wedge \bigwedge_{i\notin S} \neg y_i)$ is a prime implicant of $\neg f$. So $\neg f$ has at least 2^n prime implicants, showing the claim.

Take an arbitrary non-model v_i of $\neg f$ which by definition falsifies both $\neg x_i$ and $\neg y_i$. No matter how S was selected, either $\neg x_i$ or $\neg y_i$ appears in T_S falsifying it. Thus T_S is an implicant of $\neg f$. Now take an arbitrary proper subterm T of T_S. By the definition of T there is an index i such that neither $\neg x_i$ nor $\neg y_i$ appear in T. That means that the non-model v_i of $\neg f$ satisfies T, which implies that T is not an implicant of $\neg f$. Thus T_S is prime.

4 Conclusions

The main aim of this paper is to include the languages **SL**$_<$ and **SL** into the Knowledge Compilation Map [7] with respect to their relative succinctness compared to the languages already considered there. This goal is achieved with only one open problem remaining, namely the relation between the languages $\mathbf{OBDD}_<$ and $\mathbf{SL}$. We conjecture that $\mathbf{OBDD}_< < \mathbf{SL}$. Clearly $\mathbf{SL} \not\leq \mathbf{OBDD}_<$ as the same proof using the parity function from Section 3.2 works also in this case. However, $\mathbf{OBDD}_< \leq \mathbf{SL}$ requires a (probably non-trivial) proof (if our conjecture is correct).

Acknowledgements

The first author gratefully acknowledges a support by the Ministry of Education of the Czech Republic (project 7AMB17FR027), the second author gratefully acknowledges a support by Charles University project UNCE/SCI/004 and by SVV project number 260 453.

References

[1] G. Ausiello, A. D'Atri, and D. Sacca, *Minimal representation of directed hypergraphs*, SIAM Journal on Computing (1986), 418–431.

[2] Endre Boros, Ondřej Čepek, Alexander Kogan, and Petr Kučera, *A subclass of Horn CNFs optimally compressible in polynomial time*, Annals of Mathematics and Artificial Intelligence **57** (2009), 249–291.

[3] Endre Boros, Ondřej Čepek, and Petr Kučera, *A decomposition method for CNF minimality proofs*, Theoretical Computer Science **510** (2013), 111–126.

[4] Endre Boros, Ondřej Čepek, and Alexander Kogan, *Horn minimization by iterative decomposition*, Annals of Mathematics and Artificial Intelligence **23** (1998), 321–343.

[5] Ondrej Cepek and Radek Husek, *Recognition of tractable dnfs representable by a constant number of intervals*, Discrete Optimization **23** (2017), 1–19.

[6] Ondřej Čepek, David Kronus, and Petr Kučera, *Recognition of interval Boolean functions*, Annals of Mathematics and Artificial Intelligence **52** (2008), no. 1, 1–24.

[7] Adnan Darwiche and Pierre Marquis, *A knowledge compilation map*, Journal Of Artificial Intelligence Research **17** (2002), 229–264.

[8] Peter L. Hammer and Alexander Kogan, *Optimal compression of propositional Horn knowledge bases: Complexity and approximation*, Artificial Intelligence **64** (1993), 131–145.

[9] Peter L. Hammer and Alexander Kogan, *Quasi-acyclic propositional Horn knowledge bases: Optimal compression*, IEEE Transactions on Knowledge and Data Engineering **7** (1995), no. 5, 751–762.

[10] C.Y. Huang and K.T. Cheng, *Solving constraint satisfiability problem for automatic generation of design verification vectors*, Proceedings of the IEEE International High Level Design Validation and Test Workshop, 1999.

[11] David Kronus and Ondřej Čepek, *Recognition of positive 2-interval Boolean functions*, Proceedings of 11th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty, 2008, pp. 115–122.

[12] Daniel Lewin, Laurent Fournier, Moshe Levinger, Evgeny Roytman, and Gil Shurek, *Constraint satisfaction for test program generation*, IEEE 14th Phoenix Conference on Computers and Communications, 1995, pp. 45–48.

[13] Baruch Schieber, Daniel Geist, and Ayal Zaks, *Computing the minimum DNF representation of boolean functions defined by intervals*, Discrete Applied Mathematics **149** (2005), 154–173.

[14] Christopher Umans, *The minimum equivalent DNF problem and shortest implicants*, J. Comput. Syst. Sci. **63** (2001), no. 4, 597–611.

[15] Christopher Umans, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli, *Complexity of two-level logic minimization*, IEEE Trans. on CAD of Integrated Circuits and Systems **25** (2006), no. 7, 1230–1246.

On Computing Quasi-Pareto Solution Set of Bi-objective Networks with Genetic Algorithm

Natsumi TAKAHASHI[1] Shao-Chin SUNG[2] Tomoaki AKIBA[3]
and Hisashi YAMAMOTO[4]

[1,2] *Aoyama Gakuin University*
5-10-1 Fuchinobe, Chuo-ku, Sagamihara, Kanagawa 252-5258
Japan
[1] `takahashi@ise.aoyama.ac.jp`, [2] `son@ise.aoyama.ac.jp`

[3] *Chiba Institute of Technology*
2-17-1 Tsudanuma, Narashino, Chiba, 275-0016
Japan
`tomo@akiaki.net`

[3] *Tokyo Metropolitan University*
6-6 Asahigaoka, Hino, Tokyo 191-0065
Japan
`yamamoto@tmu.ac.jp`

Abstract

Many network systems have been applied extensively in the real world, for example, Internet, power network and traffic network. For these network systems, high reliability is required. Independently, costs may occur for the construction or maintenance of edges (links) in the networks. In this study, we consider a bi-objective network design problem with two objectives of maximizing all-terminal reliability and minimizing construction/maintenance costs of edges. In general, there is the trade-off relation between these objectives, and it is difficult to optimize these objectives simultaneously. Therefore, the meaning of solving the problem with bi-objectives is to find the set of all Pareto solutions.

On the other hand, the problem of evaluating all-terminal reliability of a given network is computationally intractable, which implies that our bi-objective network design problem is computationally intractable as well. Hence, it is reasonable to switch our goal to find a set of quasi-Pareto solutions, i.e., a set of non-dominated "good" solutions in terms of our two objectives. Our goal is achieved by developing an algorithm based on genetic algorithm, which produce a set of quasi-Pareto solutions. In order to generate quasi-Pareto solutions which are close to Pareto solutions in the solution space, we focus on distribution properties of Pareto solutions and topological characteristics of networks. In the GA procedure, these properties of Pareto solutions are adopted to selection operation and crossover operation. Moreover, we conduct numerical experiments, and the effectiveness of our proposed algorithm is evaluated in terms of the robustness of the quasi-Pareto solutions.

Hardness of Existence of Individually Rational Partitions for Additive Hedonic Games

Masaya Tanaka[1] and Shao-Chin Sung[2]

Aoyama Gakuin University
5-10-1 Fuchinobe, Chuo, Sagamihara, Kanagawa
252-5258, Japan
c5617157@aoyama.jp[1], son@ise.aoyama.ac.jp[2]

Abstract

We are concerned with the existence of stable partitions for additive hedonic games in the situation that individually rational (IR) is imposed. IR requires that each player weakly prefers her coalition to being alone, i.e., IR is very simpler concept. For some important stability concepts, including Nash stability, individually stability, core stability, strictly core stability, each of these stability concepts implies IR. In general, for each instance, the partition consisting of all singletons of each player (singletons' partition) satisfies IR, i.e., IR partition always exists. However, instances for which singletons' partition is Nash stable are trivial instances when the existence problems w.r.t. Nash stability is under consideration. In our study, we show that checking the existence of IR partitions without singletons' partitions for additive hedonic games is NP-complete. Hence, the essential of the hardness that checking the existence of Nash stable partitions is verifying the existence of IR partitions. On the other hands, each of contractually stability concepts, including contractually Nash stability, (weak) contractually individual stability, contractually core stability, contractually strict core stability, does not imply IR. Without CN stability, it is known that the existences of corresponding stable partitions which satisfy IR are guaranteed. In our study, we show that checking the existence of CN stable and IR partitions for additive hedonic games is NP-complete.

Keywords: Hedonic Games, Additive Preferences, Individually Rational

Analysis of firing element positioning in air defense simulation

Dalibor Cimr[1], Richard Cimler[2], Hana Tomaskova[1]

[1] *University of Hradec Kralove, Faculty of Informatics and Management*
Rokitanskho 62, Hradec Kralove
Czech Republic dalibor.cimr@uhk.cz

[2] *University of Hradec Kralove, Faculty of Science*
Rokitanskho 62, Hradec Kralove
Czech Republic richard.cimler@uhk.cz

Abstract

Introduction A computer simulation of effective positioning of firing elements is presented in this paper. Agent based modeling is used for the creation of an expert tool which can be used for decision making during the process of the effective positioning of resources.

Aim The aim of this research is to create an expert tool which, based on the information about the environment from GIS data, the available resources in the form of anti-air elements, and expected air-strike routes, find suitable positions for the anti-air elements on the given map.

Methods Agent based simulation has been used for the creation of the software. Each anti-air element, defended object, and aircraft, is modeled as an autonomous agent. A line of sight algorithm, a flight height algorithm, and the secant method, is used during the computation of the firing of the anti-air elements.

Results Simulation enables testing different strategies for the positioning of anti-air elements. This tool can be used to load the data about a given environment, set the available resources and predicted air strike routes, and then, based on the simulation, determine the optimal positioning of the given resources. The whole process is presented in an experiment.

Keywords: ABM, Air defense, Simulation, Agent-based modeling, Military simulation

1 Introduction

Choosing an optimal strategy in a limited time is a typical problem in real life situations. Experts use their experience for the decision making and try to choose the best solution from a given set of possible solutions. [1, 2, 3, 4] Such a process can be very difficult if the set of all possible solutions is large. In many situations it is not possible to think about all the connections and dependencies in a given system. The width of the range of solutions to be considered depends on the knowledge and experience of the expert. In some situations, it is necessary to make a decision in a very short time, and it is not possible to find an optimal solution but only to find one which is good enough for dealing with the situation.

Computer models enable simulating different scenarios and evolutions of a system by testing many or all possible combinations of parameters which affect the behavior of a system in a very short time. Repeating the same experiment helps to investigate the effect of randomness on the system. One type of computer modeling is agent-based modeling (ABM), where each entity of the modeled system is an autonomous agent. In such a model it is possible to study emerging effects, i.e., how the simple behavior of one agent on the micro-level can affect the behavior of the whole system on the macro-level.

This paper presents an expert system which helps in the decision making process of the placement of anti-air elements. In cooperation with experts from the military, there has been created a model of an air-strike on defended objects. The defended objects are located in an environment which is loaded from GIS data. Every place on Earth can be used in the simulation if the GIS data about the terrain are available. The possible sizes and directions of an attack are set by the user during the setup of the simulation. The last step is to choose the available resources which can be placed in the area. Different parameters about the air-strike and the characteristics of the anti-air elements can be set or loaded from a file.

The simulation tries different positions for the anti-aircraft elements, based on the selected strategy. After several simulation runs, the most successful placements are selected and presented to the user.

This paper is structured as follows: The problem definition as well as a review of the related literature is in Section 2. The model description is in Section 3. A case study and an experiment can be found in Section 4. The conclusions are presented in Section 5.

2 Problem definition and related work

The problem studied is from the field of military decision making. The aim of the tool created is to support the military commander during the process of deciding where to place anti-aircraft elements. This tool should help determine the optimal placement in a given situation. There are different parameters which have to be taken into account, such as the speed, the weapon load, and the position of the aircraft. For the anti-aircraft elements, one must observe their locations, ranges, and type of weapon with their loading times and accuracy. That is why an agent-based approach has been chosen for the model, where each entity is modeled as an autonomous agent.

Agent-based simulation can be used for modeling various systems, including military systems. There are models of command and control such as [5] to determine the most critical tasks at the company level. Military applications of agent-based simulations can be found in [6]. In the research described in this paper, agent-based simulation has been used for a decision support system for the military commander.

Decision support systems are created together with experts and enable using their knowledge in a given problem. Such a system can be used without direct communication with an expert. These systems can be used also in the military to help the commander's decisions [7, 8].

3 Model description

The design of the model logic is based on the information obtained from consulting military experts. The purpose of the model is to analyze the possibilities for the effective positioning of firing elements in a given geographic environment and to analyze the different flight scenarios of the aircraft. The aim is to get a better distribution of defense resources, which means comparing the added values of a new firing element with its cost in service. In a crisis situation, thanks to the knowledge gained from the model, faster and better decision making can occur.

The model simulates a direct attack of airplanes on a defended object in a selected geographic environment that can be created from GIS data. The environment is created from altitude data where the terrain contains inequalities affecting other agents. The environment has a major impact on the ability to fire on enemy elements. The firing elements, aircraft, and defended objects can be placed in the environment. The firing elements are immobile during the simulation and have full information about the positions of all aircraft. This is due to continuity with previous work of the authors [9] dealing with the placement of a radar system for the detection of aircraft.

3.1 Line of sight algorithm

In this model, the influence of the terrain is included in the simulations. The firing elements and aircraft can shoot at their target, but only if there is nothing in the way along the trajectory of the missiles. The line of sight algorithm [10] can calculate whether all elevation values between the shooting object and the target are lower than the values obtained from a linear function of the objects' coordinates. A similar method was used in the authors' previous work [9], where linear interpolation was not used to determine the approximate height between two known points, but only a nearest point for a visibility algorithm.

3.2 Secant method

Unlike aircraft, firing elements shoot their missiles at a moving target. Therefore it is important to find the right moment for shooting, taking into account the different flight times of the missile and the aircraft.

The secant method is used for this. It is shown in Fig. 1, where there is a red airplane (A) with its direct trajectory and a blue firing element (FE) with its firing range. The green points represent the places where the missile can hit a target if the previous missile failed (points 1–6). The resulting gaps between these points are due to the charging time and the time required to reach the striking point. The distance of the aircraft from the first point of a strike is denoted by d_a, and that of the missile by d_m. The speed of the aircraft is v_a and that of the misisle is v_m, while t represents the time of the motion. In this case, $v_a < v_m$, which means that $d_a < d_m$. The number of iterations is set to a maximal value of 40 or until the accuracy of the result is satisfactory.

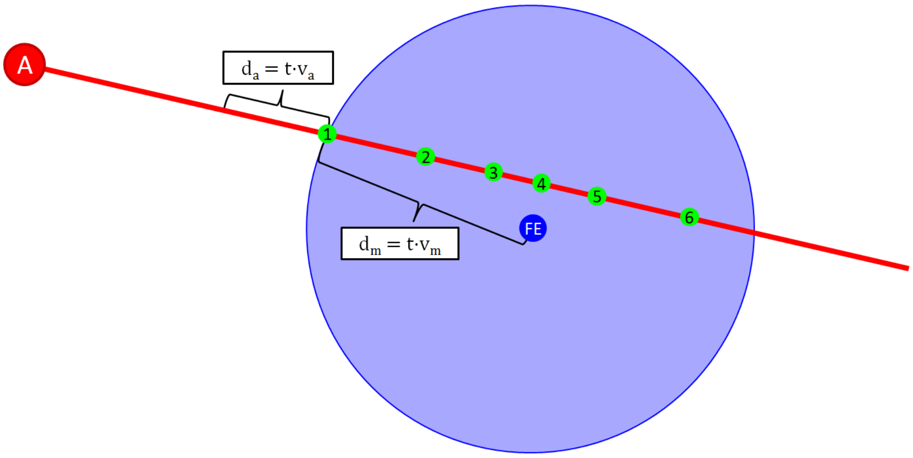


Figure 1: Firing situation (top view)

3.3 Flight height trajectory algorithm

This method is used to compute the aircraft movement over the terrain. The firing element and the defended object have a static position in the environment. The model simulates a direct flight, so the airplane performs a movement that changes only the height of the flight as they fly over the terrain. This represents a possible attack on the defended object where the goal of the attack is a quick destruction of the object of interest. Due to the characteristics of this movement, the route is determined as the agent is initialized and is changed when the defended object is destroyed and it is possible to destroy another one.

During the model initialization, the airplane's trajectory is created in several phases. In the first phase, the points at a constant altitude above the terrain in the direction of flight are obtained as the linear intersections of two neighboring elevation points. If the trajectory passes directly by the altitude point, the height above this point is used. In the next phase, adjustments are made to keep the trajectory smooth. This models the impossibility of the aircraft's rapidly changing its flight elevation. First, three consecutive points are selected. If the middle point is below the two border points, it is raised to the average of the heights of these points. If the midpoint is above, the extreme points are raised only if the airplane ascends or falls below at too high an angle, which is impossible in a real situation.

The other adjustment is to scroll through pairs. Only the second points that are lower than the first point are raised by more than the maximum height difference. Then this is performed again in the reverse order, where the points begin to be tested from the last point to the first. This has less asymptotic complexity than using this approach in one direction with backtracking. Fig. 2 shows the optimization progress of the algorithm. The first three pictures describe each step of progression and the last one is the result after 5 repetitions. The number of iterations depends on the previous result and the size of the differences from the new one.

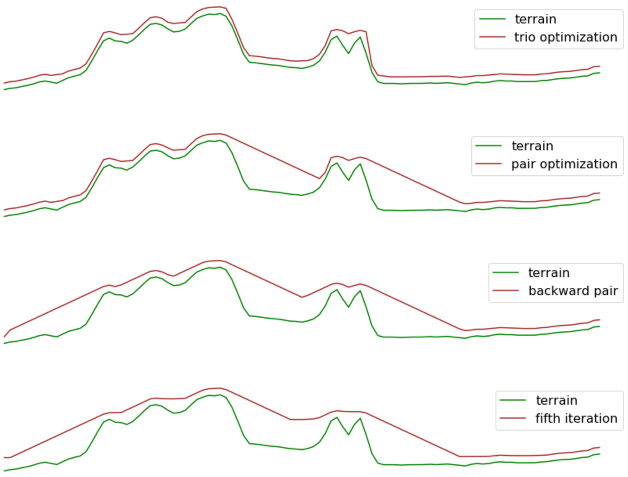


Figure 2: Flight height trajectory algorithm (side view)

3.4 Firing algorithm

Three firing algorithms determining the priority of a target selection have been created for defensive firing elements. The first one selects priority targets that will reach the defended object in the shortest possible time. Logic of this algorithm is simple – its goal is to destroy nearest threat to the defended object. It is not taking into account the number of other possible threats. During this strategy missiles are fired in the shortest possible given by reloading speed. Target is changed only if the aircraft is destroyed or any other aircraft gets closer to the defended object.

The second algorithm considers number of missiles which have been shot and are currently approaching to the target. Primary the aircraft which is closest to the defended object is selected and missile is fired. After the reloading time, second nearest airplane is selected as a target. If there is at least one rocket approaching to each aircraft, selection of targets starts from beginning – nearest plane is selected and a second rocket is fired. This round-style process continues until any of missiles miss its target – in such a case, missile is fired on the airplane on which fewer number of rocket are targeted.

These two algorithms have the common feature – whenever there is at least one target that can be shot (the airplane is not hidden behind the terrain), it launches the missile. This may cause more than one missile at a time to fly towards the airplane. The last algorithm is the modification of the second algorithm which selects as a target aircraft which is the nearest to the defended object and none or less rockets are approaching towards it. It shoots only one missile on each airplane. This simulates the scenario where saving resources (ammunition) is a priority. If the target is hit, no further rockets are needed and in another case, a new missile is fired.

3.5 Agents' parameters

There are three main types of agents: 1) Aircraft, 2) Anti-aircraft elements, 3) Defended objects. The most important parameters of these entities are:

- **Aircraft:** speed, firing distance, reload time, current ammo amount, missile speed, missile hit percentage.
- **Anti-aircraft element:** fire radius, reload time, current ammo amount, missile speed, missile hit percentage.
- **Defended object** has only one parameter, containing the information of whether it has been destroyed or not.

3.6 Firing elements positioning

User, mostly military commander, chooses the possible areas where can be firing elements placed. It might be a whole map or just selected areas. These areas are sliced into given number of squares. The highest place in the given square is chosen as an optimal place for placement of an anti-aircraft element. Number and parameters of available resources – anti-aircraft elements – are selected. Based on the simulation results, the optimal placement of these resources are showed. The optimal placement means selected squares which should be used for a placement of an anti-aircraft elements. Results are given in tables as well as graphical interpretations. The percentage of shot-down aircraft and the success rate of defense is displayed.

4 Case study

The algorithm has been tested in a situation where the location of the protected object and firing element has been selected by the expert. The testing area has been selected to demonstrate an environment with mountains and lowlands, and is represented by an altitude map containing 250x250 points where each side of the area has length 50 km. The defended object and the firing element are located on the surface of the terrain; the airplanes are located above the terrain at a flying height level.

4.1 Experiment

The purpose of this experiment is to find information about the suitability of various positionings of the firing elements. There is an option to choose between several categories of aircraft, with different optional values of parameters like speed, flying height, etc. In the same way, there are several categories of the firing elements. Based on the selected number of firing elements, variation without repetition is used to assemble all options of positioning. For each combination of positions, the simulation is triggered multiple times, depending on the selected iteration value.

4.2 Results

The results of the experiment are illustrated in Fig. 3, which shows the success rate of defending the objects and of aircraft shot down. It is clear that the most successful position, number 4, does not have the most aircraft shot down: rather, with this positioning, the object of interest has been most often defended. The firing element located at position 15 destroyed the most airplanes but failed to protect the defended object. This means that the element destroyed the aircraft too late, at a time when the airplane had managed to fire at the defended object and destroy it. This can be seen on all four positionings that are nearest to the defended object. From position 13, the firing element could not shoot down any airplanes, and situations where the defended object was not destroyed occurred only when all missiles missed the target.

Fig. 4(a) shows the best positions in each segment. In this segment, the highest height value is chosen to place a firing element. Fig. 4(b) corresponds with the best positioning after the experimentation.

5 Conclusion

A tool for decision making support has been introduced in this paper. The presented software can be used during the decision making process about the positioning of firing elements in a given environment. Based on the results of simulations, the most suitable places for anti-aircraft firing elements are chosen. The final decision is still with the commander, but the results of this computational process can significantly help with making the decision.

Any place on Earth can be loaded into the simulation by uploading GIS data about the terrain height. Different sizes and directions of attacks can be tested and evaluated. The tool can be used by military commanders as a decision support system for testing different scenarios and finding the optimal positioning of anti-aircraft elements.

The model can be further extended, for example, simulation with more accurate characterization of missiles whose speed will not have a linear course over time. Additionally, the possibility

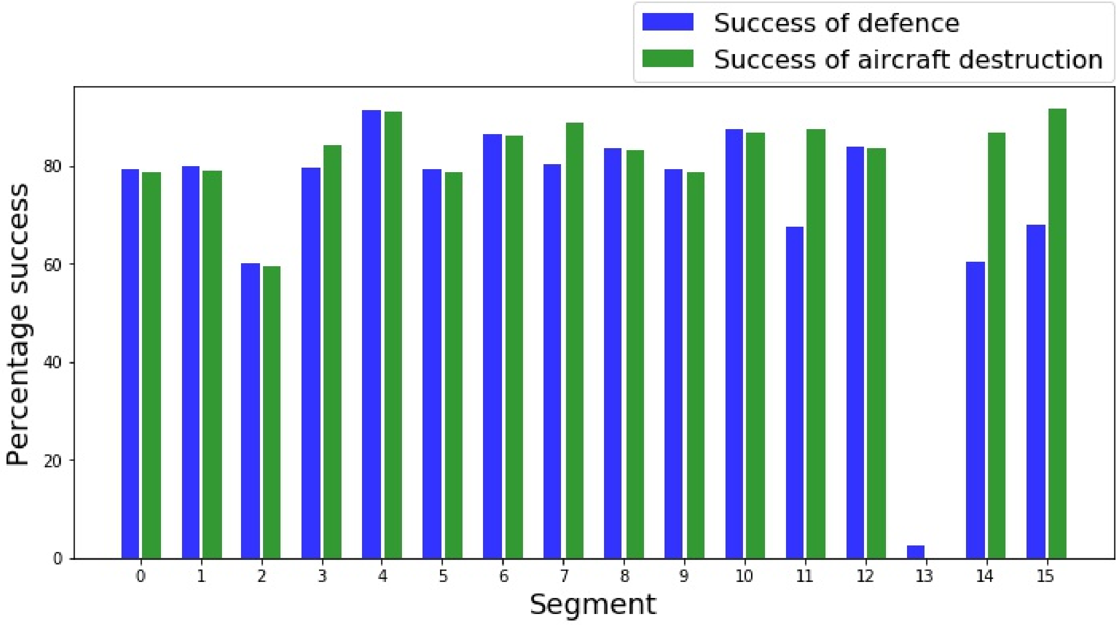
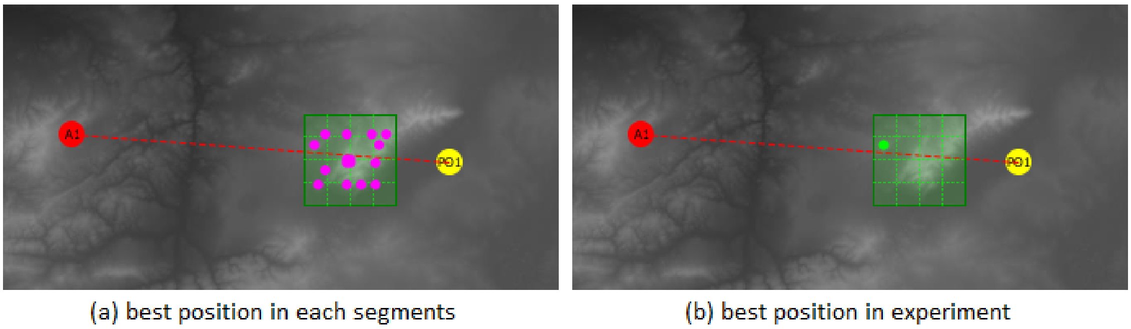


Figure 3: The success of defending, depending on the position of the anti-air element



(a) best position in each segments

(b) best position in experiment

Figure 4: Placement of firing element

of attacks may be extended, where airplanes for the elimination of the firing elements may be present. Moreover, it is envisaged to create a model combining the knowledge gained from this model with a model containing elements detecting the airplane. There might be a direct cooperation and communication between aircrafts, or a central command center may be created to mediate communication between these individual elements.

Acknowledgment

The support of the Specific Research Project at FIM UHK is gratefully acknowledged. Special thanks to Lukas Laska for participation in the research.

References

[1] M. Gavalec, J. Ramík, and K. Zimmermann, "Decision making and optimization," *Lecture Notes in Economics and Mathematical Systems.*, vol. 667, 2015.

[2] R. Cimler, M. Gavalec, K. Mls, and D. Ponce, "Decision support biomedical application based on consistent optimization of preference matrices," in *International Conference on Computational Collective Intelligence.* Springer, 2016, pp. 292–302.

[3] R. Cimler, K. Mls, and M. Gavalec, "Decision support smartphone application based on interval ahp method," in *Computational Collective Intelligence.* Springer, 2015, pp. 306–315.

[4] M. Gavalec, H. Tomášková, and R. Cimler, "Computer support in building-up a consistent preference matrix," in *Advanced Computer and Communication Engineering Technology.* Springer, 2016, pp. 947–956.

[5] W.-S. Yun, I.-C. Moon, and T.-E. Lee, "Agent-based simulation of time to decide: Military commands and time delays," *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 4, p. 10, 2015.

[6] T. M. Cioppa, T. W. Lucas, and S. M. Sanchez, "Military applications of agent-based simulations," in *Proceedings of the 36th Conference on Winter simulation.* Winter Simulation Conference, 2004, pp. 171–180.

[7] S.-h. Liao, "Case-based decision support system: Architecture for simulating military command and control," *European Journal of Operational Research*, vol. 123, no. 3, pp. 558–567, 2000.

[8] T. P. Hanratty, E. A. Newcomb, R. J. Hammell II, J. T. Richardson, and M. R. Mittrick, "A fuzzy-based approach to support decision making in complex military environments," *International Journal of Intelligent Information Technologies (IJIIT)*, vol. 12, no. 1, pp. 1–30, 2016.

[9] D. Cimr, R. Cimler, and H. Tomášková, "Combination of collision detection and visibility algorithms in simulation of the effective placement of anti-air elements," in *Advanced Topics in Intelligent Information and Database Systems.* Springer, 2017, pp. 213–221.

[10] R. Seixas, M. Mediano, and M. Gattass, "Efficient line-of-sight algorithms for real terrain data," *III Simpósio de Pesquisa Operacional e IV Simpósio de Logística da Marinha–SPOLM 1999*, 1999.

Minimal Negotiation Sets for Assignment Games on Complete Graphs

Yoshifumi KUSUNOKI and Masaki SAITO

Graduate School of Engineering, Osaka University
2-1, Yamada-oka, Suita, Osaka
Japan
kusunoki@eei.eng.osaka-u.ac.jp

1 Introduction

An assignment problem is a situation where buyers and sellers in a market aim to make profits by trading with each other. Assignment problems have been studied in terms of cooperative game theory [1, 2], in order to prescribe players' payoffs when their cooperation is assumed and the profit of the whole market is maximized. Cooperative games derived from assignment problems are called assignment games.

Consider the situation where a doubles tennis tournament will be held. Suppose that we can (probabilistically) estimate how much the prize money is earned by each pair of players. Then, we have the question that with whom each player will pair for the doubles tournament. Also, how will each player and his/her partner distribute the prize money they got? Since there is no distinction of two player groups such as buyers and sellers, this problem can not be dealt with as assignment games, which is considered on bipartite graphs. Hence, in this paper, we consider assignment games on complete graphs. Assignment games on complete graphs are closely related to matching games [3] and one-sided assignment problems [4].

Unlike the conventional assignment games, those on complete graphs can have empty core. In order to provide rational solutions for the assignment games with empty core, we propose generalizations of the core solution by applying minimum negotiation sets, proposed by Gonzalez and Grabisch [5], to the assignment games. Moreover, we investigate on properties of the generalized core solutions.

2 Preliminaries

2.1 Cooperative Games and Minimum Negotiation Sets

A transferable utility game (TU game) is defined by a pair (N, v), where the set $N = \{1, 2, \ldots, n\}$ is a finite player set and $v\colon 2^N \to \mathbb{R}$ with $v(\emptyset) = 0$ is a characteristic function. Every nonempty subset $S \subseteq N$ is called a coalition, and $v(S)$ means the value that S gains autonomously. N is called the grand coalition. We will denote by $\mathcal{G}^N$ the set of all games over N.

Cooperative game theory has been studied on how to divide the value of the grand coalition among all players. Such an allocation method is called a solution. We say $x = (x_i)_{i \in N} \in \mathbb{R}^N$ a payoff vector, where each x_i is a payoff of player i. Solutions which give one payoff vector for each game are called one-point, otherwise those which give a set of payoff vectors are called set-valued.

The core is one of the most popular set-valued solutions, which is defined by

$$C(v) = \left\{x \in \mathbb{R}^N \mid x(S) \geqslant v(S) \text{ for } \forall S \subseteq N,\ x(N) = v(N)\right\},$$

where $x(S)$ is the aggregated payoff of a nonempty set S, that is $x(S) = \sum_{i \in S} x_i$. For the empty set, we define $x(\emptyset) = 0$.

2.2 k-additive Core and Minimum Negotiation sets

To discuss rational solutions for games with the empty core, Miranda and Grabisch [6] have proposed a generalization of the core called k-additive core. They call vectors on the power set of N generalized payoff vectors. The k-additive core assigns generalized payoff vectors to each game.

Definition 1. [6] Given $v \in \mathcal{G}^N$. The k-additive core of v, denoted by $\mathbf{C}^k(v)$, is given by

$$\mathbf{C}^k(v) = \left\{ x \in \mathbb{R}^{2^N \setminus \{\varnothing\}} \;\middle|\; \begin{array}{l} \sum_{T \subseteq S} x_T \geqslant v(S) \text{ for } \forall S \subseteq N,\ \sum_{S \subseteq N} x_S = v(N) \\ x_S = 0 \text{ for } \forall S \subseteq N,\ |S| > k \end{array} \right\}.$$

Not that x is a 2^{n-1}-dimensional vector. For any $S \subseteq N$ such that $|S| > k$, the element x_S of a generalized payoff in the k-additive core is vanished. Every payoff vector in the k-additive core satisfies the same rationality of the core. Unlike the core, the k-additive core is always nonempty [6].

When building a solution from the k-additive core, there could be a conflict between the agents of a coalition S on how the payment of x_S made by the k-additive core will be allocated to each player. The more there are coalitions S with high amounts $|x_S|$, the more the solution x involves negotiation among the agents in S. Gonzalez and Grabisch [5] have introduced minimum negotiation sets to select those solutions in the k-additive core which involve the "fewest" negotiation. The minimum negotiation set is defined as follows.

Let $2 \leqslant k \leqslant n$ be fixed and $\| \cdot \|$ be a symetric norm on $\mathbb{R}^{2^n}$. The minimum negotiation set is defined by the set of optimums of the following nonlinear program:

$$\underset{x}{\text{minimize}} \ \|(x_S)_{S \subseteq N,\ |S| \geqslant 2}\| \text{ subject to } x \in \mathbf{C}^k(v). \tag{1}$$

3 Assignment Games on Complete Graphs

In the previous work, assignment games on complete "bipartite" graphs have been studied. In this paper, we consider assignment games on complete graphs.

First, we introduce graphs and related notions. We consider a graph (V, E), where V is a vertex set and $E \subseteq \{(i,j) \mid i,j \in V, i \neq j\}$ is an edge set. Any graph appear in this paper is undirected, that is we identify edges (i,j) and (j,i). In particular, the graph that every pair of distinct vertices is connected is called complete, and (V, E_V) denotes the complete graph. A matching of a graph (V,E) in an edge set $\mu^* \subseteq E$ where the edges are disjoint. It is called perfect if it covers all the vertices. Moreover, given a nonnegative weight function $d\colon E \to \mathbb{R}$ for the edges, an assignment problem (or maximum matching problem) is to find the matching μ^* which maximizes the sum of the weights of edges in μ^*, i.e., $\mu^* = \arg\max_{\mu \in \mathcal{M}} \left\{ \sum_{e \in \mu} d(e) \right\}$, where $\mathcal{M}$ is the set of all matchings. For a weight function d, we denote an assignment problem on the complete graph (V, E_V) by (V, d).

Remember N is the player set. Consider an assignment problem (N, d). Let $\mathscr{M}(S)$ be the set of all matchings for $(S, d|_S)$, where $d|_S$ is the restriction of d on (S, E_S). Then, an assignment game (on the complete graph), denoted by (N, w_d), is defined by

$$w_d(S) = \begin{cases} 0 & |S| = 0, 1, \\ \max\left\{ \sum_{(i,j) \in \mu} d(i,j) | \mu \in \mathscr{M}(S) \right\} & \text{otherwise.} \end{cases} \tag{2}$$

The next theorem shows a simple representation of the core of (N, w_d).

Theorem 1. The core of (N, w_d) can be represented by

$$C(w_d) = \left\{ x \in \mathbb{R}^N \;\middle|\; \begin{array}{l} x_i \geqslant 0 \text{ for all } i \in N \\ x_i + x_j = d(i,j) \text{ if } (i,j) \in \mu^* \\ x_i + x_j \geqslant d(i,j) \text{ if } (i,j) \notin \mu^* \\ x_i = 0 \text{ if } i \text{ is not assigned by } \mu^* \end{array} \right\}, \tag{3}$$

where μ^* is any optimal matching.

Hereinafter, by adding the dummy player if necessary, we assume $|N|$ is even without loss of generality. Moreover, we suppose that any optimal matching which appears in the rest of the paper is perfect matching.

4 Minimum Negotiation Sets for Assignment Games on Complete Graphs

The core of an assignment game on the complete graph is not necessarily nonempty. We apply minimum negotiation sets to the assignment games.

We specialize the k-additive core to the assignment games. Given an assignment problem (N, d). Let μ^* be an optimal matching of (N, d). Then, $\mathbf{C}_{\mu^*}(d)$ is defined by

$$\mathbf{C}_{\mu^*}(d) = \left\{ x \in \mathbb{R}^N \times \mathbb{R}^{E_N} \;\middle|\; \begin{array}{l} x_i \geqslant 0 \text{ for all } i \in N \\ x_i + x_j + x_{(i,j)} = d(i,j) \text{ if } (i,j) \in \mu^* \\ x_i + x_j + x_{(i,j)} \geqslant d(i,j) \text{ if } (i,j) \notin \mu^* \\ x(N) = w_d(N) \end{array} \right\}. \tag{4}$$

Roughly speaking, we can say that $\mathbf{C}_{\mu^*}(d)$ is a special case of 2-additive case. The core $C(w_G)$ is included in $\mathbf{C}_{\mu^*}(d)$. The following theorem says that like the 2-additive core, $\mathbf{C}_{\mu^*}(d)$ is always nonempty.

Theorem 2. For (N, d) and an optimal matching μ^*, $\mathbf{C}_{\mu^*}(d) \neq \varnothing$.

Then, given ℓ_p norm $\|\cdot\|_p$ to be minimized, a minimum negotiation set is defined by the set of optimums of the following problem. The set is denoted by $\mathbf{I}^p_{\mu^*}(d)$

$$\underset{x}{\text{minimize}} \;\; \|(x_{(i,j)})_{i,j \in N, i \neq j}\|_p \;\; \text{subject to } \; x \in \mathbf{C}_{\mu^*}(d). \tag{5}$$

Clearly, when the core is nonempty, for any p, $\mathbf{I}^p_{\mu^*}(d)$ coincides the core.

The next proposition says that the sign of x_e is determined after it is minimized.

Proposition 1. Given an assignment problem (N, d) and an optimal matching μ^* of (N, d). For any $x \in \mathbf{I}^p_{\mu^*}(d)$, the following statements hold:

(i) $x_e \leqslant 0$ for all $e \in \mu^*$.

(ii) $x_e \geqslant 0$ for all $e \notin \mu^*$.

From Proposition 2, an interpretation of x_e is considered to be different whether $e \in \mu^*$ or $e \notin \mu^*$. Reflecting this fact, we consider two specializations of $\mathbf{C}_{\mu^*}(d)$ as follows.

$$\check{\mathbf{C}}_{\mu^*}(d) = \{x \in \mathbf{C}_{\mu^*}(d) \mid x_{(i,j)} = 0, \; (i,j) \notin \mu^*\}. \tag{6}$$

$$\hat{\mathbf{C}}_{\mu^*}(d) = \{x \in \mathbf{C}_{\mu^*}(d) \mid x_{(i,j)} = 0, \; (i,j) \in \mu^*\}. \tag{7}$$

$\check{\mathbf{I}}^p_{\mu^*}(d)$ and $\hat{\mathbf{I}}^p_{\mu^*}(d)$ denote the corresponding minimum negotiation sets for $\check{\mathbf{C}}_{\mu^*}(d)$ and $\hat{\mathbf{C}}_{\mu^*}(d)$, respectively. $\hat{\mathbf{I}}^p_{\mu^*}(d)$ is related to BLOCKING VALUE in [3].

For $x \in \check{\mathbf{I}}^p_{\mu^*}(d)$, $x_{(i,\mu^*(i))}$ can be interpreted as a debt owned by coalition $\{i, \mu^*(i)\}$ in order to form the grand coalition. On the other hand, for $x \in \hat{\mathbf{I}}^p_{\mu^*}(G)$, $x_{(i,j)}$ can be interpreted as a dissatisfaction of coalition $\{i, j\}$ when the grand coalition is formed.

When $p > 1$, $\mathbf{I}^p_{\mu^*}(d)$ should be a singleton.

Theorem 3. Let $p > 1$. $\mathbf{I}^p_{\mu^*}(d) = C(w_G)$ if $C(w_G) \neq \varnothing$. Otherwise, $\mathbf{I}^p_{\mu^*}(d)$ consists of a singleton.

We consider the special case that $p = 1$. $\check{\mathbf{I}}^1_{\mu^*}(d)$ is equal to the set of all optimal solutions of the following problem.

$$\begin{array}{ll} \underset{x}{\text{maximize}} & \displaystyle\sum_{e \in \mu^*} x_e \\ \text{subject to} & x_i + x_j + x_{(i,j)} = d(i,j), \quad (i,j) \in \mu^*, \\ & x_i + x_j \geqslant d(i,j), \quad (i,j) \notin \mu^*, \\ & x_i \geqslant 0, \quad i \in N, \\ & x_e \leqslant 0, \quad e \in \mu^*. \end{array} \tag{8}$$

The following propositions are obtained from this fact.

Proposition 2. Given (N, d). Let ϵ be a gap between the optimal value of the following problem

$$\begin{aligned} \text{maximize} \quad & \sum_{(i,j)\in E_N} d(i,j)f(i,j) \\ \text{subject to} \quad & \sum_{j\neq i} f(i,j) \leqslant 1, \quad i \in N, \\ & f(i,j) \geqslant 0, \quad i,j \in N, \end{aligned}$$

and the value of optimal matchings. Then ϵ is equal to the optimal value of the problem (8).

The next proposition shows a simple method of finding elements of $\check{\mathbf{I}}^1_{\mu^*}(d)$.

Proposition 3. Given (N, d). Let p^* be an optimal solution of the following problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in N} p_i \\ \text{subject to} \quad & p_i + p_j \geqslant d(i,j), \ i,j \in N, \\ & p_i \geqslant 0, \quad i \in N. \end{aligned}$$

Let $q^*_{(i,j)} = d(i,j) - p^*_i - p^*_j$ if $(i,j) \in \mu^*$, and $q^*_{(i,j)} = 0$ otherwise. Then $(p^*, q^*) \in \check{\mathbf{I}}^1_{\mu^*}(d)$.

The following proposition show a relation between $\mathbf{I}^1_{\mu^*}(d)$ and $\check{\mathbf{I}}^1_{\mu^*}(d)$.

Proposition 4. We have $\check{\mathbf{I}}^1_{\mu^*}(d) \subseteq \mathbf{I}^1_{\mu^*}(d)$.

5 Conclusion

In this paper, we have introduced the assignment games on the complete graph. We have proposed the minimum negotiation sets for the assignment games, which are generalizations of the core. We discuss some properties of the minimum negotiation sets. Future tasks include further investigation on the minimum negotiation sets and study payoff vectors derived from the minimum negotiation sets.

References

[1] L. S. Shapley and M. Shubik. The Assignment Game I: The Core. *International Journal of Game Theory*, 1(1):111–130, 1971.

[2] R. van den Brink and M. Pintér. On axiomatizations of the shapley value for assignment games. *Journal of Mathematical Economics*, 60:110–114, 2015.

[3] P. Biró, W. Kern, and D. Paulusma. Computing solutions for matching games. *International Journal of Game Theory*, 41(1):75–90, 2012.

[4] B. Klaus and A. Nichifor. Consistency in one-sided assignment problems. *Social Choice and Welfare*, 35(3):415–433, 2010.

[5] S. Gonzalez and M. Grabisch. Preserving coalitional rationality for non-balanced games. *International Journal of Game Theory*, 44(3):733–760, 2015.

[6] P. Miranda and M. Grabisch. k-balanced games and capacities. *European Journal of Operational Research*, 200(2):465–472, 2010.

From Abstract to Executable Models for Multi-Agent Path Finding on Real Robots

Roman Barták, Ivan Krasičenko, Jiří Švancara, Věra Škopková, David Nohejl

Charles University, Faculty of Mathematics and Physics
Prague, Czech Republic
bartak@ktiml.mff.cuni.cz

1 Introduction

Abstraction is the process of removing details from a problem representation. It is a critical step in problem solving as without abstraction "intelligent agents would be completely swamped by the real world" [1]. Despite its importance, little attention has been paid to abstraction techniques compared to, for example, solving techniques. In areas, such as planning, the formal abstract model has been proposed and many concrete domain models are used for benchmarking, but the studies how to obtain such models and how the models relate to real world are rare.

In this paper, we look at a specific planning problem called *multi-agent path finding* (MAPF) that deals with the problem of finding collision-free paths for a set of agents (robots). MAPF has a strong practical applicability in areas such as warehousing and intelligent road junctions. An abstract model with a graph describing the environment and agents moving between the nodes of the graph has been proposed. This model is widely accepted by the MAPF community and majority of MAPF algorithms rely on this model. Our goal is studying appropriateness of MAPF abstract models from the perspective of executing the obtained plans. We argue that the classical abstract model may not be appropriate, when the plans are to be executed on real robots. We provide some preliminary empirical evidence that abstract plans deviate from real plans executed on robots, Ozobots [2] in our study (Figure 1), and we compare several variants of abstract models. The paper motivates further research on abstraction of problems with respect to applicability of solutions in practice. This is a short version of paper [3], which gives full technical details.

2 Background on MAPF

Formally, the MAPF problem is defined by a graph $G = (V, E)$ and a set of agents $a_1, \ldots, a_k$, where each agent a_i is associated with starting location $s_i \in V$ and goal location $g_i \in V$. The time is discrete and in every time step each agent can either move from its location to a neighbouring location or wait in its current location. A grid map with a unit length of each edge is often used to represent the environment [4]. The task is to find a collision-free path for each agent, where the collision occurs when two agents are at the same node at the same time or two agents move along the same edge at the same time in opposite directions. The makespan (the maximal time when all agents reached their destinations) objective function is often studied in the literature [5]. The problem to find a makespan-optimal solution is NP-hard [6]. Though the plans obtained by different MAPF solvers might be different, the optimal plans are frequently similar and tight (no superfluous steps are used). Hence, any optimal MAPF solver can be used. We used the reduction-based solver in the Picat programming language [7].

Figure 1: Ozobot Evo from Evollve used for the experiments. Picture is taken from [2].

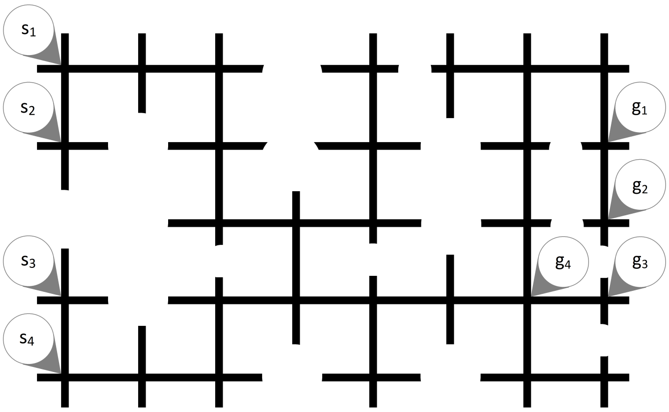


Figure 2: Instance map for Ozobots. Ozobots follow the black line, the grey circles indicate starting and goal locations (not printed on real map).

3 MAPF Models and Executable Plans

For our study we designed an environment that is intentionally close to the abstract model of MAPF, that is, it is a grid map with equal distances between vertices that are connected by lines used by robots to easily navigate between the vertices, see Figure 2. The abstract plan outputted by MAPF solvers is, as defined, a sequence of locations that the agents visit. However, a physical agent has to translate these locations to a series of actions that the agent can perform. We assume that the agent can turn left and right and move forward. By concatenating these actions, the agent can perform all the required steps from the abstract plan (recall, that we are working with grid worlds). This translates to five possible actions at each time step - (1) wait, (2) move forward, (3,4) turn left/right and move, and (5) turn back and move. As the mobile robot cannot move backward directly, turning back is implemented as two turns right (or left). Ozobot robots used in our study, can directly perform these actions, which together with the specific map simplifies typical "robotics" problems such as localisation and control.

As the abstract steps may have durations different from the physical steps, the abstract plans, which are perfectly synchronised, may desynchronise when being executed, which may further lead to collisions. The intuition says that such desynchronisation will indeed happen. In our setting, the speed of the robots was set in such a way that moving along a line takes 1600ms and turning takes 800ms. Note that the real robots only blindly follow the computed plan and do not intervene if, for example, an obstacle is detected.

In the rest of the section, we describe the studied abstract MAPF models and possible transformations of abstract plans to executable sequences of physical actions. Let t_t be the time needed by the robot to turn by 90 degrees to either side and t_f be the time to move forward to the neighbouring vertex in the grid. Both t_t and t_f are nonzero. The time spent while the agent is performing the wait operation t_w will depend on each model.

3.1 Classical Model

The first and most straightforward model is a direct translation of the abstract plan to the action sequence. We shall call this a *classic* model. At the end of each timestep, an agent is facing in a direction. Based on the next location, the agent picks one of the five actions described above and performs it. This means that all move actions consist of possible turning and then going forward. There are no independent turning moves. As the two most common actions in abstract plans are (2) and (3,4), we suggest to set the time t_w of waiting actions to be $t_f + 1/2 * t_t$ as the average of durations of actions (2) and (3,4).

One can easily see that this simple model can be prone to desynchronisation, as turning adds time over agents that just move forward. To fix this synchronisation issue, we introduce a *classic+wait* model. The basic idea is that each abstract action takes the same time, which is realised by adding some void time to "fast" actions. The longest action is (5), therefore each action now takes $2 * t_t + t_f$ including the waiting action. The consequence is that plan execution takes longer time, which may not be desirable.

Note that both of these models do not require the MAPF algorithm and model to change. They only use different durations of abstract actions which are implemented in the translation of abstract plans to executable actions.

3.2 Robust Model

Another way to fix the synchronisation problem is to create a plan π that is robust to possible delays during execution. The k-robust plan is a valid MAPF plan that in addition requires for each vertex of the graph to be unoccupied for at least k time steps before another agent can enter it [8]. In our experiments, we choose k to be 1. We presume that this is a good balance between keeping the agents from colliding with each other while not prolonging the plan too much. The 1-robust plan is then translated to executable actions using the same principle as the *classic* model. This yields a *1-robust* model. Though, this model does not solve the synchronisation issue directly, it adds some slack that can prevent collisions caused by various reasons.

3.3 Split Actions Model

By making the model less abstract, we can directly represent the executable actions, in particular, by introducing an abstract turning action. In the reduction-based solvers, this can be done by splitting each vertex v_i from the original graph G into four new vertices $v_i^{up}, v_i^{right}, v_i^{down}, v_i^{left}$ indicating directions where the agent is facing to. The new edges now represent the turn actions, while the original edges correspond to move only actions. This change needs to be accompanied by constraints restricting the agents not to be at split vertices at the same time. The abstract plan is then translated to an executable plan in a direct way as the agent is given a sequence of individual actions wait, turn left/right, and move forward. The waiting time t_w is set as the bigger time of the remaining actions: $t_w = \max(t_t, t_f)$. We shall call this a *split* model.

To make the model even closer to reality, we can exploit the weighted MAPF [9], where each edge in the graph is assigned an integer value that denotes its length. The weighted MAPF solver finds a plan that takes these lengths into account. The lengths of turning edges are assigned a length of t_t and the other edges are assigned a length of t_f (or its scaled value to integers). The waiting time t_w is set as the smaller time of the remaining actions: $t_w = \min(t_t, t_f)$. We shall call this a *weighted-split* model or *w-split* for short.

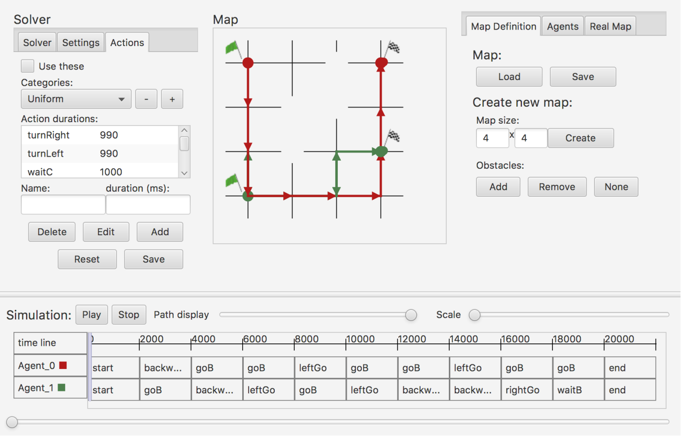
A final enhancement to the *weighted-split* model is to introduce k-robustness there. This will again ensure that the agents do not tend to move close to each other to avoid undesirable collisions. In this case, however, it is not enough to use 1-robustness, as the plan is split into more time steps. Instead, we use $\max(t_t, t_f)$-robustness. We shall call this *robust-weighted-split* model or *rw-split* for short.

4 Results of Experiments

For evaluation of various models, we developed software (Figure 3) that provides a visual editor to state the MAPF problems on a grid map, interface for solvers of the MAPF problems using the Picat system, visualisation of plans and plan execution, transformation of plans to control procedures for the Ozobot robots, and tools supporting execution of plans. The software is intended as a research tool for testing various abstract models of the MAPF problem on real robots.

For initial empirical comparison, we generated plans using each MAPF model for the problem instance described above and then we executed the plans five times in total for each model. Several properties were measured with results shown in Table 1.

Computed makespan is the makespan of the plan returned by the MAPF solver. It is measured by the (weighted) number of abstract actions and this is the value optimised by the solvers. Note that the *split* models have larger makespan than the rest because the *split* models use a finer resolution of actions, namely turning actions are included in the makespan calculation. This is even more noticeable with *w-split* and *rw-split*, where the moving-forward action has a duration (weight) of two. Total time is the actual time needed to complete the plan by all robots. To measure the level of desynchronisation, we introduced the Max Δ time. We made abstract plans for all robots



□ Use these

Categories: Uniform

Action durations:

turnRight	990
turnLeft	990
waitC	1000

Name:

duration (ms):

Map:

Create new map:

Map size: 4 x 4

Obstacles:

time line	0	2000	4000	6000	8000	10000	12000	14000	16000	18000	20000
Agent_0 🟥	start	backw...	goB	goB	leftGo	goB	goB	leftGo	goB	goB	end
Agent_1 🟩	start	goB	backw...	leftGo	goB	leftGo	backw...	backw...	rightGo	waitB	end

Figure 3: User interface of the MAPF system.

equally long by adding void wait actions to the end (where necessary). The Max Δ time is the time difference between the real end times of the first and last robots. This value is zero, if the robots remained synchronised during plan execution. The larger value means larger desynchronisation. All of the times are rounded to seconds because the measurements were conducted by hand.

The number of failed runs is also shown. The only model that did not finish any run is the *classic* model while the rest managed to finish all of the runs. A run fails if there is a collision that throws any of the robots off the track so the plan cannot be finished. The average number of collisions per run shows how many collisions that did not ruin the plan occurred. These collisions can range from small one, where the robots only touched each other and did not affect the execution of the plan, to big collisions, where the agent was slightly delayed in their individual plan, but still managed to finish the plan. For the *classic* model, where no execution finished, we present the number of collisions occurring before the major collision that stopped the plan.

5 Conclusions and Future Steps

The goal of the paper is showing that abstract models should be treated more carefully, when the results are supposed to be used in real environment. Our preliminary experiment showed that the most widely used MAPF model, the *classic* one, is actually not applicable even if the environment is made very close to the model. The reason is that durations of real actions are different from durations of abstract actions, which leads to desynchronisation of agents' plans. A naive extension to make all actions equally long worsens the quality of plan (makespan) significantly. Adding robustness to abstract plans helps, but as the Max Δ time shows, there is some desynchronisation, which may lead to collisions for longer plans. The *split* model uses abstraction closer to reality and adding weights makes the abstract plans even closer to real plans when executed. However, solving such models is more computationally expensive than solving the classical model [9].

The results show that there is indeed a gab between widely-used theoretical frameworks for MAPF and deployment of solutions in real environments. A wider experimental study is necessary

	Comp.	Failed	#Colls.	Total	Max Δ
	Mksp	Runs		Time [s]	[s]
classic	17	5	4	NA	5
classic+wait	17	0	4.2	53	0
1-robust	19	0	0	41	4
split	27	0	2	36	3
w-split	45	0	2.6	39	0
rw-split	47	0	0	39	0

Table 1: Real performance of Ozobots for studied models.

to understand better the relations between abstract models and real environments. For example, the ratio between the length of edges and the size of robots seems important (Ozobots have diameter of 3 cm and distance between nodes in our map is 5 cm). Note also, that blind execution of plans was assumed. It would be interesting to look at plan-execution policies that assume communication between agents and exploit information from sensors [10].

Acknowledgement
Roman Barták is supported by the Czech Science Foundation under the project P202/12/G061 and together with Jiří Švancara by the Czech-Israeli Cooperative Scientific Research Project 8G15027.

References

[1] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2009.

[2] Ozobot & Evollve, Inc. Ozobot — Robots to code, create, and connect with, 2018.

[3] Roman Barták, Jiří Švancara, Věra Škopková, and David Nohejl. Multi-agent path finding on real robots: First experience with ozobots. In *Advances in Artificial Intelligence – IBERAMIA 2018.* Springer, 2018.

[4] Malcolm Ross Kinsella Ryan. Exploiting subgraph structure in multi-robot path planning. *J. Artif. Intell. Res.*, 31:497–542, 2008.

[5] Pavel Surynek. Compact representations of cooperative path-finding as SAT based on matchings in bipartite graphs. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pages 875–882. IEEE Computer Society, 2014.

[6] Jingjin Yu and Steven M. LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[7] Roman Barták, Neng-Fa Zhou, Roni Stern, Eli Boyarski, and Pavel Surynek. Modeling and solving the multi-agent pathfinding problem in picat. In *29th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 959–966. IEEE Computer Society, 2017.

[8] Dor Atzmon, Ariel Felner, Roni Stern, Glenn Wagner, Roman Barták, and Neng-Fa Zhou. k-robust multi-agent path finding. In *Proceedings of the Tenth International Symposium on Combinatorial Search (SoCS)*, pages 157–158, 2017.

[9] Roman Barták, Jiří Švancara, and Marek Vlk. A scheduling-based approach to multi-agent path finding with weighted and capacitated arcs. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 748–756, 2018.

[10] Hang Ma, T. K. Satish Kumar, and Sven Koenig. Multi-agent path finding with delay probabilities. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3605–3612. AAAI Press, 2017.

Relation between Normalized Interval Vector and Imprecise Brief Function in Interval AHP

Masahiro Inuiguchi

Graduate School of Engineering Science, Osaka University
Toyonaka, Osaka 560-8531, Japan
inuiguti@sys.es.osaka-u.ac.jp

Abstract

The interval Analytic Hierarchy Process (AHP) [1] has been proposed to express vague evaluations of the decision maker as a normalized interval weight vector (NIWV). In this paper, we apply the evidence theory [2] to the representation of the vague evaluations as an alternative way. The interval weights can be seen as a model of imprecise probability distribution. Therefore, it is conceivable to apply the evidence theory to estimating imprecise weights from a given PCM because the evidence theory is also a model of imprecise probability. In the evidence theory, imprecise probabilities are expressed by a basic probability assignment (BPA). Accordingly, a basic probability assignment (BPA) is used for representing vague priority weights instead of an NIWV. We propose an approach to estimation of a BPA from a given pairwise comparison matrix (PCM). The proposed approach can be used for the estimation of a BPA over alternatives on each criterion as the conventional AHP does. However, in order to avoid the a labyrinthine argument not related to the main result of this paper, we concentrate on the estimation of a BPA over criteria from a given PCM. The application of the evidence theory has the advantage that the normalization condition of a BPA is well defined while the normalization condition of an NIWV is controversial [3]. By the comparison between the interval AHP and the AHP based on the evidential theory, we may have some new perspective on the normalization condition in the interval AHP.

We formulate the problem estimating a BPA from a given PCM as a linear programming problem in parallel to the estimation problem of an NIWV from a give PCM. We investigate the relation between the solutions to BPA estimation problem and those to NIWV estimation problem. Moreover, the relation between the preference relation induced from the estimated set of BPAs and that induced from the set of estimated NIWVs. We show that the formulated BPA estimation problem is equivalent to the NIWV estimation problem with an additional constraint that the sum of centers of interval weights is not less than one. As the result, a half range of NIWVs is well understood as a whole range of BPAs and an NIWV corresponds to an imprecise BPA.

Acknowledgement
This work is supported by JSPS KAKENHI Grant Number JP17K18952.

References

[1] Sugihara, K., Tanaka, H.: Interval evaluations in the analytic hierarchy process by possibilistic analysis, Computational Intelligence, 17, 567–579 (2001)

[2] Shafer, G.: A Mathematical Theory of Evidence, Princeton University Press (1976)

[3] Inuiguchi, M.: Non-uniqueness of interval weight vector to consistent interval pairwise comparison matrix and logarithmic estimation methods, Proceedings of 5th International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making, LNAI 9978, 39?50, Springer (2016)

Hidden Conflict of Belief Functions

Milan DANIEL[1] and Václav KRATOCHVÍL[2]

[1] *Jan Becher - Karlovarská Becherovka, a.s., Pernod Ricard group*
Přemyslovská 43, CZ - 130 00 Prague 3
Czech Republic
milan.daniel@pernod-ricard.com

[2] *Czech Academy of Sciences, Inst. Information Theory and Automation*
Pod Vodárenskou veží 4, CZ - 182 08 Prague 8
Czech Republic
velorex@utia.cas.cz

Abstract

Hidden conflict of belief functions in case where the sum of all multiples of conflicting belief masses being equal to zero was observed. Degrees of hidden conflict and of non-conflictness are defined and analysed including full non-conflictness. Hidden conflict between two belief functions is distinguished from internal hidden conflict(s) of the individual belief function(s). Finally, computational issues of hidden conflict and non-conflictness are presented.

Keywords: belief functions, Dempster-Shafer theory, uncertainty, conflicting belief masses, internal conflict, conflict between belief functions, hidden conflict, full non-conflictness.

1 Introduction

When combining belief functions (BFs) by the conjunctive rules of combination, some conflicts often appear (they are assigned either to $\emptyset$ by non-normalised conjunctive rule $\odot$ or distributed among other belief masses by normalisation in Dempster's rule of combination $\oplus$). Combination of conflicting BFs and interpretation of their conflicts are often questionable in real applications. Thus a series of papers related to conflicts of BFs was published, e.g. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. A new interpretation of conflicts of belief functions was introduced in [11]: important distinction of internal conflicts of individual BFs (due to their inconsistency) from conflicts between BFs (due to conflict/contradiction of evidences represented by the BFs) was introduced there. Note that zero sum of all multiples of conflicting belief masses (denoted by $m_{\odot}(\emptyset)$) is usually considered as non-conflictness of the belief functions in all these approaches.

When analysing the conflict between belief functions based on their non-conflicting parts[1] [4] a positive value of conflict was observed even in a situation when sum of all multiples of conflicting belief masses equals to zero. This arose a series of new questions: how to interpret the sum of conflicting masses, is the conflict based on non-conflicting parts of belief functions correct? Some of the answers are provided in this text. They are positive in favour of the conflict based on non-conflicting parts. This led us to a definition of a hidden conflict of BFs (Section 3).

Going further, different levels / degrees of hidden conflicts are defined and a maximal degree of hidden conflict is investigated. Analogously to the degrees of hidden conflict, there also exist different degrees of non-conflictness. Full non-conflictness and conditions, under which belief functions are fully non-conflicting, are defined and presented.

In accordance with approach from [11], there are observed and presented not only hidden conflicts between two belief functions, but also internal hidden conflicts of individual BFs (Section 5). Finally, computational aspects of hidden conflict are presented in Section 6.

[1] Conflicting and non-conflicting parts of belief functions originally come from [12].

2 Preliminaries

We assume classic definitions of basic notions from theory of *belief functions* [13] on finite exhaustive frames of discernment $\Omega_n = \{\omega_1, \omega_2, ..., \omega_n\}$.

A *basic belief assignment (bba)* is a mapping $m : \mathcal{P}(\Omega) \longrightarrow [0,1]$ such that $\sum_{A \subseteq \Omega} m(A) = 1$; the values of the bba are called *basic belief masses (bbm).* $m(\emptyset) = 0$ is usually assumed. $\mathcal{P}(\Omega) = \{X | X \subseteq \Omega\}$ is a *power-set* of Ω. A *belief function (BF)* is a mapping $Bel : \mathcal{P}(\Omega) \longrightarrow [0,1]$, $Bel(A) = \sum_{\emptyset \neq X \subseteq A} m(X)$. A *plausibility function* $Pl : \mathcal{P}(\Omega) \longrightarrow [0,1]$, $Pl(A) = \sum_{\emptyset \neq A \cap X} m(X)$. Because there is a unique correspondence between m and corresponding Bel and Pl, we often speak about m as of a belief function.

A *focal element* is a subset of the frame of discernment $X \subseteq \Omega$, such that $m(X) > 0$. If all focal elements are *singletons* (i.e. one-element subsets of Ω), then we speak about a *Bayesian belief function* (BBF); in fact, it is a probability distribution on Ω. If there are only focal elements such that $|X| = 1$ or $|X| = n$ we speak about *quasi-Bayesian BF (qBBF).* In the case of $m(\Omega) = 1$ we speak about *vacuous BF* (VBF) and about *a non-vacuous BF* otherwise. In the case of $m(X) = 1$ for $X \subset \Omega$ we speak about *categorical BF.* If all focal elements have a non-empty intersection, we speak about *a consistent BF*; and if all of them are nested, about *a consonant BF.*

Dempster's (normalized conjunctive) rule of combination $\oplus$: $(m_1 \oplus m_2)(A) = \sum_{X \cap Y = A} K m_1(X)$ $m_2(Y)$ for $A \neq \emptyset$, where $K = \frac{1}{1-\kappa}$, $\kappa = \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)$, and $(m_1 \oplus m_2)(\emptyset) = 0$, see [13]. Putting $K = 1$ and $(m_1 \textcircled{\cap} m_2)(\emptyset) = \kappa$ we obtain the *non-normalized conjunctive rule of combination* $\textcircled{\cap}$, see e. g. [14].

Smets' *pignistic probability* is given by $BetP(\omega_i) = \sum_{\omega_i \in X \subseteq \Omega} \frac{1}{|X|} \frac{m(X)}{1-m(\emptyset)}$, see e.g. [14]. *Normalized plausibility of singletons*[2] of Bel is a probability distribution Pl_P such that $Pl\_P(\omega_i) = \frac{Pl(\{\omega_i\})}{\sum_{\omega \in \Omega} Pl(\{\omega\})}$ [15, 16].

A conflict of BFs Bel', Bel'' based on their non-conflicting parts Bel'_0, Bel''_0 is defined by the expression $Conf(Bel', Bel'') = (m'_0 \textcircled{\cap} m''_0)(\emptyset)$, where non-conflicting part Bel_0 (of a BF Bel) is unique consonant BF such that $Pl\_P_0 = Pl\_P$ (normalized plausibility of singletons corresponding to Bel_0 is the same as that corresponding to Bel); m_0 is a bba related to Bel_0. For an algorithm to compute Bel_0 see [4].

3 Hidden Conflict

3.1 An Introductory Example

Let us assume two simple consistent belief functions Bel' and Bel'' on $\Omega_3 = \{\omega_1, \omega_2, \omega_3\}$ given by the bbas $m'(\{\omega_1, \omega_2\}) = 0.6$, $m'(\{\omega_1, \omega_3\}) = 0.4$, and $m''(\{\omega_2, \omega_3\}) = 1.0$. Then $(m' \textcircled{\cap} m'')(\emptyset) = 0$ what seems — and it is usually considered — to be a non-conflictness of m' and m'', but there is positive conflict based on non-conflicting parts $Conf(Bel', Bel'') = (m'_0 \textcircled{\cap} m''_0)(\emptyset) = 0.4 > 0$. (This holds true despite of Theorem 4 from [4] which should be revised in future).

We can easily verify this: the only focal element of m'' has a non-empty intersection with both focal elements of m', thus $\sum_{(X \cap Y) = \emptyset} m'(X) m''(Y)$ is an empty sum; Bel'' is consonant, thus $Bel''_0 = Bel''$, $m''_0 = m''$, $Pl'(\{\omega_1\}) = 1$, $Pl'(\{\omega_2\}) = 0.6$, $Pl'(\{\omega_3\}) = 0.4$, thus $m'_0(\{\omega_1\}) = 0.4$, $m'_0(\{\omega_1, \omega_2\}) = 0.2$, $m'_0(\{\omega_1, \omega_2, \omega_3\}) = 0.4$, hence $Conf(Bel', Bel'') = (m'_0 \textcircled{\cap} m''_0)(\emptyset) = m'_0(\{\omega_1\})$ $m''_0(\{\omega_2, \omega_3\}) = 0.4 \cdot 1 = 0.4$.

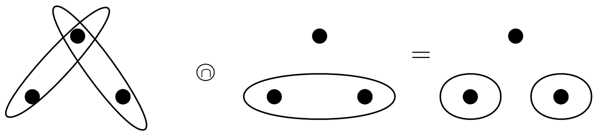


Figure 1: Introductory Example: focal elements of m', m'', and of $m' \textcircled{\cap} m''$.

[2] Plausibility of singletons is called *contour function* by Shafer in [13], thus $Pl\_P(Bel)$ is a normalization of contour function in fact.

Table 1: Hidden conflict in the Introductory Example

X :	$\{\omega_1\}$	$\{\omega_2\}$	$\{\omega_3\}$	$\{\omega_1,\omega_2\}$	$\{\omega_1,\omega_3\}$	$\{\omega_2,\omega_3\}$	Ω_3	$\emptyset$
$m'(X)$:	0.0	0.0	0.0	0.60	0.40	0.00	0.00	—
$m''(X)$:	0.0	0.0	0.0	0.00	0.00	1.00	0.00	—
$(m'\textcircled{\cap}m'')(X)$:	0.00	0.60	0.40	0.00	0.00	0.00	0.00	0.00
$(m'\textcircled{\cap}m''\textcircled{\cap}m'')(X)$:	0.00	0.60	0.40	0.00	0.00	0.00	0.00	0.00
$(m'\textcircled{\cap}m''\textcircled{\cap}m')(X)$:	0.00	0.36	0.16	0.00	0.00	0.00	0.00	0.48
$(m'\textcircled{\cap}m''\textcircled{\cap}m'\textcircled{\cap}m'')(X)$:	0.00	0.36	0.16	0.00	0.00	0.00	0.00	0.48

3.2 Interpretation of the Example – Observation of a Hidden Conflict

The following questions arise: Does $(m'\textcircled{\cap}m'')(\emptyset) = 0$ really represent non-conflictness of respective BFs as it is usually assumed? Is the definition of conflict based on non-conflicting parts correct? Are m' and m'' conflicting? What does $(m'\textcircled{\cap}m'')(\emptyset) = 0$ mean?

Suppose that Bel' and Bel'' are non-conflicting now. Thus their combination should be also non-conflicting with both of them. Does this hold for BFs from our example? This holds true when we combine $m'\textcircled{\cap}m''$ with m'' one more time (assuming m'' comming from two independent belief sources). It follows from the idempotency of categorical m'': $m'\textcircled{\cap}m''\textcircled{\cap}m'' = m'\textcircled{\cap}m''$ and therefore $(m'\textcircled{\cap}m''\textcircled{\cap}m'')(\emptyset) = 0$ again. On the other hand, we obtain positive $(m'\textcircled{\cap}m''\textcircled{\cap}m')(\emptyset) = (m'\textcircled{\cap}m'\textcircled{\cap}m'')(\emptyset) = 0.48$ (assuming m' comming from two independent belief sources). See Table 1 and Figure 2. When m'' and m' are combined once, then we observe $m_{\textcircled{\cap}}(\emptyset) = 0$. When combining m'' with m' twice then $m_{\textcircled{\cap}}(\emptyset) = 0.48$. We observe some kind of *a hidden conflict*. Moreover, both individual BFs are consistent. I.e. there are no internal conflicts. Thus the hidden conflict is *hidden conflict between the BFs* and we have an argument for correctness of positive value of $Conf(Bel', Bel'')$.

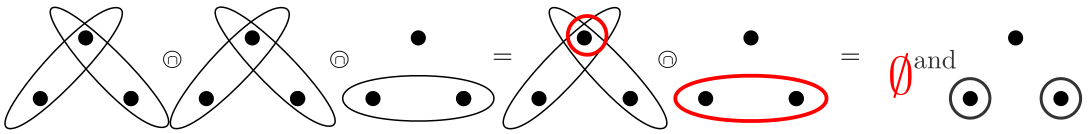


Figure 2: Arising of a hidden conflict between BFs in the Introductory Example: focal elements of m', m', m'' — $m'\textcircled{\cap}\, m', m''$ and of $(m'\textcircled{\cap}\, m')\textcircled{\cap}\, m''$.

What is a decisional interpretation of our BFs? Contours, i.e. plausibilities of singletons are $Pl' = (1.0, 0.6, 0.4)$ and $Pl'' = (0.0, 1.0, 1.0)$, we obtain $Pl\_P' = (0.5, 0.3, 0.2)$ and $Pl\_P'' = (0.0, 0.5, 0.5)$ by normalization; thus at Bel', ω_1 is significantly preferred, whereas at Bel'', one of ω_2, ω_3; this is also an argument for mutual conflictness of the BFs. Considering Smets' pignistic probability we obtain $BetP' = (0.5, 0.3, 0.2)$ and $BetP'' = (0.0, 0.5, 0.5)$, just the same values as in the case when normalized plausibility of singletons (normalized contour) is used for decision. Both the, in general different, probabilistic approximations $BetP$ and Pl_P give highest value to different singletons for Bel' and Bel''. Thus the argument for mutual conflictness of the BFs is strengthened and we obtain the same pair of incompatible decisions based on the BFs in both frequent decisional approaches: using either normalized contour (which is compatible with conjunctive combination of BFs) or pignistic probability (designed for betting).

Hence $(m'\textcircled{\cap}\, m'')(\emptyset)$ does not mean non-conflictness of the BFs. It means a simple or partial compatibility of their focal elements only.

3.3 Objections Against Our Interpretation

There are several objections against our interpretation of the introductory example:

- *in case of a combination of two identical belief functions some idempotent rule of combination should be used.* Yes, this would be right for belief functions coming from two dependent belief sources. But this is not true for two or more numerically same BFs coming from two or more independent belief sources.

- *The result is not surprising, because conflict is increasing when combining more belief functions.* This is definitely not true:
  - to be correct it should be stated 'non-decreasing' instead of increasing.
  - more precisely, a conflict is non-decreasing when more conflicting BFs are combined. When truly non-conflicting BFs are combined, any positive conflict can never arise there; e.g. $(\textcircled{\cap}_1^k Bel_1 \textcircled{\cap} \textcircled{\cap}_1^k Bel_2)(\emptyset) = 0$ for any $k > 0$ and Bel_i on Ω_3 given by $m_1(\{\omega_1\}) = 0.3$, $m_1(\{\omega_1, \omega_2\}) = 0.2$, $m_1(\{\omega_1, \omega_3\}) = 0.1$, $m_1(\{\omega_1, \omega_2, \omega_3\}) = 0.4$, $m_2(\{\omega_1, \omega_3\}) = 0.7$, $m_2(\{\omega_1, \omega_2, \omega_3\}) = 0.3$.
  - Thus an appearance of a positive $(m'\textcircled{\cap}m'\textcircled{\cap})(\emptyset)$ implies a conflict between m' and m'' or between m' and m' in case of consonant BFs; there cannot be any conflict between consonant m' and m' thus there must be conflict between m' and m''.
- The result is rather unsurprising. Because one can see clearly that the hidden conflict occurs when the first combination results in disjoint focal sets. Yes, in the very simple Introductory Example, this may by unsurprising for someone; but there are no disjoint sets after the first combination in the following Little Angel example. Moreover, this should be surprising for all who accept the following assumption / axiom: BFs Bel' and Bel'' are non-conflicting whenever $(m'\textcircled{\cap}m'')(\emptyset) = 0$, e. g. [1, 5, 7, 8] and the previous Daniel's publications, e. g. [2, 4, 3].
- It is obvious that a combination results in a conflict if a Bayesian BF ($m'\textcircled{\cap}m''$ in the Introductory Example) is combined with any other BF. Yes, this is true in the very simple introductory example, but not in a general example, see e. g. the following Little Angel example again.

Analysing these objections we can see, that is was not easy to observe the hidden conflicts, in simple cases the observation seems to be obvious, thus not interesting, in more general examples this seems to be really hidden.

3.4 Definition of Hidden Conflict

Definition 1 *Let us suppose two BFs Bel', Bel'' defined by bbas m', m'', such that $(m'\textcircled{\cap}m'')(\emptyset) = 0$. If there further holds $(m'\textcircled{\cap}m''\textcircled{\cap}m')(\emptyset) > 0$ or $(m'\textcircled{\cap}m''\textcircled{\cap}m'')(\emptyset) > 0$ we say that there is* a hidden conflict *of the BFs.*

Observation 1 *A condition $(m'\textcircled{\cap}m''\textcircled{\cap}m')(\emptyset) > 0$ or $(m'\textcircled{\cap}m''\textcircled{\cap}m'')(\emptyset) > 0$ from Definition 1 is equivalent to the following condition $(m'\textcircled{\cap}m''\textcircled{\cap}m'\textcircled{\cap}m'')(\emptyset) > 0$.*

We have to note that a hidden conflict is quite a new phenomenon, qualitatively different from the ideas of all previous Daniel's works on conflict of belief functions and also different from the other referred approaches. Till now, it was supposed that $m_{\textcircled{\cap}}(\emptyset)$ includes both conflict between BFs and also internal conflicts of individual BFs. Thus conflict between BFs was supposed to be less or equal to $m_{\textcircled{\cap}}(\emptyset)$. Here, we deal with a situation of a positive conflict between BFs while $m_{\textcircled{\cap}}(\emptyset) = 0$.

We have already observed that $m_{\textcircled{\cap}}(\emptyset) = 0$ does not mean full non-conflictness of BFs and that the condition $(m'\textcircled{\cap}m''\textcircled{\cap}m'\textcircled{\cap}m'')(\emptyset) > 0$ together with $(m'\textcircled{\cap}m'')(\emptyset) = 0$ defines hidden conflict. What about the condition $(m'\textcircled{\cap}m''\textcircled{\cap}m'\textcircled{\cap}m'')(\emptyset) = 0$? Is this condition sufficient for full non-conflictness of BFs Bel' and Bel''? May some conflict be still hidden there?

The zero version of the condition seems to imply non-conflictness on Ω_3, the frame of discernment of the Introductory Example. To solve the question in general, we have to consider a larger frame of discernment.

3.5 Little Angel Example

For Ω_5 one can find the following Little Angel Example, see Table 2. Similarly to Introductory Example, we have two consistent BFs Bel^i and Bel^{ii} with disjoint sets of max-plausibility elements while zero condition $(m^i\textcircled{\cap}m^{ii})(\emptyset) = 0$ holds true.

Table 2: Hidden Conflict in the Little Angel Example

X :	$A=\{\omega_1,\omega_2,\omega_5\}$	$B=\{\omega_1,\omega_2,\omega_3,\omega_4\}$	$C=\{\omega_1,\omega_3,\omega_4,\omega_5\}$	$X=\{\omega_2,\omega_3,\omega_4,\omega_5\}$	$\emptyset$
$m^i(X)$:	0.1	0.30	0.60	0.00	—
$m^{ii}(X)$:	0.0	0.00	0.00	1.00	—

X :	$A{\cap}X$	$B{\cap}X$	$C{\cap}X$	$A{\cap}B{\cap}X$	$A{\cap}C{\cap}X$	$B{\cap}C{\cap}X$	$\emptyset$
$(m^i\textcircled{\cap}m^{ii})(X)$:	0.1	0.3	0.6	0.0	0.0	0.0	0.000
$(m^i\textcircled{\cap}m^{ii}\textcircled{\cap}m^{ii})(X)$:	0.10	0.30	0.60	0.00	0.00	0.00	0.00
$(m^i\textcircled{\cap}m^i\textcircled{\cap}m^{ii})(X)$:	0.01	0.09	0.36	0.06	0.12	0.36	0.00
$(m^i\textcircled{\cap}m^i\textcircled{\cap}m^{ii}\textcircled{\cap}m^{ii})(X)$:	0.01	0.09	0.36	0.06	0.12	0.36	0.00
$(m^i\textcircled{\cap}m^{ii}\textcircled{\cap}m^{ii}\textcircled{\cap}m^{ii})(X)$:	0.010	0.090	0.360	0.060	0.120	0.360	0.000
$(m^i\textcircled{\cap}m^i\textcircled{\cap}m^i\textcircled{\cap}m^{ii})(X)$:	0.001	0.027	0.216	0.036	0.126	0.486	0.108
$m^i\textcircled{\cap}m^i\textcircled{\cap}m^i\textcircled{\cap}m^{ii}\textcircled{\cap}m^{ii}\textcircled{\cap}m^{ii}(X)$:	0.001	0.027	0.216	0.036	0.126	0.486	0.108

In addition to Introductory Example, $(m^i\textcircled{\cap}m^{ii}\textcircled{\cap}m^i\textcircled{\cap}m^{ii})(\emptyset) = 0$ (see Table 2) while $Conf($ $Bel^i, Bel^{ii}) = 0.1$ is positive again. Positiveness of the $Conf$ value can be easily seen from the fact that sets of max-plausibility elements are disjoint for Pl^i and Pl^{ii}. Numerically, we have again $Bel_0^{ii} = Bel^{ii}$, and $Pl\_P^i = (\frac{10}{39}, \frac{4}{39}, \frac{9}{39}, \frac{9}{39}, \frac{7}{39})$. We obtain $m_0^i(\{\omega_1\}) = 0.1, m_0^i(\{\omega_1, \omega_3, \omega_4\}) = 0.2, m_0^i(\{\omega_1, \omega_3, \omega_4, \omega_5\}) = 0.3, m_0^i(\{\Omega_5\}) = 0.4$, and $Conf(Bel^i, Bel^{ii}) = m_0^i(\{\omega_1\})m^{ii}(X) = 0.1$. Analogous arguments hold true for the positive $Conf$ and hidden conflict again (of the 2nd degree this time). $BetP^i = (0.2583, 0.1083, 0.2250, 0.2250, 0.1833)$ which is not numerically the same as Pl_P^i, but both prefer ω_1, whereas $BetP^{ii} = Pl\_P^{ii} = (0.00, 0.25, 0.25, 0.25, 0.25)$.

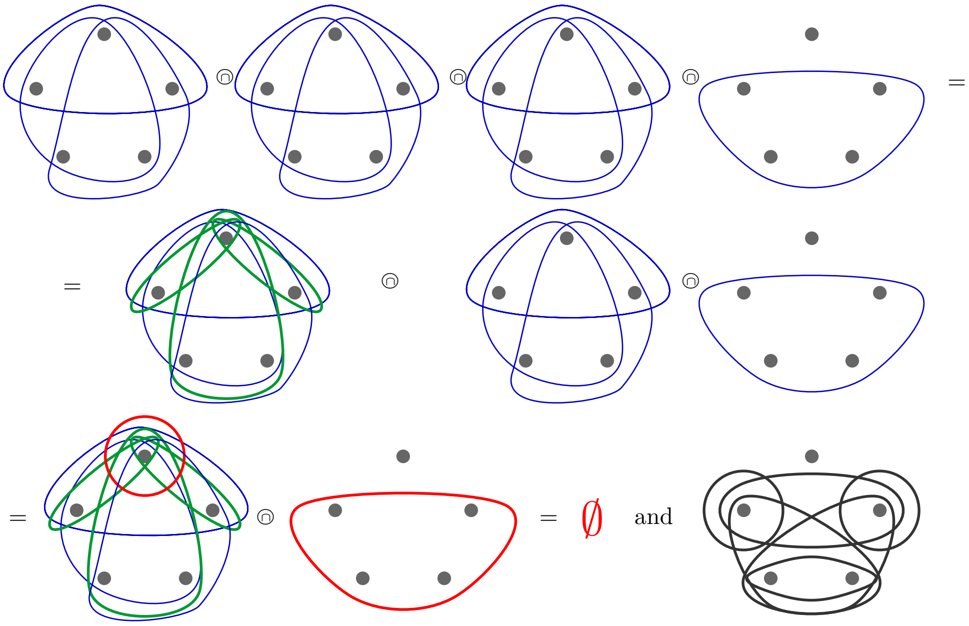


Figure 3: Arrising of a hidden conflict between BFs in the Little Angel Example. Focal elements of m^i, m^{ii}, $m^i\textcircled{\cap}m^i$, $m^i\textcircled{\cap}m^i\textcircled{\cap}m^i$ and of $(m^i\textcircled{\cap}m^i\textcircled{\cap}m^i)\textcircled{\cap}m^{ii}$. Red-colored focal elements are those responsible for creation of the empty-set in the last step.

For an existence of a hidden conflict, it is the structure of focal elements that is important — not their belief masses. Belief masses are important for the size of a conflict. In general, we can take $m^i(A) = a$, $m^i(B) = b$, $m^i(C) = c$, for A, B, C defined in Table 2, and for any $a, b, c > 0$, such that $a + b + c = 1$, we obtain $m(\emptyset) = 6abc$ as a hidden conflict of the 2nd degree (in our numeric case there is $6abc = 6 \cdot 0.1 \cdot 0.3 \cdot 0.6 = 0.108$). For graphical presentation of the Little Angel Example see Figure 3.

Degrees of hidden conflict, its maximal value, and the issue of full non-conflictness will be analyzed in the following section.

4 Degrees of Hidden Conflict and Full Non-conflictness

When analyzing examples from the previous section, we have observed different degrees of hidden conflict. We can formalize it in the next definition.

Definition 2 *Assume two BFs Bel^i, Bel^{ii} defined by bbas m^i, m^{ii}, such that for some $k > 0$ $(\circledcirc_{j=1}^{k}(m^i \circledcirc m^{ii}))(\emptyset) = 0$. If there further holds $(\circledcirc_{j=1}^{k+1}(m^i \circledcirc m^{ii}))(\emptyset) > 0$ we say that there is* a conflict of BFs *Bel^i and Bel^{ii}* hidden in the *k*-th degree*.*

Analogously to particular degrees of hidden conflict, there are degrees of non-conflictness. Particular degrees of non-conflictness are not very important. However, there is an important question whether there is some hidden conflict or not, i.e. whether or not the BFs in question are fully non-conflicting.

Definition 3 *We say that BFs Bel^i and Bel^{ii} are* fully non-conflicting *if $(m^i \circledcirc m^{ii})(\emptyset) = 0$ and, further, if there is no hidden conflict of any degree. I.e. if $(\circledcirc_{j=1}^{k}(m^i \circledcirc m^{ii}))(\emptyset) = 0$ for any $k \geq 0$.*

Thus there is a question how many times we have to combine $(m^i \circledcirc m^{ii})$, i.e., for which k value of $(\circledcirc_{j=1}^{k}(m^i \circledcirc m^{ii}))(\emptyset)$ shows whether there is some hidden conflict of the BFs Bel^i and Bel^{ii} or not. For answers to this question see corollaries of the following two theorems.

Theorem 4 (maximal degree of hidden conflict) *For any non-vacuous BFs Bel^i and Bel^{ii} defined by m^i and m^{ii} on any frame Ω_n it holds that*

$$(\circledcirc_{j=1}^{n-1}(m^i \circledcirc m^{ii}))(\emptyset) = 0 \quad \textit{iff} \quad (\circledcirc_{j=1}^{k}(m^i \circledcirc m^{ii}))(\emptyset) = 0$$

for any $k > n - 2$.

Corollary 5 *Hidden conflict of any non-vacuous BFs on any Ω_n has always degree less or equal to $n - 2$; i.e., the condition*

$$(\circledcirc_{j=1}^{n-1}(m^i \circledcirc m^{ii}))(\emptyset) = 0$$

always means full the non-conflictness of any BFs Bel^i and Bel^{ii} on any Ω_n.

Let us present an example of such highly hidden conflict now.

Example 6 *Example of hidden conflict of the $(n-2)$-th degree: Let us suppose n-element frame of discernment $\Omega_n = \{\omega_1, \omega_2, ..., \omega_n\}$. Bel^i and Bel^{ii} are given by $m^i(\{\omega_1, \omega_2, ..., \omega_{n-1}\}) = \frac{1}{n-1}$, $m^i(\{\omega_1, \omega_2, ..., \omega_{n-2}, \omega_n\}) = \frac{1}{n-1}$, $m^i(\{\omega_1, \omega_2, ..., \omega_{n-3}, \omega_{n-1}, \omega_n\}) = \frac{1}{n-1}$, . . . $m^i(\{\omega_1, \omega_3, \omega_4, ..., \omega_n\}) = \frac{1}{n-1}$, $m^{ii}(\{\omega_2, \omega_3, ..., \omega_n\}) = \frac{1}{1}$. At $m^i \circledcirc m^i$ $(n-2)$-element focal elements appear, at $m^i \circledcirc m^i \circledcirc m^i$ $(n-3)$-element focal elements appear, at $\circledcirc_{j=1}^{k} m^i$ $(n-k)$-element focal elements appear, at $\circledcirc_{j=1}^{n-2} m^i$ 2-element focal elements appear, all these focal elements have non-empty intersections with the only focal element of m^{ii}, and finally at $\circledcirc_{j=1}^{n-1} m^i$ singleton focal element $\{\omega_1\}$ appears which has empty intersection with the only focal element of m^{ii} $\{\omega_2, \omega_3, ..., \omega_n\}$.*

What does m^i express? It gives a big support to all elements of the frame, to the entire frame Ω_n and even greater support to ω_1 which is included in all focal elements; ω_1 is preferred and, moreover, it has plausibility 1. We can modify m^i and express this more easily: $\overline{m}^i(\Omega_n) = \frac{n-1}{n}$, $\overline{m}^i(\{\omega_1\}) = \frac{1}{n}$, or more generally, $\widetilde{m}^i(\Omega_n) = 1 - a$, $\widetilde{m}^i(\{\omega_1\}) = a$ for some $0 < a < 1$. We can easily see evident conflict corresponding to positive $\overline{m}(\emptyset) = (\overline{m}^i \circledcirc m^{ii})(\emptyset) = \frac{1}{n}$, $\widetilde{m}(\emptyset) = a$ for these modifications of m^i. Hence either hidden conflict of the $(n-2)$-th degree of m^i and m^{ii} or positive $Conf(m^i, m^{ii}) = Conf(\overline{m}^i, m^{ii}) = \frac{1}{n}$ should not be very surprising.

We have to note that the Introductory Example is a special instance of Example 6 for $n = 3$.

Theorem 7 *(i) Any non-vacuous BFs Bel^i, Bel^{ii} have a conflict hidden at most in $(c$–$1)$-th degree where $c = min(c^i, c^{ii}) + sgn(|c^i - c^{ii}|)$. where c^i, c^{ii} are maximal cardinalities of focal elements of Bel^i, Bel^{ii} different from Ω. In the other words*

$$(\circledcirc_{j=1}^{c}(m^i \circledcirc m^{ii}))(\emptyset) = 0 \quad \textit{iff} \quad (\circledcirc_{j=1}^{k}(m^i \circledcirc m^{ii}))(\emptyset) = 0$$

for any $k \geq c = min(c^i, c^{ii}) + sgn(|c^i - c^{ii}|)$.
(ii) There are no hidden conflicts of any non-vacuous BFs on any two-element frame Ω_2.
(iii) There are no hidden conflicts of any non-vacuous quasi-Bayesian BFs on any frame Ω_n.
(iv) For a BF Bel^i and a quasi-Bayesian BF Bel^{ii} there is a hidden conflict of (at most) the first degree; if it appears then it is an internal conflict of Bel^{ii} in fact.

Corollary 8 *(i) Assume two non-vacuous BFs Bel^i, Bel^{ii} on Ω_n. The zero value of the expression $(\circledcirc_{j=1}^{c}(m^i \circledcirc m^{ii}))(\emptyset)$, i.e., the condition*

$$(\circledcirc_{j=1}^{c}(m^i \circledcirc m^{ii}))(\emptyset) = 0$$

means full non-conflictness of the BFs for $c = min(c^i, c^{ii}) + sgn(|c^i - c^{ii}|)$, where c^i, c^{ii} are maximal cardinalities of focal elements of Bel^i, Bel^{ii} different from Ω_n.
(ii) For any two non-vacuous quasi Bayesian BFs Bel^i, Bel^{ii} on any frame of discernment Ω_n the condition $(m^i \circledcirc m^{ii})(\emptyset) = 0$ always means full non-conflictness of the BFs.
(iii) For any BF Bel^i and any quasi-Bayesian BF Bel^{ii} the condition $(\circledcirc_{j=1}^{2}(m^i \circledcirc m^{ii}))(\emptyset) = 0$ always means full non-conflictness of the BFs.

5 Internal Hidden Conflict

Little Angel Modification

Let us take m^{iii} instead of m^i, such that $m^{iii}(A) = m^i(A)$, $m^{iii}(C) = m^i(C)$, and $m^{iii}(D) = m^{iii}(\{\omega_2, \omega_3, \omega_4\}) = 0.30$ instead of $m^i(B)$. There is $(m^{ii} \circledcirc m^{ii} \circledcirc m^{iii} \circledcirc m^{iii})(\emptyset) = 0$, but $(m^{ii} \circledcirc m^{iii} \circledcirc m^{iii} \circledcirc m^{iii})(\emptyset) > 0$, even $(m^{iii} \circledcirc m^{iii} \circledcirc m^{iii})(\emptyset) > 0$, i.e. $(\circledcirc_1^3 m^{iii})(\emptyset) > 0$, for detail see [17].

We observe a conflict of the belief functions hidden in the 2-nd degree again. Nevertheless, the situation of focal elements is different now: the only focal element X of $m^{ii} = \circledcirc_1^3 m^{ii}$ has non-empty intersection with any focal element of $\circledcirc_1^3 m^{iii}$, but $(\circledcirc_1^3 m^{iii})(\emptyset) > 0$ now. Thus this is not a hidden conflict between m^{ii} and m^{iii}, but *an internal hidden conflict* of m^{iii}.

Computation of $\circledcirc_1^k(m' \circledcirc m'')(\emptyset)$ and internal hidden conflict have a relation to Martin's auto-conflict [18, 8], thus we can speak about hidden auto-conflict here. See [19] for more details.

In general, we can observe internal hidden conflict when at least one of BFs in hidden conflict is not consistent. Let us present an example with highly hidden internal hidden conflict of the $(n-2)$-th degree on a general frame of discernment now.

Example 9 *Let us consider the following modification of Example 6 on Ω_n. Instead of m^i we take m^{iii} having all focal elements of cardinality $n-1$, such that $m^{iii}(\Omega_n \setminus \{\omega\}) = \frac{1}{n}$ for any $\omega \in \Omega_n$; m^{ii} same as in Example 6. m^{iii} is not consistent; $Pl^{iii}(\{\omega\}) = \frac{n-1}{n}$ for any $\omega \in \Omega_n$. We observe hidden conflict of the $(n$–$2)$-th degree again. Because of same plausibilities of all singletons $m''_0(\Omega_n) = 1$ and $Conf(m^{iii}, m^{ii}) = 0$ now.*

There is a positive hidden conflict of BFs Bel^{iii} and Bel^{ii}, but zero conflict between them. We say that there is *an internal hidden conflict.* This corresponds to non-consistency of BF Bel^{iii}; Bel^{ii} is consistent thus there is an internal hidden conflict of BF Bel^{iii} in this case.

A numeric example was computed on Ω_{16}, see Table 3 for a comparison of focal elements and $m_{\circledcirc}(\emptyset)$ values of Examples 6 and 9. For simplicity, same bbms $m^i(X) = \frac{1}{15}$ and $m^{ii}(X) = \frac{1}{16}$ were used there.

Ω_{16}		$m' = m^i \quad m'' = m^{ii}$			$m' = m^{iii} \quad m'' = m^{ii}$		
Degree	$m_{\textcircled{\cap}}$	no of f.e.	Card. of f.e.	$m_{\textcircled{\cap}}(\emptyset)$	no of f.e.	Card. of f.e.	$m_{\textcircled{\cap}}(\emptyset)$
–	m'	15	15	–	16	15	–
–	m''	1	15	–	1	15	–
0	$m' \textcircled{\cap} m''$:	15	14	0	16	14–15	0
1	$m' \textcircled{\cap} m'' \textcircled{\cap} m' \textcircled{\cap} m''$	120	13–14	0	121	13–15	0
2	$\textcircled{\cap}_{j=1}^{3}(m' \textcircled{\cap} m'')$	575	12–14	0	576	12–15	0
	. . .			0			0
k	$\textcircled{\cap}_{j=1}^{k+1}(m' \textcircled{\cap} m'')$	. . .	(14-k)–14	0	. . .	(14-k)–15	0
	. . .			0			0
13	$\textcircled{\cap}_{j=1}^{14}(m' \textcircled{\cap} m'')$	32766	1–14	0	32767	1–15	0
14	$\textcircled{\cap}_{j=1}^{15}(m' \textcircled{\cap} m'')$	32766	1–14	2.98–06	32767	1–15	1.13–06

Table 3: Hidden conflict between BFs Bel^i and Bel^{ii} from Example 6 and internal hidden conflict of Bel^{ii} and Bel^{iii} from Example 9, both on Ω_{16}.

6 Computational Complexity and Computations of Examples

Based on Definition 2 and Theorem 4, the complexity of computation of the degree of hidden conflict of two BFs Bel^i and Bel^{ii} is — on a general Ω_n — $O(n)$ of $\textcircled{\cap}$ operations. In the case of checking existence of a hidden conflict of the BFs we obtain the complexity $O(log_2(n))$ of $\textcircled{\cap}$ operations utilizing a simplification of computation based on $\textcircled{\cap}_{j=1}^{2k}(m^i \textcircled{\cap} m^{ii}) = \textcircled{\cap}_{j=1}^{k}(m^i \textcircled{\cap} m^{ii})$ $\textcircled{\cap}$ $\textcircled{\cap}_{j=1}^{k}(m^i \textcircled{\cap} m^{ii})$. Note that the complexity of $\textcircled{\cap}$ operation depends on the number and structure of focal elements.

During our analysis of hidden conflicts a series of example computations was performed on frames of discernment of cardinality from 5 to 16. A number of focal elements rapidly grows up to $|\mathcal{P}(\Omega)| = 2^{|\Omega|} - 1$ when conjunctive combination $\textcircled{\cap}$ is repeated, see e.g. 32766 and 32767 focal elements in the presented Examples 6 and 9 at Table 3. Because the degree of the hidden conflict and existence of the hidden conflict depends on the number and the structure of focal elements not on their bbms, we have used same bbms for all focal elements of a BF in our computations on frames of cardinality greater than 10.

All our experiments were performed in R [20] using R Studio [21]. We are currently developing an R package for dealing with belief functions on various frames of discernment. It is based on a relational database approach - nicely implemented in R, in package called data.table [22].

7 Several Important Remarks

We have to underline that hidden conflict of belief functions is not a new measure of conflict. It just improves/extends the classic measure defining the conflict by $m(\emptyset)$ in situations where $m(\emptyset) = 0$; it distinguishes fully non-conflicting BFs from those with a positive hidden conflict. This notion serves for deeper understanding of conflictness / non-conflictness, it enables to point out the conflict also in situations where conflicts had not been expected, in situations where $m_{\textcircled{\cap}}(\emptyset) = 0$; hence to point out and to help to understand the conflicts which are hidden due to $m_{\textcircled{\cap}}(\emptyset) = 0$.

Particular numeric values of a hidden conflict have no reasonable interpretation so far. For now, we are interested whether the value is zero (i.e. no conflict) or not.

Repeated applications of the conjunctive combination $\oplus$ of a BF with itself is used here to simulate situations where different independent believers have numerically the same bbm. Thus this has nothing to do with idempotent belief combination (where of course no conflict between two BFs is possible).

There is brand new idea of hidden conflicts in [19] and in this contribution. Brand new interpretation of $m(\emptyset)$ distinguishing fully non-conflicting BFs from those with hidden conflict. The assumption of non-conflictness when $m_{\textcircled{\cap}}(\emptyset) = 0$ was relaxed, due to observation of a qualitatively new phenomena — observation of hidden conflict even in the cases where $m_{\textcircled{\cap}}(\emptyset) = 0$. Both these studies want to point out the existence of hidden (auto-)conflicts in situations where no conflict was expected till now. Thus the definitions of hidden conflict and hidden auto-conflict[19] are not anything against the previous Daniel's research and results on conflict of belief functions e.g. [11, 2, 4]. Of course, some parts of the previous approaches should be updated to be fully consistent with the newly presented results on hidden conflicts and auto-conflicts.

Our study was motivated by investigation of conflict $Conf$ of BFs based on their non-conflicting parts [4], thus we were interested in independent BFs when hidden conflict was observed. But we have to note that conflictness / non-conflictness of BFs has nothing to do with dependence/independence of the BFs. Repeated computation of several (up to n) numerically identical BFs, when looking for hidden conflict is just a technical tool for computation of $m(\emptyset)$ or more precisely say for computation of $\kappa = \sum_{X \cap Y = \emptyset} m_j(X)m_j(Y)$. We are not interested in entire result of repeated application of $\textcircled{\cap}$, we are interested only in $m(\emptyset)$ or more precisely say in $\kappa = \sum_{X_1 \cap X_2 \cap \ldots \cap X_k = \emptyset} m_j(X_1)$ $m_j(X_2) ... m_j(X_K)$. Thus our computation has nothing to do with any idempotent combination of BFs. And we can compute hidden conflict using $\textcircled{\cap}_1^k$ (or κ) in the same way for both dependent and indepentent BFs. It is either not necessary to include any independence assumption to our Definitions 1 and 2.

8 Summary and Conclusion

Hidden conflicts of belief functions in situations where mutual intersections of any focal element of one BF with all focal element of the other BF are non-empty has been presented and analysed. There may be a positive conflict in situations, where sums of conflicting belief masses are empty, i.e. in situations which have been usually considered to be non-conflicting till now.

Several levels — degrees of hidden conflict were observed, maximal degree of hidden conflicts dependent on size of corresponding frame of discernment was found. A variety of hidden conflicts of degrees 1 – $(n-2)$ was described for an n-element frame of discernment. A necessary and sufficient condition for full non-conflictness of BFs in dependence on maximal cardinality of their focal elements has been specified and computational aspects analysed. Analogously to the evident conflicts, internal hidden conflicts are distinguished from the hidden conflicts between BFs.

This qualitatively new phenomenon of conflicts of BFs moves us to better understanding of nature of conflicts of belief functions in general and brings a challenge to elaborate and update existing approaches to conflicts of BFs.

This may consequently serve as a basis for better combination of conflicting belief functions and better interpretation of the results of belief combination whenever conflicting belief functions appear in real applications.

Acknowledgement

This work was supported by the Czech Science Foundation (project 16-12010S).

References

[1] Thomas Burger. Geometric views on conflicting mass functions: From distances to angles. *International Journal of Approximate Reasoning*, 70:36–50, 2016.

[2] Milan Daniel. Properties of plausibility conflict of belief functions. In L. Rutkowski, editor, *ICAISC 2013*, volume 7894 of *LNCS*, pages 235–246, Heidelberg, 2013. Springer.

[3] Milan Daniel and Jianbing Ma. Conflicts of belief functions: continuity and frame resizement. In U. Straccia and A. Calì, editors, *SUM 2014*, volume 8720 of *LNCS*, pages 106–119, Cham, 2014. Springer.

[4] Milan Daniel. Conflict between belief functions: a new measure based on their non-conflicting parts. In F. Cuzzolin, editor, *BELIEF 2014*, volume 8764 of *LNCS*, pages 321–330, Cham, 2014. Springer.

[5] Sébastien Destercke and Thomas Burger. Toward an axiomatic definition of conflict between belief functions. *IEEE transactions on cybernetics*, 43(2):585–596, 2013.

[6] Eric Lefèvre and Zied Elouedi. How to preserve the conflict as an alarm in the combination of belief functions? *Decision Support Systems*, 56:326–333, 2013.

[7] Weiru Liu. Analyzing the degree of conflict among belief functions. *Artificial Intelligence*, 170(11):909–924, 2006.

[8] Arnaud Martin. About conflict in the theory of belief functions. In *Belief Functions: Theory and Applications*, pages 161–168. Springer, Heidelberg, 2012.

[9] Johan Schubert. The internal conflict of a belief function. In *Belief Functions: Theory and Applications*, pages 169–177. Springer, Heidelberg, 2012.

[10] Philippe Smets. Analyzing the combination of conflicting belief functions. *Information fusion*, 8(4):387–412, 2007.

[11] Milan Daniel. Conflicts within and between belief functions. In Eyke Hüllermeier, Rudolf Kruse, and Frank Hoffmann, editors, *IPMU 2010*, volume 6178 of *LNCS*, pages 696–705. Springer, Berlin, Heidelberg, 2010.

[12] Milan Daniel. Non-conflicting and conflicting parts of belief functions. In *7th International symposium on Imprecise Probability: Theories and Applications (ISIPTA 2011)*, pages 149–158, Innsbruck, 2011. SIPTA.

[13] Glenn Shafer. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.

[14] Philippe Smets. Decision making in the tbm: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38(2):133–147, 2005.

[15] Barry R Cobb and Prakash P Shenoy. On the plausibility transformation method for translating belief function models to probability models. *International journal of approximate reasoning*, 41(3):314–330, 2006.

[16] Milan Daniel. Probabilistic transformations of belief functions. In L. Godo, editor, *ECSQARU 2005*, volume 3571 of *LNCS (LNAI)*, pages 539–551. Springer, Heidelberg, 2005.

[17] Milan Daniel and Václav Kratochvíl. Hidden conflict of belief functions (full text in preparation). http://www.cs.cas.cz/∼milan/for_download/hidden_conflict_ext.pdf, in preparation.

[18] Arnaud Martin, Anne-Laure Jousselme, and Christophe Osswald. Conflict measure for the discounting operation on belief functions. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.

[19] Milan Daniel and Václav Kratochvíl. Hidden auto-conflict in the theory of belief functions. In *Proceedings of the 20th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty*, pages 34–45, 2017.

[20] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2016.

[21] RStudio Team. *RStudio: Integrated Development Environment for R.* RStudio, Inc., Boston, MA, 2015.

[22] Matt Dowle and Arun Srinivasan. *data.table: Extension of ‘data.frame‘*, 2016. R package version 1.10.0.

Comparison of estimated preference relations between interval UTA and SMAA

Masahiro Inuiguchi[1], Yuji Mukai[1], Roman Słowiński[2] and Salvatore Greco [3]

[1] *Graduate School of Engineering Science, Osaka University*
Toyonaka, Osaka 560-8531, Japan
Japan
inuiguti@sys.es.osaka-u.ac.jp

[2] *Institute of Computing Science, Poznań University of Technology*
60-965 Poznań
Poland
roman.slowinski@cs.put.poznan.pl
& Systems Research Institute, Polish Academy of Sciences

[3] *Department of Economics and Business, University of Catania*
Corso Italia, 55, 95129 Catania
Italy
salgreco@unict.it
& Portsmouth Business School, University of Portsmouth

Abstract

Interval UTA method [1] has been proposed as an intermediate approach between the conventional and robust ordinal regression methods. In interval UTA method, a solid block being a subset of all instances of utility functions compatible with the provided preference information is used to estimate a preference relation on the set of alternatives. Therefore, if a preference relation holds in the interval UTA method for a pair of alternatives, it is expected to hold also for a large part of the set of all instances of compatible utility functions.

In this paper, we investigate whether strict preference relations obtained by the interval UTA are frequently confirmed by randomly chosen instances of compatible additive utility functions or not. To check this hypothesis, we apply Stochastic Multicriteria Acceptability Analysis (SMAA) [2] on the whole set of instances of compatible utility functions, and we compare the strict preference relations obtained by both methods. Several reasonable evaluation functions are considered to obtain interval utility function models to be tested. By a numerical experiment, we show that, except a few cases, the strict preference relations obtained by any of the considered interval utility function models frequently hold in SMAA. However, this is not well confirmed for any single interval utility function model. Based on the results of the numerical experiment, we propose a strategy for a better application of the interval UTA method using two evaluation functions so that most of the estimated strict preference relations frequently hold for randomly chosen instances of compatible additive utility functions.

Acknowledgement

This work is supported by JSPS KAKENHI Grant Number JP17K18952.

References

[1] Inuiguchi, M., Okumura, A., Słowiński, R., Greco, S.: Semi-robust multiple criteria decision aiding by interval UTA method. CD-ROM Proceedings of the 12th International Conference on Modeling Decisions for Artificial Intelligence, Skövde, Sweden, pp.123–136 (2015)

[2] Tervonen, T., Figueira, J.: A survey on stochastic multicriteria acceptability analysis methods, Journal of Multi-Criteria Decision Analysis 15, 1–14 (2008)

An Application of Revealed Preference for Network Flow in Sequential Decision Model

Puchit SARIDDICHAINUNTA[1] and Masahiro INUIGUCHI[2]

Graduate School of Engineering Science, Osaka University,
1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan
[1] puchit@inulab.sys.es.osaka-u.ac.jp

[2] inuiguti@sys.es.osaka-u.ac.jp

In this research, we study the aggregation of preference data of the users in network flow model. These users can be referred to followers in the sequential decision model in the our sequential decision problem. We transform the demand data into a bundle of linear constraints to represent the strategic norms of the followers in the centralized decision model. We show that the revealed preference theory is a useful foundation to construct such a multiple users' rational actions without losing much the richness of preference by our simplification. Although the aggregate preference become a set of ambiguous decision norms of the representative follower, in our case the convex polyhedron, we can apply this framework to the bilevel optimization and formulate this sequential decision problem as a minimax problem which is the centralized decision problem of the leader in our sequential decision model.

Keywords: Minimax Problem, Sequential Decision Model, Revealed Preference

Visual method to compare shuffling algorithms

Ondrej DOLEZAL and Hana TOMASKOVA

University of Hradec Kralove, Faculty of Informatics and Management
Rokitanskeho 62, Hradec Kralove
Czech Republic

Abstract

Aim of this paper is to present method of evaluation and comparison of algorithms for shuffling. unlike for sorting algorithms, there is no simple way to check if algorithm produce truly random sequence which is an issue this paper is dealing with. Our method is purely visual and does not require statistical analysis, although it can be used a secondary tool to verify conclusions. We are using Monte Carlo simulation method in order to generate data for our analysis. four well known shuffling algorithms are described and used for calibration and testing of our method. results of all four algorithms are presented and discussed. Figures and examples are used in order to demonstrate strengths and weaknesses of each of those algorithms. For a better idea, instead of general form, we are using example of standard playing cards deck as a base for the whole paper.

Keywords: Shuffling, Comparison, Algorithm, Simulation, Playing cards

1 Introduction

As the usual discussion is all about sorting algorithms and their properties, we decided to take a different direction and focus on very opposite, shuffling or mixing algorithms. It is simple to validate results of sorting algorithm since you just have to check if the current value in finite sequence is smaller or the same as next value (for descending order). But question of validation of sorting algorithm is much more complex. Also benefits and use cases of sorting are easy to see all around us, but one could ask, what is the mixing good for. We decided rather than describe whole topic in general manner, to focus on one of notorious use cases of shuffling which is shuffling of playing cards. therefore, from now on, the paper will focus on playing cards shuffling, although every algorithm or method describe in this paper can be generalized to optional finite sequence of ordinal values.

People usually have a problem when their activity is meant to be completely random. This is due to the tendency to create patterns of behavior that can be predicted. One of the experiments [1], which confirms this hypothesis, is when one can see by the naked eye, if the "random" sequence of 0s and 1s was generated by person or computer.

The number of permutations (possible combinations) of cards in the standard playing deck is 52! (Approximately 8×10^{67}). However, despite this immense number of combinations, it is possible to determine the probable state of the deck if cards were shuffled in inappropriate way or with not enough repetitions.

Diaconis [2] and Bayer [3] are dealing with different ways of mixing cards and their mathematical generalizations. Riffle shuffle, which is widely used around the world, has been described most thoroughly. In addition to the statistical features, the recommended number of repetitions of this shuffling technique has been set so that the resulting cards distribution is unpredictable, which is extremely important in the entertainment industry. Based on statistical models, seven repetitions were set as a value sufficient to randomly order entire deck. With six or fewer repetitions, it would be possible to predict the positions of certain cards. And with more than seven repetitions, the increase in randomness was negligible.

Result of this research has become a gold standard for casinos around the world, but the general public has little awareness of these properties, and so it happens that business and private tournaments are threatened by people who use this knowledge to manipulate odds in card game in their favour.

2 Methodology

In this section, a method for evaluating randomness of cards distribution after repeated shuffles, will be presented. In the second subsection, the Monte Carlo simulation method is briefly described.

2.1 Evaluation methodology

The very first question is how to find that 52 cards are shuffled into random order. The problem is that each of the 52! Permutations should have the same probability to occur. Thus, it is not possible to determine from one observation whether the deck is in random order, or biased in some way. For example, an extreme case where the deck is perfectly sorted and after one shuffle, the deck is in the same order with the exception of just two cards that have swapped positions. One would certainly argue that mixing was not satisfactory because the deck is almost the same as it was before. However, from a probability perspective, this could be absolutely acceptable. Therefore, it is necessary to repeat the shuffling and investigate long-terms patterns of card positions. therefore we have to focus on algorithm itself, rather than similarities between two consecutive states of the deck.

Our method is based on the frequency of occurrence position of every single card in the deck. Just as every permutation has the same chance to occur, each card must have the same chance of being placed at any of the 52 positions in the deck. Thus good algorithm would show no correlation of cards with a certain positions. It is therefore possible to test quality of algorithm without checking every possible permutation.

The method itself works as follows. The ideal frequency of a card at certain position is arithmetic average of $n = 52/s$ where s is the number of deck shuffling processes. Processes are intentionally used, because shuffling itself can be repeated several times during one process, as it is in the real world. Thus, n is the number of times each card should appear in each position if the card distribution is completely random.

For better visualization, a grid where the rows are the individual cards, and the columns are positions on which the cards are placed, was created. The cells are then dyed with red and blue shades. Red color indicates a lower frequency than n. Blue indicates a higher frequency than n. Lighter shades mean a value that is closer to n. Darker shades mean a value that is more further from n. For red fields, the value range i from 0 to n. For blue fields, it is from $n + 1$ to $2n$.

We still have to mention values $x > 2n$ which is more precisely $s - 2n$. Cells with this value are dark blue and show a strong positive correlation between a card and specific position.

if we have an algorithm that will generate a perfectly random sequence. The grid will look like the light blue and light red cells are represented in the same amount and their position should not form any pattern. Otherwise, the darker shades and patterns of cells with similar colors will appear in the grid.

Thanks to this principle of visualization of results, the quality of the algorithm can be seen at first glance. When changing parameters at the riffle shuffle or overhand shuffle, you can also effectively watch progression in time based on which the recommendations were formed for both.

2.2 Monte Carlo

The Monte Carlo simulation method is used to generate data for visual validation. In general, this method serves to find a mean value of a certain random event. In our case, it is a card position after shuffling, and the mean value is the number of shuffles when the card ended in the very same place. Each shuffling is repeated 100,000 time. this value has been chosen because the time required to complete the simulation is acceptable, and also because the major change in visual distribution has not been observed when more repeats were performed.

3 Shuffling

Four methods of shuffling will be briefly presented in this section. two methods are rather pure algorithmic solution useful for general tasks. Other two are common methods for shuffling playing cards, described in [4, 5, 6].

3.1 Simple swap

This method was chosen rather to illustrate how the algorithm should not work. But this is arguably one of the first algorithms that ordinary person would come up with. At first glance, the algorithm seems to be so simple that it is not possible it does not work. The algorithm itself will be described first and then its weaknesses.

1. Select one random cards from a deck.
2. Swap chosen card with first untouched card.
3. Repeat steps 1. and 2. until there is no untouched cards left in the deck.

The problem is that this method creates a seemingly evenly mixed order, but after multiple shuffling, algorithm reveals its imperfections. The flaw of this approach is that the cards are not equally likely to be picked for swap. This results in a visible visual correlation of certain cards to certain positions. This, of course, is an undesirable effect, so the algorithm that seems to be successful at first sight is discarded as inappropriate by simulation. The same conclusions can also be made by statistical analysis, but that is not goal of this paper.

```
for (i = 0; i < 51; i++)
            {
                j = random(0, 52)
                swap(deck[i], deck[j])
            }
```

Time complexity big O of this approach is $\mathcal{O}(n)$.

3.2 Fisher-Yeates

This is an algorithm, although very similar to previous one, is designed to generate a random final sequence of ordinal values. In our case, the sequence of 52 values (cards). This is a very effective algorithm that ensures that each of the possible permutations is equally probable.

```
for (i = 0; i < 51; i++)
            {
                j = random(i, 52)
                swap(deck[i], deck[j])
            }
```

Time complexity of algorithm big O is $\mathcal{O}(n)$.

3.3 Riffle shuffle

Riffle shuffle is a professional style of shuffling cards that is also used by general public. Unlike the Fisher-Yates, this method is only associated with card shuffling as it would be highly inefficient to use this algorithm for general purposes.[7]

1. The deck is divided into two approximately equal halves.
2. Both halves are hold side by side.
3. By slide of hands, cards are randomly left to fall. Cards are falling randomly from both hands and forming new pile in the process.
4. After the last card is used, the deck is complete and the process can be repeated.

Throughout the shuffling process, there are several elements of chance. Splitting of the deck into two halves is described by a binomial distribution with a parameter

$$p(x) = \frac{1}{2^{52}}\binom{52}{x}, \tag{1}$$

where x is the number of cards for which we want to calculate the probability of being next. Another element is the half from which the cards are falling into a newly formed pile. The last variable is how many cards from one half are moved. That amount is based on probability

$$p(A) = \frac{a}{a+b}, \tag{2}$$

$$p(B) = 1 - p(A), \tag{3}$$

where A, B are both halves of the deck and a, b are current amounts of cards within. Which means that there is a higher probability that card from hand with more cards will be next.

```
current1 = a-1
current2 = 51

for (i = 0; i < 52; i++)
            {
                if (random < (a / (a + b)))
                {
                    deckNew[i] = deck[current1]
                    current1 = current1 - 1
                }
                else
                {
                    deckNew[i] = deck[current2]
                    current2 = current2 - 1
                }
            }
```

Time complexity big O is also $\mathcal{O}(n)$.

3.4 Overhand shuffle

Overhand shuffle is a style of mixing cards, which is widely used among general public, and is also one of the easiest to perform. whole goal is to repeatedly reallocate uninterrupted group of cards from top to bottom.

1. Remove a random number of cards from the top of the deck.
2. Put these cards in the second hand so they are stacked at the top of the new pile.
3. Repeat the previous two steps while there are some cards remaining at first hand.

With this mixing style, the only parameter is, how many cards are removed at one time from the top of the deck. It would be a mistake to assume the same number of cards would always be taken during one pass, neither could we assume that each individual person grabs the same number of cards.

```
current = 0

for (i = 0; i < 52; )
             {
                 x = random(min, max)
                 current = current + x + 1

                 if (current > 52) current = 52

                 for (k = 0; k <= x; k++)
                 {
                     j = 52 - current + k
                     deckNew[j] = deck[i]
```

```
                              i = i + 1
                         }
                    }
```

Time complexity of this algorithm big O is also $\mathcal{O}(n)$.

4 Results

Now the results of the individual algorithms, together with their graphical evaluations will be presented.

4.1 Simple swap

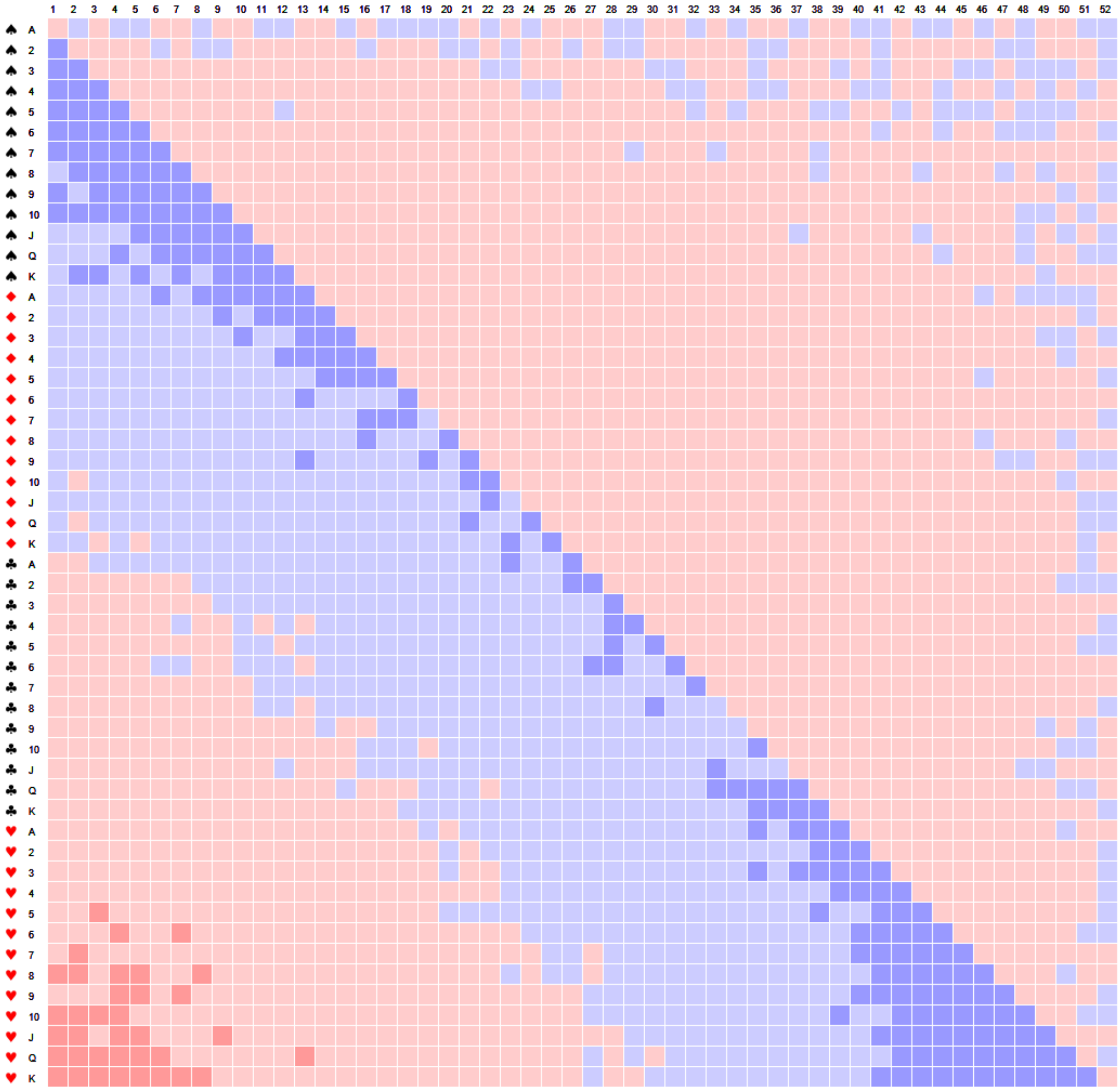


Figure 1: Visual result of the simple swap algorithm.

From results Fig.1 is clear that simple swap algorithm generate a pattern. From the color scheme, it is easy to conclude that certain cards are strongly correlated (positively or negatively) with certain positions. The assumption that this mixing method does not generate random distribution was confirmed, and therefore is not suitable for mixing cards or for general randomization.

4.2 Fisher-Yeates

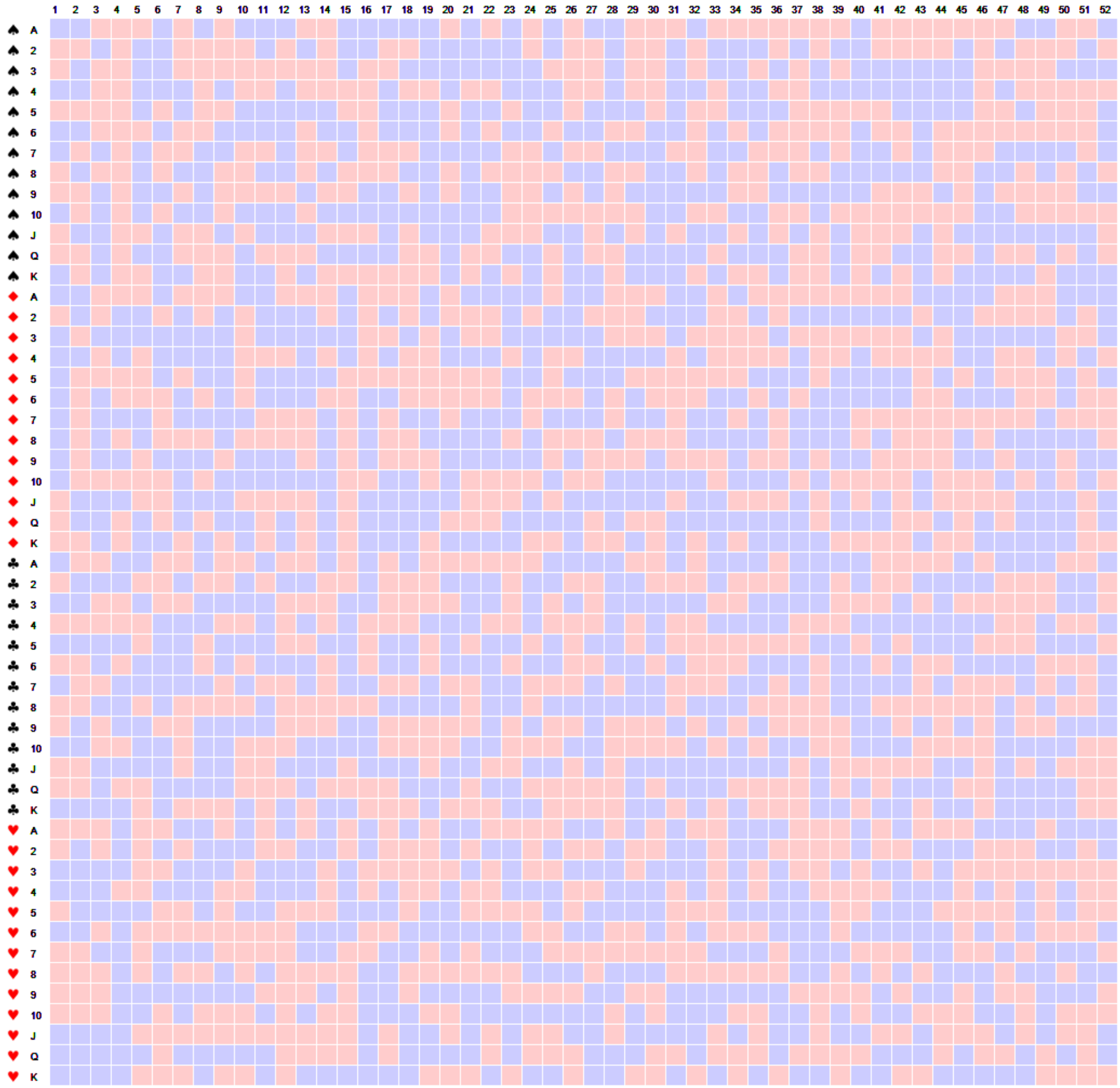


Figure 2: Visual results of Fisher-Yeates algorithm.

As expected, it is clear from Fig.2 that Fisher-Yates has flawless results. Relative occurrences are evenly distributed. Although this result is not surprising, the implementation of this algorithm contributed to verifying the functionality of the methodology that was created.

4.3 Riffle shuffle

Based on Fig.3 is obvious that in order to achieve the same satisfying result as Fisher-Yates, it is necessary to repeat the riffle shuffle seven times. The results of the research from the Diaconis [2] were confirmed by this simulation method.

4.4 Overhand shuffle

There is no research on overhand shuffle that would clearly specify shuffling parameters. so these parameters and their values had to be estimated. With this method, it is not necessary to divide the deck into two halves, so the only identified parameter is the number of cards that are moved in one shuffle. The following scenarios are purely hypothetical.

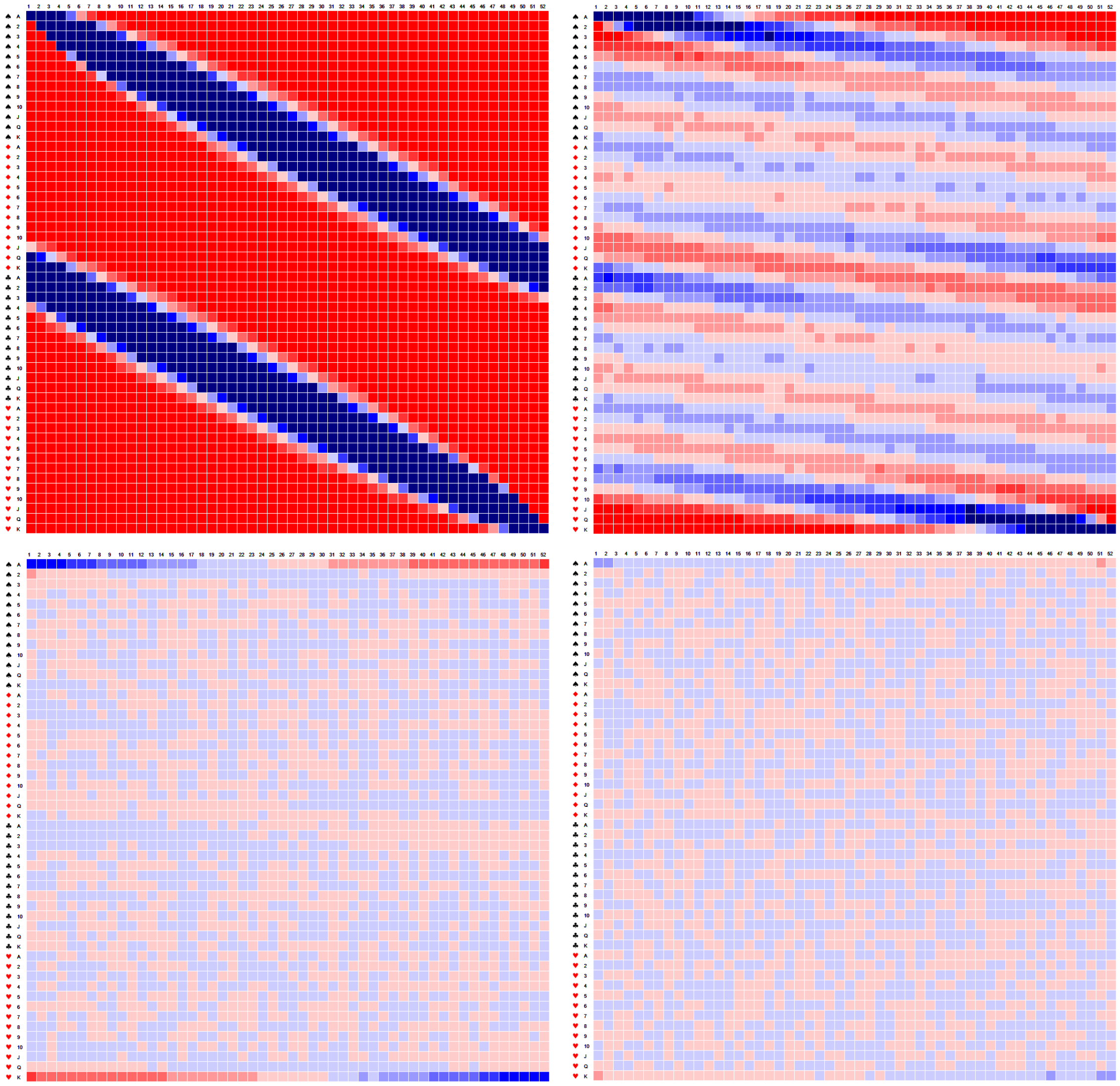


Figure 3: Visual results of riffle shuffle after 1, 3, 5 and 7 repetitions respectively.

4.4.1 Scenario A:

Fig.4 Person grabs fewer cards in one move. This quantity is given by uniform distribution with a minimum and a maximum of 4 and 13 cards.

4.4.2 Scenario B:

Fig.5 Person grabs a larger amount of cards in one move. This quantity is again given by uniform distribution with a minimum and a maximum value of 10 and 20 cards.

4.4.3 Comparison:

In scenario A, 80 repetitions were needed to achieve a result that could be considered satisfactory, although it is still possible to find a certain bias in the distribution of the cards. For scenario B, however, we only needed 30 repetitions to achieve results that meet our requirements for random distribution.

The first conclusion is that it is better to grab cards in larger portion, which also contributes to a faster shuffling.

But the second conclusion is that this shuffling method is less efficient than the previous riffle shuffle, where only seven repetitions were enough to achieve the same result.

5 Discussion

Selected techniques are just few among many. These four were selected because they were well described and therefore suitable for calibrating our method that is now robust enough to test other shuffling methods that could offer more surprising results. Other algorithms include Hindu shuffle, Pile shuffle, Corgi shuffle,r Mongean shuffle and more.

Although the Overhand shuffle has only one parameter, it would be possible to create more real life scenarios. The number of cards removed is certainly not evenly distributed. People in one shuffle pass can change the number of cards without restrictions. Occasionally, cards may not fall on top of the newly created deck, or not all the cards are removed from the top of the deck.

Although the results are overall sufficient the method of validation is not without its flaws. For more accurate analysis, it would be necessary to implement the validation of a large number of different patterns that could not be easily spotted by this visual method. Such patterns may include, correlation of two or more cards in regards to their positions.

6 Conclusion

Article presented an issue of random distribution of the playing cards in a deck. Four common algorithms were selected and subjected to a test that should verify if those algorithms truly generate random distribution of the cards in the deck. For that purpose was presented a simple visual method to detect correlation of cards with certain positions that could easily disqualify tested algorithm. same technique can be used with other mixing algorithms as well.

Acknowledgement
Financial support of the project TACR ETA TL01000300 is gratefully acknowledged.

References

[1] Malgorzata Figurska, Maciej Stalczyk, and Kamil Kulesza. Humans cannot consciously generate random numbers sequences: Polemic study. *Medical Hypotheses*, 70(1):182 – 185, 2008.

[2] Persi Diaconis. Mathematical developments from the analysis of riffle-shuffling. In *Groups, Combinatorics and Geometry*, pages 73–97. World Scientific, 2003.

[3] Dave Bayer and Persi Diaconis. Trailing the dovetail shuffle to its lair. *Ann. Appl. Probab.*, 2(2):294–313, 05 1992.

[4] Persi Diaconis. Mathematical developments from the analysis of riffle shuffling. 2002.

[5] Richard Durstenfeld. Algorithm 235: Random permutation. *Commun. ACM*, 7(7):420–, July 1964.

[6] Robin Pemantle. Randomization time for the overhand shuffle. *Journal of Theoretical Probability*, 2(1):37–49, 1989.

[7] Persi Diaconis, Jason Fulman, and Susan Holmes. Analysis of casino shelf shuffling machines. *Ann. Appl. Probab.*, 23(4):1692–1720, 08 2013.

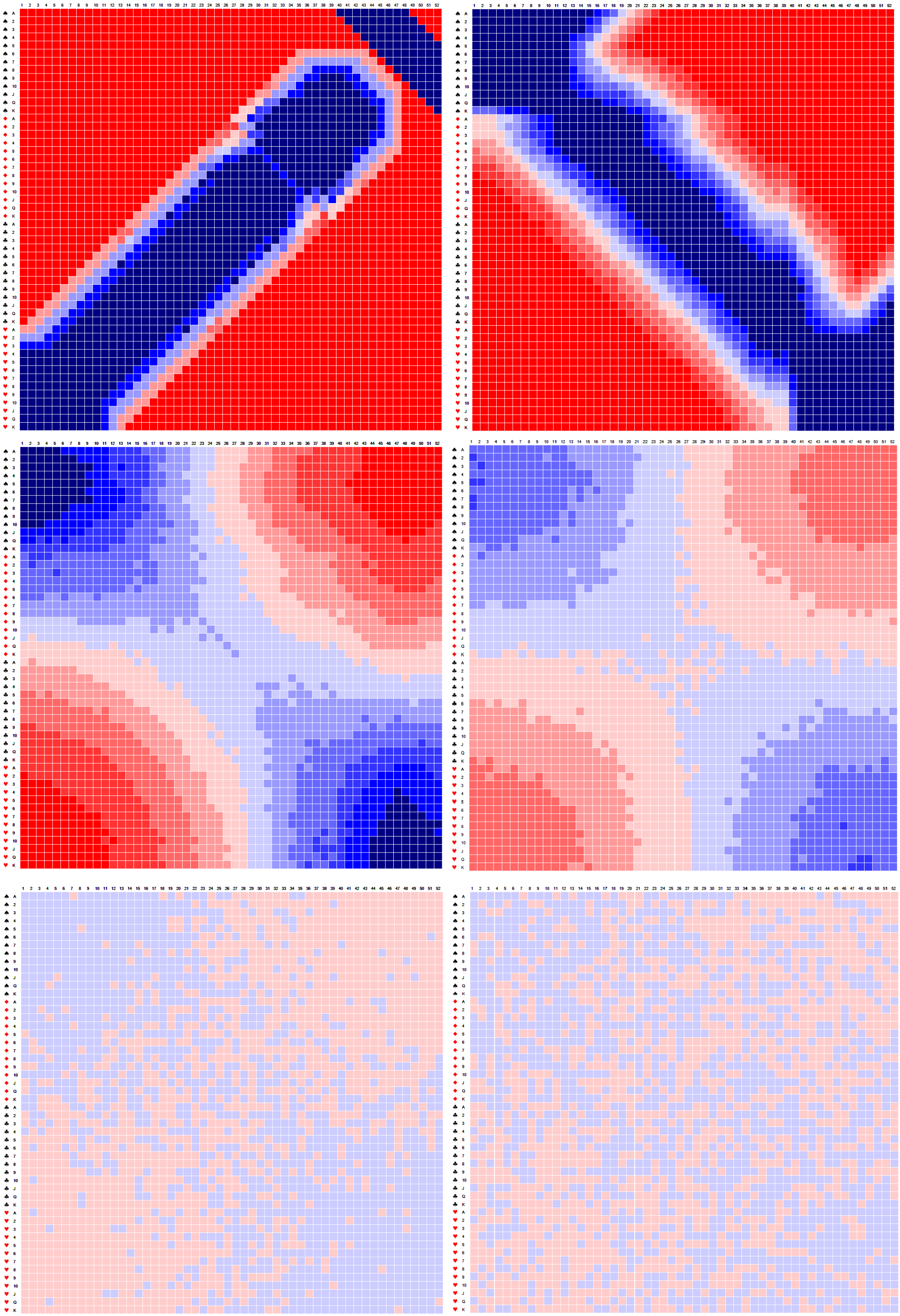


Figure 4: Visual results of overhand shuffle scenario A after 1, 2, 10, 20, 60 and 80 repetitions respectively.

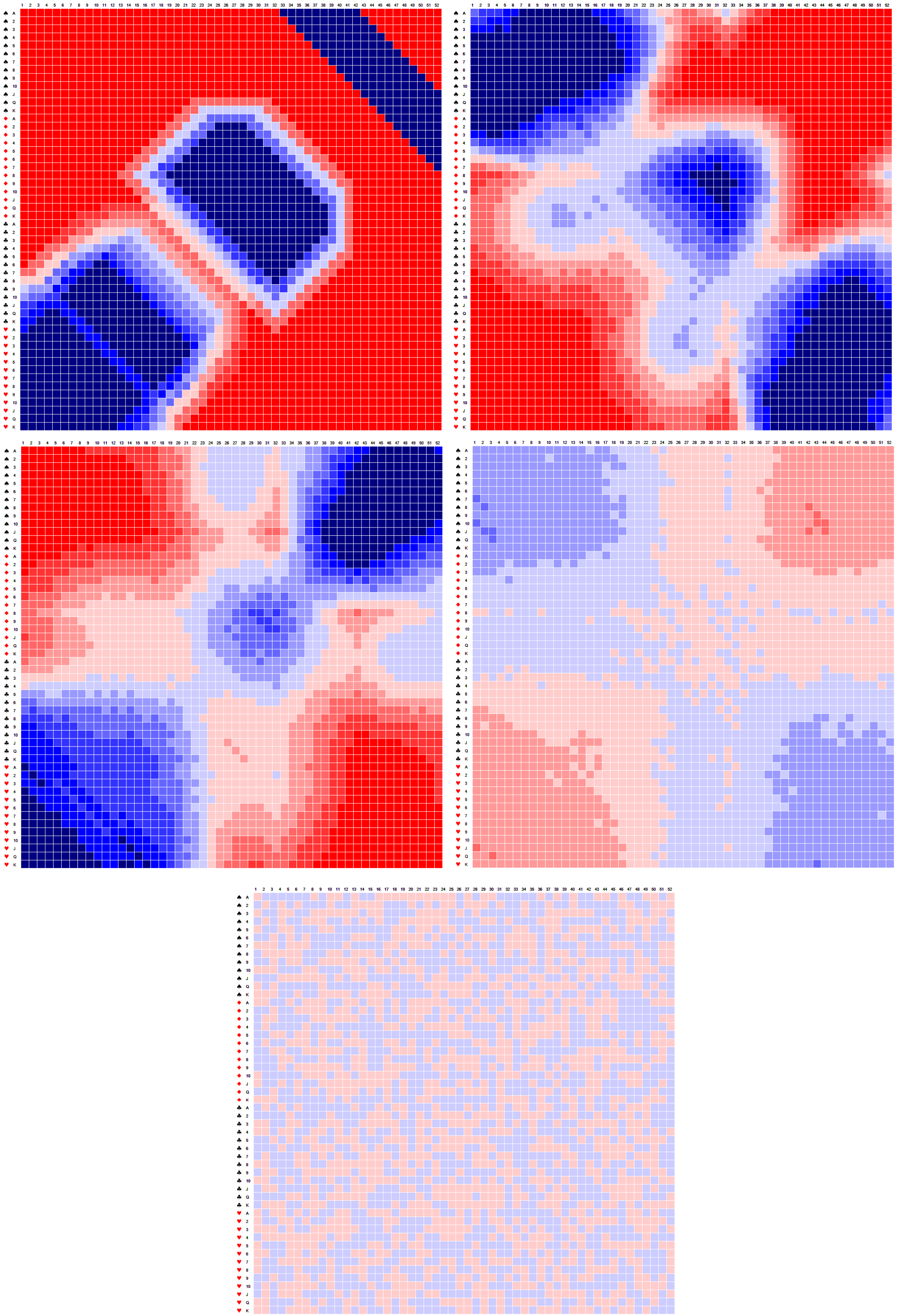


Figure 5: Visual results of overhand shuffle scenario B after 1, 2, 3, 10 and 30 repetitions respectively.

Accuracy Score of Convergence of Estimated Interval Weights in Interval AHP

Issei Torisu[1] and Masahiro Inuiguchi[2]

[1] *Graduate School of Engineering Science, Osaka University*
1-3, Machikaneyamacho, Toyonaka-shi, Osaka, 560-8531, Japan
torisu@inulab.sys.es.osaka-u.ac.jp

[2] *Graduate School of Engineering Science, Osaka University*
1-3, Machikaneyamacho, Toyonaka-shi, Osaka, 560-8531, Japan
inuiguti@sys.es.osaka-u.ac.jp

Abstract

Interval AHP [1] is proposed to estimate interval weights from a pairwise comparison matrix for representing the vagueness of human evaluation. Several interval weight estimation methods have been proposed in order to improve the estimation quality [2]. As the desirable properties of the interval priority estimation method, (1) the observed pairwise comparison matrices should be realizable under the estimated interval priority weights, (2) the correct crisp priority weight vector is estimated from a perfectly consistent comparison matrix, and (3) the estimated interval priority weight vector satisfies the normalization condition, have been considered so far.

In this paper, as an additional desirable property for an interval priority weight estimation method, we consider the increasing convergence of the quality of estimated interval priority weights with respect to the number of observations of the pairwise comparison matrix. This convergence implies that the quality of the estimated interval priority weights is improved and converges to a certain value as the preference information obtained from the decision maker increases.

By a numerical experiment, we show that previously proposed interval priority weight estimation methods do not possess this convergence property although the initial estimation quality is sufficiently good. Then we propose several interval priority weight estimation methods having this convergence property as well as a sufficiently good initial estimation quality.

Acknowledgement
This work is supported by JSPS KAKENHI Grant Number JP17K18952.

References

[1] K. Sugihara and H. Tanaka. Interval evaluations in the analytic hierarchy process by possibility analysis. *Computational Intelligence*, vol. 17, no. 3, pp. 567–579, 2001.

[2] M. Inuiguchi and S. Innan. Non-parametric interval weight estimation methods from a crisp pairwise comparison matrix. In *Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems*, #159, 2017.

Withdrawn Paper

Withdrawn Paper

Withdrawn Paper

Withdrawn Paper

Numerical solution of Black-Scholes models with the help of F-transform

Michal HOLČAPEK and Radek VALÁŠEK

Institute for Research and Applications of Fuzzy Modelling, NSC IT4Innovations,
30. dubna 22, 701 03 Ostrava 1,
Czech Republic
{Michal.Holcapek,Radek.Valasek}@osu.cz

Option pricing is an important part of market practice. In literature, one can find several models and methodologies for option pricing, e.g., the Black-Scholes models, Heston or Lévy model. It is well-known that no analytic solution can be found for certain classes of more complex models; therefore, instead of an exact solution one seeks an approximate solution by means of classical numerical methods such as finite difference or element method, Monte Carlo method or modern methods like the discontinuous Galerkin method (see, e.g., [1, 2]) or Galerkin methods with wavelet bases (see, e.g., [3, 4, 5]).

Among modern approaches to the numerical solution of (partial) differential equations one can also include finite difference and finite element based methods that use the fuzzy transform technique in their designs (see, e.g., [6, 7, 8, 9, 10]). Recall that the *fuzzy transform* (*F-transform*, for short) was introduce by Perfilieva in [6] (see also [11]) as a soft computing method that is used for the approximation of functions. The F-transform has two phases: *direct* and *inverse*. The direct F-transform transforms a bounded continuous (integrable) real function with the help of fuzzy sets of a fuzzy partition of a given space (e.g., an interval or a product of intervals) to a finite vector of real numbers that are called the components of F-transform. The inverse F-transform sends the latter vector back to a continuous function as the result that approximates the original one.

In [12], we proposed a solution of the Black-Scholes model with one asset by a numerical method, which combines the finite difference method and the F-transform technique. Note that the one-asset Black-Scholes model is a partial differential equation (PDE) describing a single plain vanilla option pricing problem. If σ denotes a constant volatility of the underlying asset S and r is a constant risk-free interest rate, then the price of the option $V(S,t)$ is obtained by solving the following backward PDE (see, e.g., [13]):

$$\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rS\frac{\partial V}{\partial S} + rV = 0, \tag{1}$$

where $V : (0, S^{\max}) \times (0,T) \to \mathbb{R}$, which satisfies the following boundary and initial conditions:

$$\begin{aligned} V(0,t) = V(S^{\max},t) &= h(S), \quad t \in (0,T), \\ V(S,0) &= g(S), \quad S \in (0, S^{\max}). \end{aligned}$$

Our proposed method has several steps. First, we transform (1) to a homogeneous PDE. Further, we use the Crank-Nicolson scheme for time discretization, which results in a system of (ordinary) differential equations. Finally, we apply the F-transform on the left and right side[1] of each differential equation and express the F-transform components of derivatives as the differences of F-transform components of the unknown function, which is the solution of the homogeneous PDE. An approximative solution of the homogeneous PDE is obtained as the inverse F-transform of the solution of the given system of algebraic equations. The approximate solution of (1) is given by the inverse transformation to that which makes (1) homogeneous.

Although, the idea of the method outlined above is clear and the obtained results are competitive with results of the standard numerical methods, the application of F-transform on the left side of differential equations has one drawback. Since the F-transform of the product of functions is not the product of their F-transforms in general, before the application of the F-transform on the left side of the differential equations, we have to use a trick and replace the coefficients of PDE (e.g.

[1] Later we will discuss the application of the F-transform on the left side of differential equations.

$\frac{1}{2}\sigma^2 S^2$) by their values at the nodes (e.g. $\frac{1}{S}\sigma^2 S_i^2$) over which a fuzzy partition of a given space is considered. Therefore, the approximate solution of PDE by our proposed method is not found in an approximate space by solving of equations that are truly the F-transforms of the original differential equations, but rather the F-transforms of differential equations with discretized coefficients. Note that the original attempts to apply the F-transform technique in solving of ordinary and partial differential equations consider the constant coefficients (see, e.g., [6, 7]).

The aim of our talk is to present a novel approach in solving PDE, especially, the Black-Scholes models, trying to overcome the partial discretization of coefficients in the original approach that combines the finite difference method and the F-transform technique with the help of variational calculus. We compare the results of the previous method and the novel one.

Acknowledgement
This work was supported by the project LQ1602 IT4Innovations excellence in science. The additional support was provided by the Czech Science Foundation through the project of No.16-09541S.

References

[1] J. Hozman and T. Tichý. A discontinuous Galerkin method for pricing of two-asset options. In *33rd International Conference Mathematical Methods in Economics*, Cheb, ZČU Plzeň, pp. 273–278, 2015.

[2] J. Hozman and T. Tichý. DG framework for pricing European options under one-factor stochastic volatility models. *Journal of Computational and applied mathematics*, 344:585-600, 2018.

[3] D. Černá. Numerical solution of the Black-Scholes equation using cubic spline wavelets. In *AIP Conference Proceedings*, volume 1789, 2016.

[4] D. Černá. Adaptive wavelet method for pricing two-asset Asian options with floating strike. In *AIP Conference Proceedings*, volume 1910, 2017.

[5] D. Černá and V. Finěk. Valuation of options under Heston stochastic volatility model using wavelets. In *Proceedings - 2017 4th International Conference on Mathematics and Computers in Sciences and in Industry, MCSI 2017*, pages 16–20, 2018.

[6] I. Perfilieva. Fuzzy transform: Application to the reef growth problem. In R.V. Demicco and G.J. Klir, editors, *Fuzzy Logic in Geology*, chapter 9, pages 275–300. Academic Press, Amsterdam, 2003.

[7] M. Štěpnička and R. Valášek. Fuzzy transforms and their application to wave equation. *J. Electr. Eng.*, 55(12):7–10, 2004.

[8] M. Štěpnička and R. Valášek. Numerical solution of partial differential equations with help of fuzzy transform. In *Proc. of the 2005 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2015)*, pages 1153–1162, 2005.

[9] W. Chen and Y. Schen. Approximate solution for a class of second-order ordinary differential equations by the fuzzy transform. *Intelligent and Fuzzy Systems*, 27:73–82, 2014.

[10] A. Khastan, I. Perfilieva, and Z. Alijani. A new fuzzy approximation method to Cauchy problems by fuzzy transform. *Fuzzy sets syst.*, 288:75–89, 2016.

[11] I. Perfilieva. Fuzzy transforms: Theory and applications. *Fuzzy sets syst.*, 157(8):993–1023, 2006.

[12] M. Holčapek and R. Valášek. F-transform based numerical solution of partial differential equations. *Proc. of the 20th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty*, pages 54–64, 2017.

[13] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–659, 1973.

Proposal of probability risk Evaluation for System Development Project Based on Requirements Analysis and Bayesian estimation

Shinji Mochida[1]

[1] *University of Marketing and Distribution Sciences*
Shinji_Mochida@red.umds.ac.jp
Japan

Abstract

All companies employ business systems. Additionally in most business system development projects are planned in order to reduce amount of running cost or increase benefit. Thus more efficient project management is needed in order to meet schedules and reduce cost. Although there are several stages in business system development projects. There are some factors that cause over cost or schedule delay of projects. Especially proper requirements from customer is the most essential in project managements. Also accuracy of estimates is essential. Although there are plus and minus risks in project management. So it is needed to take into account minus risks. Thus this paper proposes the method to calculate probability of accuracy of estimates and considering plus and minus risks with Bayesian estimation in project management.

Keywords: Risk Evaluation System, Development Project,Requirements Analysis,Bayesian estimation.

1 Introduction

Business System development projects are challenging in that are many requirements demanded from customers even while these requirements are proposed with the same priority. Thus it is important to narrow down and prioritize requirements according to their essentiality and criticality to finish on schedule. Although system developers estimate according to the complexity of projects [2], but customers expect the cost to be based on the number of requirements they propose. Then customers and system developers estimate differently, there are often **conflicting estimates**. Thus, this paper proposes cost share rate for business system development projects based on requirements analysis in order to estimate accurately. Cost share rate is defined as the percentage of total cost assigned to each requirement.Cost share rate could distinguish essential requirements. And requirements that has large cost share rate must have large risk, also should be under strict control. Because big change or modification for essential requirement give large impact to costs or schedule. Then this paper aim to propose a method to identify essential requirements in order to estimate accurately. And there are plus risks and minus risks in project. Although usually minus risks are taken in to account.Then risks are assigned to each requirements with cost share rate.Cost share rates show accuracy of estimate and probability to finish under budget and on schedule. This research shows potential for predicting cost share rate accurately using Bayesian estimation considering plus and minus risks. One purpose of this paper is showing that risk management could contribute reducing cost of business system development project. Additionally this paper demonstrates the potential to evaluate risk by requirements analysis for business system development project management with Bayesian estimation. This research is not focused on estimating costs based on the method to measure amount of source code of system, but rather allocating costs to each requirement. Previous research typically focused on either schedule, cost estimation or productivity. Improving productivity contribute to finish project fine[3]. About changing requirement in mechanical engineering design, one example of requirements analysis research explored the ability of predicting requirements change through graphical models of the requirements documents and historical change trends[4]. There are plus and minus risks in system development project. although there are no research that refers subjective factor. This paper propose the method take into subjective factor in account with Bayesian estimation.

2 Project management and Risk management

2.1 Project management and Risk management

In project management there are two important methods[5]. One is schedule management, and another one is risk management.On the other hand Equation 1 shows there are controllable factors or uncontrollable factors in business. Sale is uncontrollable factor, because sale come under the influence of markets, customers. Although cost is controllable factor, cost includes payment,material costs, for example, payment for staff could be cut off by manager. It is important to identify which factor is controllable and uncontrollable. And how appropriately controllable factor could be controlled. Also there is possibility uncontrollable factor could be controlled with Bayesian analyses.Usually extra budget is settled aside for refinement or fixing trouble in project management. This extra budget is called for contingency budget or only contingency. If risk management would work well,contingency budget would not be used, then contingency budget would come to benefit. At the result prospect of profit would increase. Thus risk management could contribute reducing contingency cost,And risk management has potential to increase contingency profit.

$$Gaining = Sale - Cost \tag{1}$$

2.2 Risk analysis

Risk is defined as factors that make uncertain when they will achieve their objectives under ISO31000. Usually risk Analysis is started from risk identification in risk management. Then, risk evaluation is considered by qualitative evaluation and quantitative evaluation. Thus quantitative risk analysis is calculated by possibility × cost. This calculated risk(cost) should be spend ,if risks comes up. It is called expected monetary value. Risks is evaluated by expected monetary value Equation 2. And risks are prioritized by the order of expected monetary value. In quantitative evaluation for risk management usually probability is given subjectively by staffs subjectively. Or probability is given by experts, Delphi method or questionnaire for skilled staffs.Thus getting accurate probability is very essential to cucullate expected monetary value correctly. Thus this paper takes into account β distribution to calculate probability,and expect monetary value. This paper propose conditional possibility in order to cucullate risk correctly. It is very essential point in risk management there are plus and minus risks, in addition probability is subjective;(see Figure 1).In this research conditional possibility is given to essential risks. In many project risk comes up by misunderstanding requirements or defects in requirements from clients in business system development project. It is vital to get certain requirements and predict risk in requirements properly.

$$Risk(Expected\ Monetary\ Cost) = Probability \times Cost \tag{2}$$

3 Cost Prediction Methods for Business System Development

3.1 Cost Prediction Methods

Proper estimate is essential to finish projects on schedule and under budget. Over cost or schedule delay is caused by missing estimate. Usually amount of program source code is predicted by some prediction method in order to estimate in business system development project. Then amount of program source code is converted into base monetary cost. Next total cost is made by adding contingency cost to base monetary cost. It is finally budget for project. Then this section explains some current methods to predict costs of business system development projects. There is typical

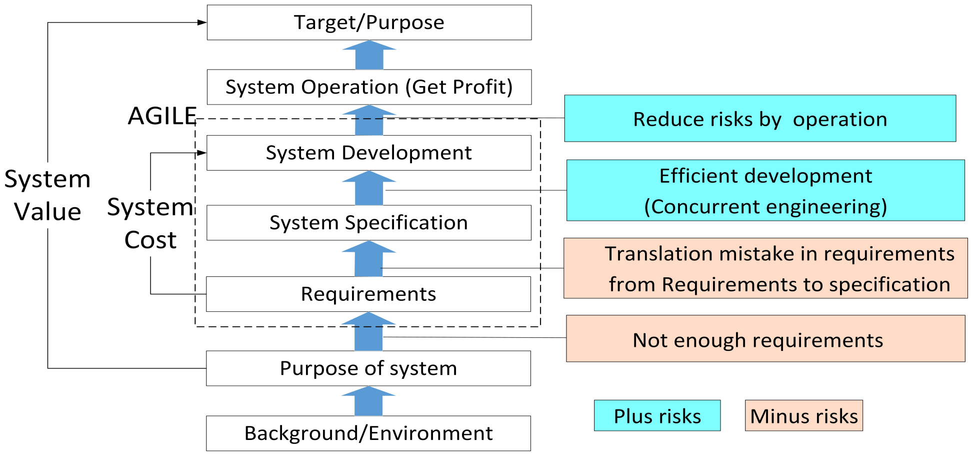


Figure 1: Plus and Minus risks in system development project

methods to estimate for business system,typical methods;COCOMO method and Function Point method. Both methods predict costs for business system. In case of COCOMO method, it estimate by amount of program source code.On the other hand In case of Function Point method it accumulates points according to the complexity of system;the number of db tables,dialog boxes, print forms and interfaces. Acquired points could be converted into cost.

3.2 COCOMO Method

COCOMO Method[6] estimate the duration (Person-Months) in system development projects. In the COCOMO method, volume of source code is estimated by Equation 3. Duration (Person-Months) could be calculated with dividing volume of source code by the number of staffs. COCOMO Method propose Equation 3. And it uses the parameters as follows: C_0,C_e, P_1, P_2, and P_3.

C_e: estimate duration (PM:Person-Months) for expectation
C_0: estimate volume of source code
P_1: parameter for estimated productivity
P_2: exponent parameter for software development
P_3: calibration parameter
A challenge with this method is the parameters used for the cost estimation method are empirically derived and contextually dependent on many different factors, such as team size, project complexity, cultural environment, and others.

$$C_e \ = (C_0 \times \ P_1)^{P_2} \times \ P_3 \tag{3}$$

3.3 Function Point Method

Function Point Method estimate duration (PM:Person-Months) as those of COCOMO methods [5][6]. In the Function point method it is necessary to count the number of internal and external files, tables and internal and external interfaces. Function Point Method propose equation 4.And it uses the parameters as follows: C_e,F_p, F_1, P_1, and P_2.

C_e: estimate duration (PM:Person-Months) for expectation
F_p: estimate function points
F_1: function points
P_1: parameter for estimated productivity

P2: calibration parameter

$$C_e \ = (F_p \times \ P_1), \ \ F_p = F_1 \times \ P_2 \tag{4}$$

3.4 Other Methods to Estimate

Additionally,there is another cost estimate method as Experience method.In the experience method, total cost is estimated based on previous experiences. In these methods, there are often gaps between system developers cost estimates and customer expectations. This results from differences in how developers and customers group costs. Estimating correctly is important to finish building system on schedule and under budget. Thus, translating requirements into factors to estimate is essential. Certain requirements are needed to estimate properly. But there is no cost estimate method taking certainty of requirements into account.This paper considers taking certainty of requirements into account to estimate in order to gain customer agreement.

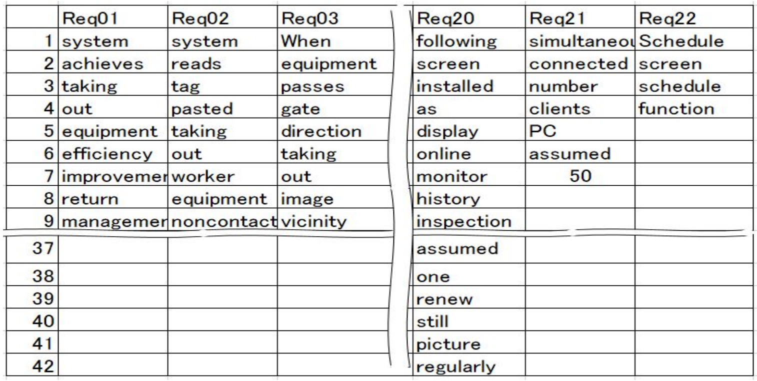
4 Requirement Analysis using Linguistic analysis

Taking correct requirements is essential to estimate properly.And over cost or schedule delay is caused by missing evaluation of requirements. Otherwise over cost or schedule delay is caused by many remediation of requirements. Remediation for essential requirement and uncertain requirement has large risk.thus It is vital to distinguish the requirement which gives large impacts to specification or budget of project.This research propose the method to distinguish influential requirements that has large risks.This paper shows a method to distinguish an influential requirements with with linguistic analysis and cost share rate. Cost share rate is defined as the percentage of total cost assigned to each requirement. This paper analyzes the requirements that were requested in past small system development projects by linguistic analysis. This small project is building a knowledge collecting system. In this project requirements were revised four times. Thus this paper analyzes requirement version one and version four. This paper predict risks from the result from analyzing requirement version one. Risk is considered as cost in risk management. This results;acquired cost is compared with the actual cost at the completion of project. If risk for requirement could be predicted properly, it would contribute project management. In this research,overlapping keywords are extracted form each requirements with linguistic analysis. Overlapping keywords are words that appear in one requirement and groups of keywords that appear in each category or phase in system development project. Categories or phases are Design, Development, Print, Test, Interface and Document. Overlapping keywords indicates relationships between one requirement and each other. The number of relationships that each requirement has with other requirements indicate essentiality and importance. Steps of linguistic analysis are as follows:
(1) Extract keywords from each requirement in version four. (see Figure 2).

(2) Count overlapping key words from extracted keywords, also count overlapping key words from extracted keywords in version four(see Figure 3).

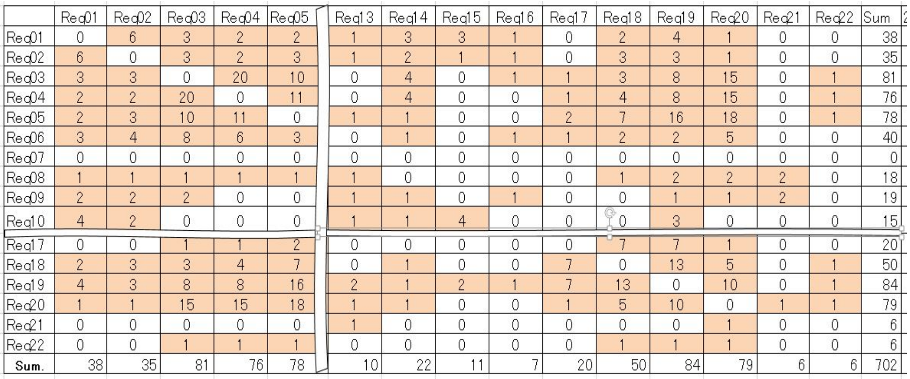
(3)Distinguish essential requirements by counting the number of over lapping keywords and subjective cost share rate from three engineers, and measure the distance from most essential requirement to each requirement in version four (see Figure 4).

(4)Distinguish essentiality of each requirements from these results, and distinguish categories that each requirements belong in version one.



	Req01	Req02	Req03	Req20	Req21	Req22
1	system	system	When	following	simultaneou	Schedule
2	achieves	reads	equipment	screen	connected	screen
3	taking	tag	passes	installed	number	schedule
4	out	pasted	gate	as	clients	function
5	equipment	taking	direction	display	PC	
6	efficiency	out	taking	online	assumed	
7	improvemer	worker	out	monitor	50	
8	return	equipment	image	history		
9	managemer	noncontact	vicinity	inspection		
37				assumed		
38				one		
39				renew		
40				still		
41				picture		
42				regularly		

Figure 2: Extract keywords from requirement version four



	Req01	Req02	Req03	Req04	Req05	Req13	Req14	Req15	Req16	Req17	Req18	Req19	Req20	Req21	Req22	Sum
Req01	0	6	3	2	2	1	3	3	1	0	2	4	1	0	0	38
Req02	6	0	3	2	3	1	2	1	1	0	3	3	1	0	0	35
Req03	3	3	0	20	10	0	4	0	1	1	3	8	15	0	1	81
Req04	2	2	20	0	11	0	4	0	0	1	4	8	15	0	1	76
Req05	2	3	10	11	0	1	1	0	0	2	7	16	18	0	1	78
Req06	3	4	8	6	3	0	1	0	1	1	2	2	5	0	0	40
Req07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Req08	1	1	1	1	1	1	0	0	0	0	1	2	2	2	0	18
Req09	2	2	2	0	0	1	1	0	1	0	0	1	1	2	0	19
Req10	4	2	0	0	0	1	1	4	0	0	0	3	0	0	0	15
Req17	0	0	1	1	2	0	0	0	0	0	7	7	1	0	0	20
Req18	2	3	3	4	7	0	1	0	0	7	0	13	5	0	1	50
Req19	4	3	8	8	16	2	1	2	1	7	13	0	10	0	1	84
Req20	1	1	15	15	18	1	1	0	0	1	5	10	0	1	1	79
Req21	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	6
Req22	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	6
Sum.	38	35	81	76	78	10	22	11	7	20	50	84	79	6	6	702

Figure 3: Count over lapping keywords from requirement version four

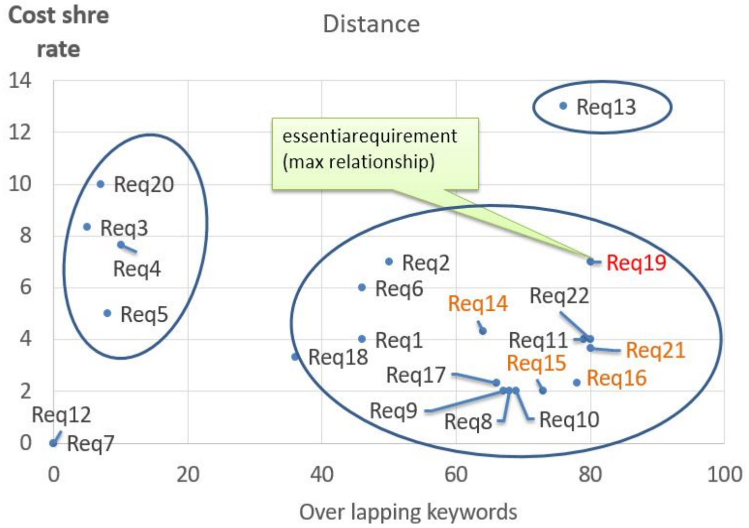


Figure 4: Specify important requirement from requirement version four

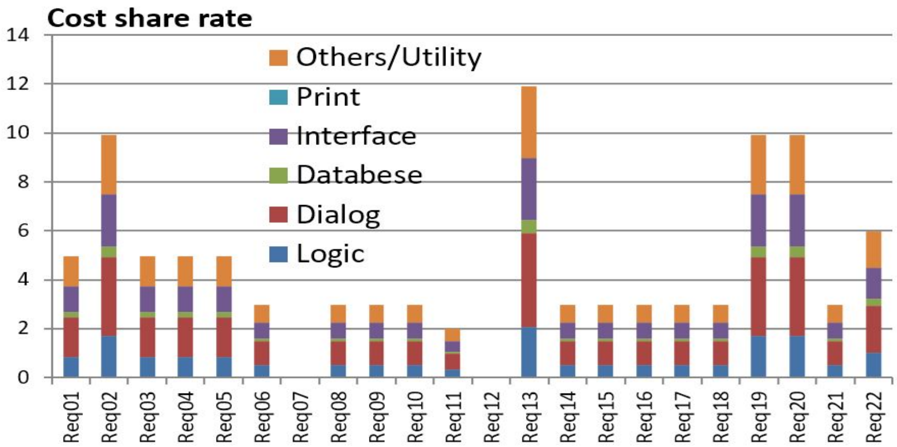


Figure 5: Cost share rate of requirement version four

5 Prediction of Cost Share Rate

Usually costs for system development projects are estimated by grouping costs with the number of dialog boxes, interfaces or print forms . Alternatively, costs may be estimated by associating cost to logic design, development, test, adjustment and documents; but, not according to the requirements .System developers estimate according to the complexity of projects, but customers expect the cost according to the number of requirements. Thus, customers could not understand the estimates provided by system developers. This paper shows a method to calculate cost share rate for each requirement in order to evaluate requirements accurately with mutual understanding of the developer and customers. Also cost share rate indicates importance of each requirement.Cost share rate is defined as the percentage of total cost assigned to each requirement. Figure 5 sows cost share rate of version our of past project. Although in this research cost rate(%) for each requirements are subjective figure from the two staffs that worked on this system development project. And cost share rates are gained by multiply cost rate by cost under the estimate. This cost rate and estimated cost are the value that according to the category.

6 Probability Prediction of schedule delay

This paper aim to predict risk of system development project based on requirements analysis. This paper consider one of risk of system development project is schedule delay. Thus this research suppose probability of schedule delay follows β distribution(see Equation 5),Sample data in Table 1 are surveyed in past system development project; Rdf system for tool tracking in machine factory. Then parameters (see Table 2) are gained by curve fitting sample data into β distribution. In this analyzing process x is probability parameter that indicates start day for each task, and y is ratio for schedule delay against actual days. Table **??** shows that survey/preparation,design and programming process have risk of schedule delay.Although test and writing document process have no risk of schedule delay.Figure 6 shows β distribution curve in this case from parameters(see Table 2).

$$f(x) = c \times x^{\alpha\ -1}(1-x)^{\beta\ -1} \tag{5}$$

7 Risk Prediction for business system development project

Figure 6 shows that probability of schedule delay is 0.225. Thus this research assign probability of risk;0.225 to essential requirements that distinguished by requirements analysis.Actually Ta-

Table 1: Sample data from past project

Item	Survey/	Design			Programing			Test		
(days)	Prepare	Dia.	Fun	DB	Dia.	Fun	DB	Test	Doc.	Sum
Schejule	11	29	27	27	9	58	39	3	13	216
Actual	35	34	36	27	9	55	62	2	7	267
Delay	24	5	9	0	0	-3	23	-1	-6	51
Prog	0	0.52	0.6	0.59	0.7	0.74	0.59	0.95	0.93	

Prog.:Progress Rate Fun:function Doc.:Document Dia.:Dialog

Table 2: Acquire Parameter for β distribution

C	α	β
0.839835	1.020625	3.047617

ble 3 shows that requirement3 and requirement4 are about design and development, requirement6 is about development.These requirements are essential, and assigned probability of risk as 0.225. Requirement1 is about Main Design,and assigned probability of risk as 0.1,because in past three project contingency is set as10% (see Table 4). Contingency is set for refinement.Refinment contains deleting,adding or refactoring.Table 4 shows occupying cost rate for each work process from surveying of past system development project; medical record system,knowledge management system.Thus risk is calculated probability of risk × cost(see equation 2),cost is analyzed by cost share rate. In estimated cost of requirement version one estimated total cost is 109(see Table 3).And at completion actual total cost is 109.1(see Table 3). At the result these results are equal.

8 Bayesian estimation

At the start of project costs are estimated. Also accuracy of estimation is needed. However, usually estimate isn't accurate . There are reasons why there are subjective factors, and the accuracy of the requirements is low in the beginning of project. Thus, this paper propose the method to estimate of the project properly using Bayesian estimation(expression 6) and cost share rate proposed in before section. The proportion of breakdown of cot in the past actual projects is shown(see Table 4). This table shows there are over cost as 10% in almost project. Table 5 shows actual cost and budget of small system development project. From this table the proportion of actual and budget is 100/104=0.96. Also the rate of over cost is 0.1 from past project(table 4). Thus these figure 0.96 and 0.1 are inputted into Bayesian estimation(expression 7)[6]. So the result of factor P(Est — Obud)=0.72 is gained. It shows accuracy of estimation of staffs. Also this figure is reasonable from experience in past projects. Also this results shows potential of gaining accuracy of estimation from staffs with Bayesian estimation.

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{6}$$

$$\begin{aligned} &P(Obud|Est) = \\ &\frac{P(Est|Obud)P(Obud)}{P(Obud)P(Est|Obud) + P(Obud)P(Est|Obud)} \\ &Obud : Proceedon\ budget\ Actually \\ &Est : Estimation \end{aligned} \tag{7}$$

Table 3: Estimated budget from requirement version one with risk analysis

Requirement Version one	Attribute	Cost Share Rate Ver one(%)	Schdule delay Probability	Conditional Probability	Monitary Risk (Contengency)	Estimated Cost Ver one
Requirement1	M	22.5	0	0.1	2.25	24.75
Requirement2		25	0	0	0	25
Requirement3	D/De	5	0.225	0	1.125	6.125
Requirement4	D/De	7.5	0.225	0	1.6875	9.1875
Requirement5		22.5	0	0	0	22.5
Requirement6	De	17.5	0.225	0	3.9375	21.4375
Toatal		100			9	109

M:Main, D:design, De:Development

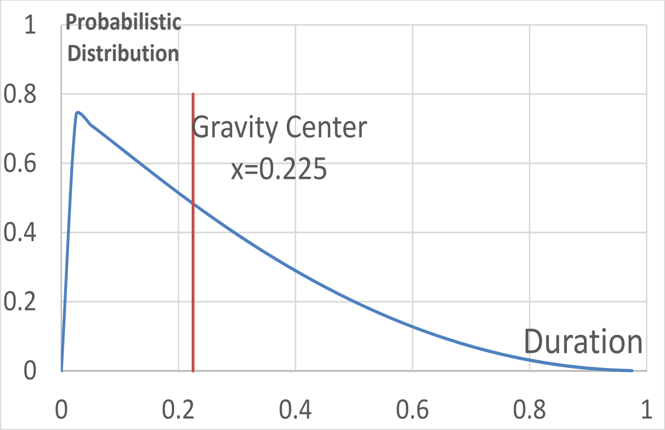


Figure 6: Beta distribution from past project

9 Conclusion

Meeting budget, finishing on schedule, and maintaining high quality are all important in project management. If cost and duration would be gained accurately, meeting budget and finishing on schedule would be achieved and managing project well. Estimate miss by developer causes finally over cost and schedule delay. Subjective factor and inaccuracy in requirements are reasons why estimate miss is caused. Generally estimate is measured by amount of source code of system or complexity of system. Also it is not estimated based on essentiality or risk of requirements. Misunderstanding of requirements and subjective factors cause misses in estimate or schedule.Then, this paper proposes cost share rate to measure essentiality of requirements in order to estimate accurately. Cost share rate is defined as the percentage of total cost assigned to each requirement Usually on risk management risks are evaluated according to staffs' experience. Also risks are not evaluated based requirement analysis. Evaluating risk properly is needed in order to manage project well. Thus this paper show better results by using cost share rate from requirement analysis. Additionally, this method helps prioritize requirements and narrow down specifications of the project. Prioritizing requirements and narrowing down specifications accurately help ensure it meets budget and duration targets. Additionally, this paper shows possibility to obtaining

Table 4: Data from past project

Project	Design	Development	Testing	Document	Refinement	Total
P1	3	4	1	1	1	10
P2	1	6	1	1	1	10
P3	1	6	1	1	1	10
Average	1.7	5.3	1	1	1	10

Table 5: Actual cost and budget of small system development project

Rwq	Contents	First	Final
Req1	main	10	6
Req2	rfid tag	13	10
Req3	gate	5	7
Req4	gate	5	7
Req5	process	5	7
Req6	exception	5	7
Req7	deleted		
Req8	limit	2	1
Req9	limit	2	1
Req10	tag type	2	2
Req11	exception	2	2
Req12	deleted		
Req13	dialog	13	14
Req14	process	5	8
Req15	tag type	2	2
Req16	output	2	2
Req17	authority	2	2
Req18	process	5	5
Req19	maintenance	5	5
Req20	process	5	5
Req21	limit	2	2
Req22	schedue	8	9
	Total	100	104

probability for risks accurately with cost share rate and Bayesian estimation. However, this result was obtained by small case. Thus further research and study is needed to refine and improve this method to obtain cost share rate and risk more accurately.

10 Discussion

Customers and developers estimate costs differently, resulting in differing expectations for project cost. Because in business system development projects there are many ways to implement requirements, there are large variability in translating user's requirement into system specification. It differs greatly according to staff's skill. Additionally subjective factors is another reasons why estimate does not meet final cost. Thus proper estimate by requirement analysis is needed in order to finish business system development project fine. Also correct probability for risk is needed to build proper estimate. Requirements have invisible risks. There are plus risk and minus risk. Minus risk gets prospect of profit worse, but plus risk gets prospect of profit well. But plus risk is not visible, plus risk is only in mind of staffs individually.This is one reason why estimated cost:109 match total cost:109(see Table 3). This paper shows potential to estimate accurately with Bayesian estimation in project managements. Additionally this paper shows potential to predict risk a accurately using conditional probability and Bayesian estimation [7][8][9].

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 17K00354.
The author would like to thank INTERLINK Inc. and TECHNO SOLUTION Inc. for data and advices.

References

[1] Shinji Mochida,James Righter, Joshua D. Summers: *A Study of Project Cost Management Based on Requirements Analysis, International Journal of BMFSA Vol. 21, No. 1* ,jp, 2016. 21-27

[2] Parviz F. Rad, Kou Itou, japanese translation: *PROJECT ESTIMATING AND COST MANAGEMENT* ,Seisansei Syutupan,jp, 2004

[3] Shinji Mochida: *A Study on Method of Measuring Performance for Project Management, 20th ISPE International Conference on Concurrent Engineering* ,IOS Press Ebooks,US, 2013. 264-273

[4] Beshoy Morkos,James Mathieson,Joshua D. Summers: *Comparative analysis of requirements change pre-diction models: manual, linguistic, and neural net-work, Research in Engineering Design Vol. 25* ,Springer,US, 2014.139-156

[5] Project Management Institute: *A Guide to the Project Management Body of Knowledge Fourth Ed. (JAPANESE TRANSLATION*,Project Management Institute,US, 2008

[6] Kazuya Fujita: *That is Bayes*,OHMSHA,jp, 2015

[7] Christian Robert : *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation* ,Springer Texts in Statistics,US, 2007

[8] Paul G Hoel,Akira Asai,Masayasu Murakami japanese translation : *ELEMENTARY STATISTICS* ,Springer Texts in Statistics,jp, 2011

[9] David Blackwell,Kazuyoshi Mikuni japanese translation : *BASIC STATISTICS* ,Kogakusha,jp, 1972

Analysis of activity data from fitness trackers with a focus on an active life of older people

Jitka KUHNOVA[1] and Richard CIMLER[1]

[1] *University of Hradec Kralove, Faculty of Science*
Rokitanskeho 62, Hradec Kralove
Czech Republic
jitka.kuhnova.2@uhk.cz

1 Background

Population is getting older and an active life-style in an advanced age is good not only for themselves but also for the society. Active elderly people will be in a better health condition, that means, they can enjoy more from a longer life. Moreover, healthier older population reduce costs on health care in general.

Modern technology can help people to enhanced their habits for workouts by monitoring their activities, by suggesting to be more active or to compare their progress with other people. There are many different type of sport wearables, each with its own level of measuring and storing the data.

Different studies deals with a problem of active life of elder people. These studies gathers and evaluates data about daily routines of elder people. We have created a system which might help in such studies by automatically gathering data from different sources and by sending predefined questionnaires. These questionnaires can help researchers to get better and accurate information from members of a study. Results are automatically processed into the reports for better understanding of the data. Researchers can be also notified about problems during study such as low response rate or lack of data from fitness trackers or other data sources.

1.1 Health react

System called Health React consists of a Server application and an Application for smart phones. Main purpose of the server application is to gather data from different sources, evaluate them and send questionnaires to users based on created rules. Primary, the system is designed to gather data from fitness trackers from various vendors, but there might be different data sources as well.

However, the system is not limited to monitor activities from fitness trackers. It can be connected to different systems providing data and reacts based on predefined conditions (e.g. if persons stays in the bed each day for unusually long time for a few days in a row – system can send a questionnaire about health state or mood).

Gathered data are automatically processed and reports are sent to the users, researchers or doctors. There might be different layouts and information in the reports. Users get an overview about goals they achieve, researchers might be interested in comparisons of different users or groups, for doctors, information about unusual behavior or change in persons habits might be important.

Visualization of the data helps to better understand what are behind them. Graphs are automatically created from the gathered data and inserted into the reports.

Fitness trackers provide various data such as data about sleep, its stages and quality, number of steps per given time frame, heart rate. Other indicators might be computed based on the gathered data such as activity classification depending on heart rate and steps.

During the presentation, the system will be introduced with a deeper insight into the reports created during a pilot study.

Acknowledgement
The authors would like to thank Faculty of Science, University of Hradec Kralove.

Unobtrusive Nocturnal Estimation Method for Total Scratching Time with Piezoelectric Ceramics

Keita NISHIO[1] and Takashi KABURAGI[1] and Satoshi KUMAGAI[1] and Toshiyuki MATSUMOTO[1] and Yosuke KURIHARA[1]

[1] *Aoyama Gakuin University*
5 Chome-10-1, Fuchinobe, Chuo-ku, Sagamihara-city, Kanagawa
Japan
nishio.engineering.agu@gmail.com

1 Introduction

Itching is caused by many kinds of skin diseases and it makes a patient to scratch the skin. Scratching relieves the itching temporarily, however, it deteriorates the skin condition. In general, the scratching motion is monitored because the time length of scratching is related to the severity of the skin diseases accompanied by itching. A diagnosis based on monitoring scratching while a patient is awake is not so accurate because the patient can deliberately control scratching motion during a waking day. On the other side, a patient scratches the skin without restraint while the patient is asleep, so monitoring nocturnal scratching achieves more effective diagnosis. The information on scratching time is converted into an index known as TST%, which is the ratio of the total scratching time (TST) to the total measurement time during sleep. Measurement of the nocturnal scratching time is currently conducted using an infrared camera in hospital [1], [2], though this measurement method can violate the privacy of the patient. Thereby, some privacy-conscious methods have been recently proposed. As one of high-accuracy monitoring methods, a wrist-worn accelerometer is proposed [3], [4]. Although the method doesn't invade the privacy of the patient, the sensing device is obtrusive. In this paper, we propose an unobtrusive scratching estimation method with sensing devices based on piezoelectric ceramics. In addition, this paper presents the signal processing for estimating TST from output signals from piezoelectric ceramics.

2 Proposed Sensing Device and Signal Processing Flow for Estimation of TST

We devise a sensing device that measures the vibration due to scratching motion on a bed. The device is consisted of a piezoelectric ceramic, an upper metal plate, a lower metal plate and a cylinder-shaped metal spacer. A cylinder-shaped spacer is sandwiched between the upper metal plate and the lower metal plate. A piezoelectric ceramic is attached on the back side of the upper metal plate. The sensing device is placed under each leg of the head side of a bed frame. Piezoelectric ceramics bend beneath by the weight of the bed. Under this equipment, we can obtain the vibration due to scratching motion of a patient lying on the bed as electric signal from the piezoelectric ceramic. Let the output analog signals from the sensing device place under the right and left leg be $I_r(t)$ and $I_l(t)$ respectively. $I_r(t)$ and $I_l(t)$ are imported into a computer through A/D converter as digital signals $I_r(k)$ and $I_l(k)$. Moreover, let the max amplitude of $I_r(k)$ and $I_l(k)$ be max_r and max_l. From $I_r(k)$, $I_l(k)$, max_r and max_l, $I(k)$ is calculated as shown in equation (1).

$$I(k) = I_r(k) + \frac{max_r}{max_l} I_l(k) \tag{1}$$

$I(k)$ is analyzed by continuous wavelet transform with the mother wavelet of Morlet. $W(f,k)$ is the wavelet transform of $I(k)$ and it represents the intensity of signal component of frequency f at discrete time k. $A(f,k)$ has an absolute value of each element of $W(f,k)$. Then, $A(f,k)$ is factorized into two matrices $V(f,m)$ and $H(m,k)$ by nonnegative matrix factorization with order m. $H(1,k)$, the first row vector of $H(m,k)$, is used as feature vector $F(k)$. The optimal threshold τ for estimation of scratching is determined utilizing ROC curve. The intersection point of ROC curve and the line, which connects the two points p_1 and p_2, is calculated. At the point p_1, true positive rate is 1 and false positive rate is 0. At the point p_2, true positive rate is 0 and false positive rate is 1. The threshold value corresponding to the intersection point is used as the optimal threshold τ. If $F(k)$ is greater than τ, the discrete time k is estimated as scratching time. Otherwise, if $F(k)$ is less than or equal to τ, the discrete time k is estimated as not scratching time.

3 Result

In order to validate the proposed method, we implemented the validation experiment. The sensing devices are placed under the legs on head side of ordinary bed frame. Output signals are imported through A/D converter (NR-2000, KEYENCE CORPORATION) in which sampling interval is 10ms and the measurement time is 30s. We obtained 14 data from 4 subjects. After obtaining informed consent from each subject, the experiment is implemented. At the start of experiment, the subject lies on the bed with his arms and legs straight. During the measurement, the subject is required to scratch the right cheek at an arbitrary timing. When the subject starts and finishes scratching, the subject touches a piezoelectric ceramic placed near the other hand and it is used as reference data. Then, the subject puts the arm to the original position. Order m for nonnegative matrix factorization is set to 2. Using the estimation result, we count the true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In order to evaluate the method, *sensitivity*, *specificity*, positive predictive value (PPV), negative predictive value (NPV), and *F_value* are calculated from TP, TN, FP, and FN as shown in equations (2)-(6).

$$sensitivity = \frac{TP}{TP+FN} \tag{2}$$

$$specificity = \frac{TN}{TN+FP} \tag{3}$$

$$PPV = \frac{TP}{TP+FP} \tag{4}$$

$$NPV = \frac{TN}{TN+FN} \tag{5}$$

$$F\_value = \frac{2TP}{2TP+FP+FN} \tag{6}$$

As the result, we obtained 0.97 of sensitivity, 0.97 of specificity, 0.93 of PPV, 0.98 of NPV, and 0.95 of F-value, respectively.

4 Conclusion

Monitoring nocturnal scratching is effective for diagnosing the severity of skin diseases. However, the TST calculation method using an infrared camera has a problem of privacy violation. As one of high-accuracy monitoring methods, wrist-worn accelerometers is proposed although the sensing device is obtrusive. In this paper, we proposed the unobtrusive scratching monitoring method using piezoelectric ceramics and achieved 0.97 of *sensitivity*, 0.97 of *specificity*, 0.93 of PPV, 0.98 of NPV, and 0.95 of F-value, respectively. We succeeded in estimating scratching motion in unobtrusive and privacy-conscious manner.

References

[1] T. Ebata, H. Aizawa, and R. Kamide. An Infrared Video Camera System to Observe Nocturnal Scratching in Atopic Dermatitis Patients. *Brit. J. Dermatol.*, vol:23, pp. 155, Sep. 1996.

[2] H. Izumi, T. Ebata, Y. Sato, H. Aizawa, R. Kamide and M. Niimura. A Simplified Method for the Measurement of Nocturnal Scratching with an Infrared Video Camera. *Skin Res.*, vol:39, issue:6, pp. 560-563, 1997.

[3] A. Moreau, P. Anderer, M. Ross, A. Cerny, T. Almazan and B. Peterson. Detection of Nocturnal Scratching Movements in Patients with Atopic Dermatitis Using Accelerometers and Recurrent Neural Networks. *IEEE Journal of Biomedical and Health Informatics*, vol:22, issue:4, July 2018.

[4] J. Petersen, D. Austin, R. Sack, and T.L. Hayes. Actigraphy-Based Scratch Detection Using Logistic Regression. *IEEE Journal of Biomedical and Health Informatics.*, vol:17, issue:2, pp. 277-283, March 2013.

Prediction of a spread of Alzheimers disease using computer simulations

Richard CIMLER[1] and Jitka KUHNOVA[1] and Ondrej DOLEZAL[2]

[1] *University of Hradec Kralove, Faculty of Science*
Rokitanskeho 62, Hradec Kralove
Czech Republic
richard.cimler@uhk.cz

[2] *University of Hradec Kralove, Faculty of Informatics and Management*
Rokitanskeho 62, Hradec Kralove
Czech Republic

Abstract

Aim of this paper is to present a computer simulation for a prediction of a number of patients with Alzheimer's disease (AD) in Japan. Predictions are based on the population tables and characteristics of the disease prevalence rate in Japan. As a methodology, a numeric model for the prediction and a statistic evaluation of given results has been used. Prevalence of the disease has been gained from results of review papers focused on dementia and AD in Japan. Results show that prevalence rate differs significantly in various papers based on the year of the study. Simulation compares different scenarios based on the selected prevalence rate. There is a huge difference in the results according to chosen prevalence rate. The maximum of predicted patients with AD vary from 4 to 8 millions of people depending on the chosen prevalence.

Keywords: Prediction, Numeric model, Alzheimer's disease, Japan, Population

1 Introduction

Dementia has become a serious medical, social and economic problem, especially in developed countries where a population is getting older. Alzheimer's disease is the most common type of dementia with 60 %–80 % cases of Alzheimer's disease.[1] Prevalence rate of the Alzheimer's disease is increasing with an age of persons. Japan is the worlds fastest aging society, according to Prince [2] there is 3.2 % of older people, therefore it is crucial to focus on a prediction of people with this disease. However, there is a lack of studies with detailed data about number of people with AD, prevalence or incidence rate from recent years or estimations to the future.

Sekita [3] shows, that the prevalence rate of Japanese population (75 and older) with AD have indeed risen significantly since 1985.

Montgomery [4] discusses economical and social burden that rising AD population puts on families and whole Japanese medical system. According to their estimation, expenditures related to dementia have more than doubled from 2002. Ohara [5] just confirm very much the same trend for more recent past.

Life expectancy rise will most likely significantly impact the structure of Alzheimer's Disease patients in the future. In case that we could accurately anticipate population development and specifically development of AD trough population, we would have a helpful instrument for different fields such as monetary burden of treatment of AD patients, mortality expectations, research goals and more [6, 7].

Population simulations are useful apparatus to predict oncoming changes and to show the drift in a society. They can help to prepare for the future increased costs associated with the rise of the population with Alzheimer's disease. In this paper, a computer simulation for a prediction of a number of Alzheimer's disease patients in Japan in next decades is introduced.

Article has structure as follows: Our previous work is in the following section 2. Methodology and model background is described in the section 3. Results of the simulations can be found at 3.1 and are discussed in the section 4.

2 Modeling methods

In our previous work, we focused on the selection of a correct methods for creating of simulations. Three different approaches has been compared in [8]. Numeric models, system dynamic and agent-based models. Result shows different pros and cons for each method. Numerical models are fast and enables to create models which are not very complex however it is time consuming to make changes in models or to add new parameters. For more complex models, where different parameters are observed and which consist of more complex rules, it is more suitable to use one of the following approaches.

Advantage of System dynamic models is its interactivity and possibility to create models by drag and drop, however major changes or adding new parameters is still difficult and time consuming. Result of a simulation created by this approaches can be found in [9]. Population of European Union has been simulated until the year 2070 and number of people with AD has been observed.

Agent based modeling enables to change simulation or add new parameters very easy however this approach is very demanding on hardware resources if the vast population is simulated. Different modeling languages were compared during a creation of an agent based population in [10]. Results show that using the RUST programming language can save significant amount of computational resources and shorten run time of the simulation.

3 Methodology

For this research, a numeric model is created. Population data and disease characteristics are needed for the initial setup of the simulation. Population data has been gained from [11]. Three different scenarios of population prediction for Japan can be found in the materials – Low, Moderate and High. Population is divided into five-years age cohorts and the prediction time starts in 2015 and end in 2100 after 18 steps (each in five year from the previous).

Disease characteristics has been gained from different reviews of literature focused on Dementia and AD in Japan. According to [4], in a Hisayama study [12], AD prevalence increased from 1.4 % in 1985 to 12.3 % in 2012. This indicates, that older data about prevalence can be insufficient. It was also shown that relative increase in the prevalence of dementia was insignificant.

Prevalence rate for Japan population has been calculated based on the newest results of a review paper [13], see Table 1.

Table 1: Prevalence rate by [13]

Study	Age 70-79	Age 80-89	Age 90+	Number of subjects
Hisayama (1985)	3.41	19.50	57.14	440
Hisayama (1992)	3.35	13.39	46.43	537
Okinawa (1991–1992)	3.67	14.93	36.83	1481
Hiroshima (1992-1996)	5.67	19.04	41.67	741
Hisayama (1998)	3.99	16.84	35.59	676
Tajiri (1998)	5.18	22.33	51.52	908
Hisayama (2005)	6.86	27.74	53.49	729
Ama-cho study (2008)	6.7	17.8	41.3	432

In the paper, authors summarized epidemiological studies of dementia in Japan. Review is gaining information from eight large population studies of dementia in Japan. However, missing prevalence rate for people younger than 65 years old, were obtained from older study Prince (2008) [14] (see Table 2).

Table 2: Prevalence rate by [14]

60–64	65–69	70–74	75–79	80–84	85+
0.6	1.4	2.6	4.7	10.4	22.1

We used four different prevalence rates to compare predictions. The first one is the one described above. We will refer to the result of the simulation with this prediction as "Basic" model. The second one is using only an older prevalence according to [14], it would be referred as "Prince (2008)". Moreover, we would like to compare the prediction with the prevalence based on the incidence from our previous research ([8]). This prevalence seems to be higher than both previously mentioned, although it is more convenient, both previous prevalence rates are based on researches not younger than ten years and according to [4], it seems reasonable that prevalence would increase from that time. This prevalence was not constant for all the prediction time, therefore we chose two alternatives – mean of the prevalence for each group of age cohorts, and maximal value.

3.1 Results

Models have been run and results can be seen in the following Tables and Figures. Firstly, the prediction of population has been done based on the data from [11] for all three population scenarios. Figure 1 shows there is no difference in a prediction of population with AD for the first 65 years of the simulation.

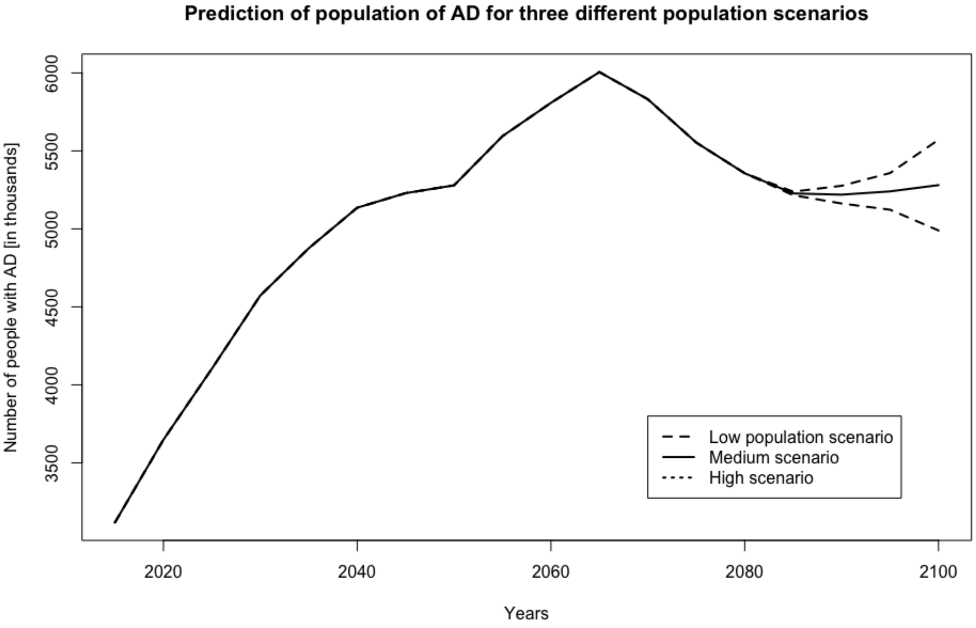


Figure 1: Prediction of patients with AD based on three population prediction scenarios.

This is because the three scenarios diverge only for younger population cohorts. This does not affect older age cohorts hence people with AD. As the simulations go on, young cohorts are getting older and after 65 years it finally change results in the three scenarios. Example of the population distribution with the union of the three scenarios in the year 2065 can be seen at the Figure 2.

Depending on this finding, we will further use only the medium prediction scenario. Thereafter, we can simulate four different prevalence rates as described in the section 3. Results of our simulations can be seen in the Table 3 in a comparison with the results of a study [15]. Sekine in [15] studied the prevalence of dementia in Toyama prefecture. The prevalence has increased from 4.7 % in 1985 to 15.7 % in 2014 and based on these values, they made a prediction of a population with dementia. To be able to compare our result with theirs, we had to multiplied number of people with dementia by 0.7 to obtained adequate number of people with AD. Sekine [15] made two predictions – the first one based on estimates of prevalence of dementia in 2014, the second one as an estimation of a linear regression model (LRM).

It can be seen in the Table 3 and Figure 3 that simulation using prevalence from [13] gives similar results as [15] first model. Models using EU prevalence (either mean or maximal) gives in

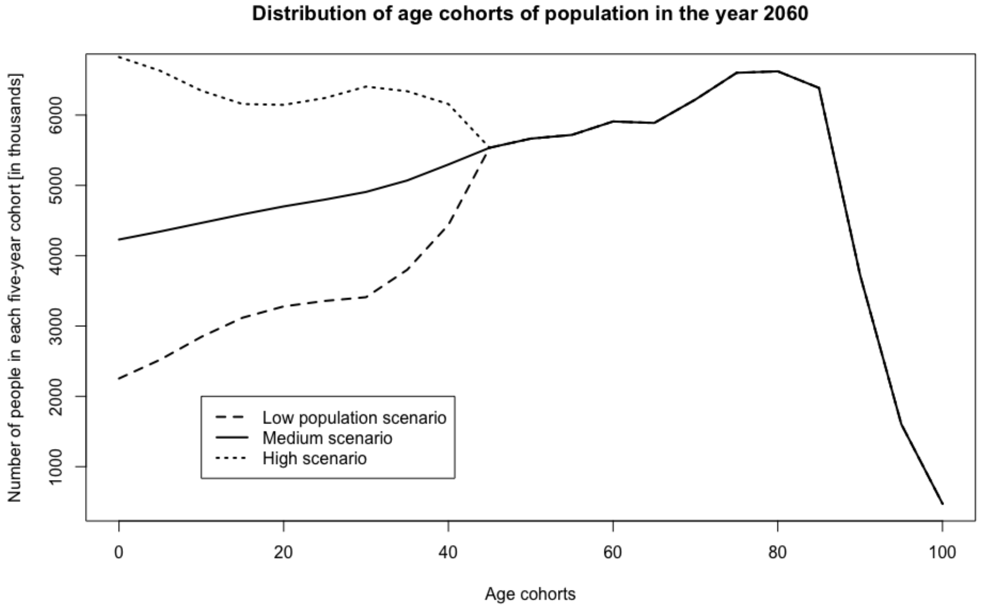


Figure 2: Comparison of distribution of age cohorts in all three considered scenarios.

Table 3: Comparison of results – number of millions of people with AD

Model	2015	2025	2035
Sekine prediction 1 (2014)	3.4	4.3	5.0
Sekine prediction 2 (LRM)	3.3	5.0	6.8
Our results with Basic prevalence	3.1	4.1	4.9
Our results with Prince (2008) prevalence	2.2	2.8	3.4
Our results with mean EU prevalence	4.2	5.3	6.1
Our results with maximal EU prevalence	4.8	6.0	6.9

the most of the cases higher results.

We can also compare the prevalence rate for the whole population of AD patients, means the prevalence of AD within people older than 65 years. According to [4], the prevalence rate of dementia due to AD was 12.3 % in the Hisayama in the year 2012. This corresponds with our prediction with mean EU prevalence (see Table 4).

Table 4: Comparison of results – prevalence rate for 65+

Model	2015	2025	2035	2045	2055	2065	2075	2085	2095
Basic prevalence	9.4	11.3	13.0	13.1	14.4	16.7	16.8	16.7	17.3
Prince (2008) prevalence	6.6	7.8	9.0	8.6	9.6	10.8	10.8	10.7	11.0
Mean EU prevalence	12.6	14.5	16.2	16.3	17.8	19.9	20.7	20.7	21.4
Maximal EU prevalence	14.3	16.5	18.3	18.5	20.0	22.4	23.1	23.1	23.8

4 Discussion and Conclusion

Results of the simulations shows predictions of number of people with Alzheimer's disease in Japan until the year 2100. We made several prediction scenarios using different prevalence rates of

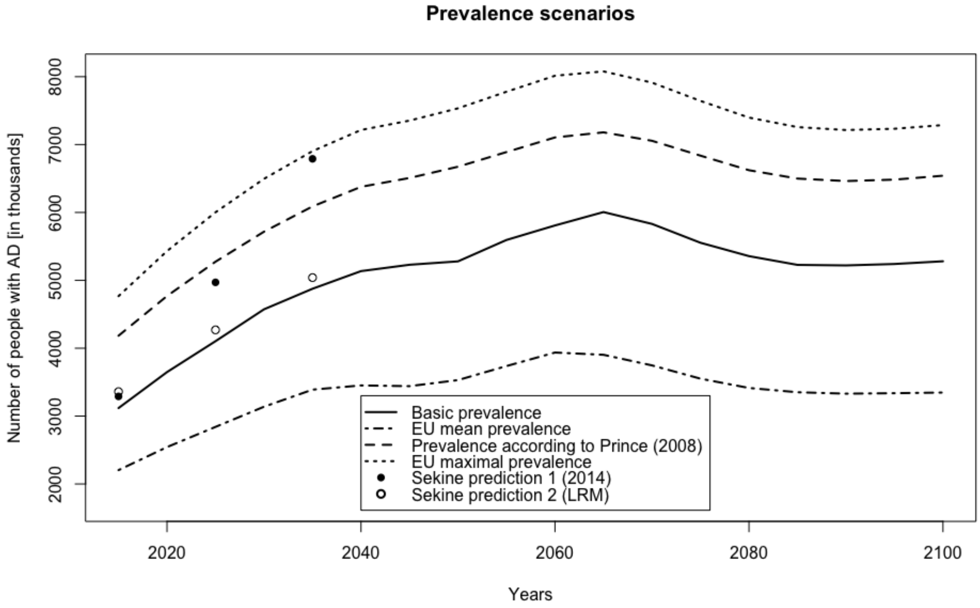


Figure 3: Comparison of prediction of patients with AD according to different prevalence, see Table 3

Alzheimer's disease. Basic model prediction corresponds with the first prediction from [15]. The population of patients with AD will increase and will reach the maximum slightly over 6 millions of ill people in 2065. In the worst case scenario, the maximum will be more than two millions higher with 8 millions of patients. On the other hand, in the scenario with lower prevalence rate, the maximum will be only four millions of people with AD.

Model is prepared for a future extension in which different affects of drugs or treatments will simulated and number of people with AD will be predicted,

Acknowledgement
Financial support of the project TACR ETA TL01000300 is gratefully acknowledged.

References

[1] Association Alzheimers. 2015 alzheimer's disease facts and figures. *Alzheimer's & dementia: the journal of the Alzheimer's Association*, 11(3):332, 2015.

[2] M Prince, A Wimo, M Guerchet, GC Ali, YT Wu, and M Prina. World alzheimer report 2015. the global impact of dementia. alzheimers disease international. *Alzheimers Disease International (ADI), London*, 2015.

[3] A Sekita, T Ninomiya, Y Tanizaki, Y Doi, J Hata, K Yonemoto, H Arima, K Sasaki, M Iida, T Iwaki, et al. Trends in prevalence of alzheimers disease and vascular dementia in a japanese community: the hisayama study. *Acta Psychiatrica Scandinavica*, 122(4):319–325, 2010.

[4] William Montgomery, Kaname Ueda, Margaret Jorgensen, Shari Stathis, Yuanyuan Cheng, and Tomomi Nakamura. Epidemiology, associated burden, and current clinical practice for the diagnosis and management of alzheimers disease in japan. *ClinicoEconomics and outcomes research: CEOR*, 10:13, 2018.

[5] Tomoyuki Ohara, Jun Hata, Daigo Yoshida, Naoko Mukai, Masaharu Nagata, Toru Iwaki, Takanari Kitazono, Shigenobu Kanba, Yutaka Kiyohara, and Toshiharu Ninomiya. Trends in dementia prevalence, incidence, and survival rate in a japanese community. *Neurology*, pages 10–1212, 2017.

[6] P Maresova and K Kuca. Missing uniform costs classification for alzheimer's disease treatment and care. *Current Alzheimer research*, 2018.

[7] Petra Marešová, Josef Dolejš, and Kamil Kuca. Call for a uniform strategy of collecting alzheimers disease costs: A review and meta-analysis. *Journal of Alzheimer's Disease*, (Preprint):1–12, 2018.

[8] Richard Cimler, Hana Tomaskova, Jitka Kuhnova, Ondrej Dolezal, Pavel Pscheidl, and Kamil Kuca. Numeric, agent-based or system dynamics model? which modeling approach is the best for vast population simulation? *Current Alzheimer Research*, 15(8):789–797, 2018.

[9] Hana Tomaskova, Jitka Kuhnova, Richard Cimler, Ondrej Dolezal, and Kamil Kuca. Prediction of population with alzheimers disease in the european union using a system dynamics model. *Neuropsychiatric disease and treatment*, 12:1589, 2016.

[10] Richard Cimler, Ondřej Doležal, and Pavel Pscheidl. Comparison of rust and c# as a tool for creation of a large agent-based simulation for population prediction of patients with alzheimers disease in eu. In *International Conference on Computational Collective Intelligence*, pages 252–261. Springer, 2016.

[11] United Nations. World population prospects. 2017.

[12] Hisayama study. http://www.neurodegenerationresearch.eu/cohort/hisayama-study/. 2018-09-30.

[13] Hiroko H Dodge, Teresa J Buracchio, Gwenith G Fisher, Yutaka Kiyohara, Kenichi Meguro, Yumihiro Tanizaki, and Jeffrey A Kaye. Trends in the prevalence of dementia in japan. *International Journal of Alzheimers Disease*, 2012, 2012.

[14] M Prince. The prevalence of dementia worldwide. *Alzheimer's Disease International Factsheet. London: Alzheimer's Disease International*, 2008.

[15] M Sekine, M Suzuki, H Kido, M Yamada, and T Tatsuse. The projected number and prevalence of dementia in japan: Results from the toyama dementia survey. *European Psychiatry*, 33:S206, 2016.

Efficient implementation of compositional models for data mining

Václav KRATOCHVÍL[1], Radim JIROUŠEK[1], and Tzong-Ru LEE[2]

[1] *Czech Academy of Sciences, Inst. Information Theory and Automation*
Pod Vodárenskou věží 4, CZ-182 08 Praha 8,
Czech Republic
velorex@utia.cas.cz, radim@utia.cas.cz

[2] *Department of Marketing, National Chung Hsing University*
145 Xingda Rd., South Dist., Taichung City 402,
Taiwan (R.O.C.)
trlee@dragon.nchu.edu.tw

Abstract

A compositional model encodes probabilistic relationships among variables of interest. In connection with various statistical techniques, it represents a practical tool for data modeling and data mining. Structure of the model represents (un)conditional independencies among all variables. Relationships of dependent variables are described by low-dimensional probability distributions. Having a compositional model, a data miner can easily apply an intervention on variables of interest, fix values of other variables (conditioning), or to narrow the context of a problem (marginalization). The model learning process can be controlled to avoid overfitting of data.

In this paper, we present a new semi-supervised web application that will enable researchers to design probabilistic (compositional) models (both causal and stochastic). Thanks to the web architecture of the system, the researchers will always have a possibility to influence the data-based model construction process from any place of the world. It is also expected that the application of this methodology to practical problems will open new problems that will be an inspiration for further theoretical research.
Keywords: Data mining, Mutual information, Compositional model, Conditional independence, Probability theory.

1 Introduction

This paper is a first introduction to a new computer system trying to implement the theory of compositional models for data-mining. We hope that it will attract researchers to apply the theory and encourage them in studying it because there are still blanks to be filled.

The basic idea of compositional models is very simple: it is beyond human capabilities to represent/express/understand global knowledge of an application area - one always has to work with pieces of local knowledge only. Such local knowledge can be, within probability theory, easily represented by a low-dimensional distribution. It should be stressed out that, based on the laws of mathematical statistics, it is evident that the dimensionality of the estimated distributions is strictly limited by the application of data-based models. Whatever size of data is at our disposal, we can hardly expect to obtain reliable estimates of probabilities of a 20-dimensional distribution (even for binary variables). Typically, one can assume that dimensionality of the considered distributions is between 2 and 8.

1.1 Compositional models

When pieces of local knowledge are represented by low-dimensional distributions, the global knowledge should be represented by a multidimensional probability distribution. The technique of compositional models describes directly how the multidimensional distribution is computed/composed from a system of low-dimensional distributions. Usually, one starts constructing such a model from a (usually great) number of low-dimensional distributions. Such a model resembles a jig-saw puzzle that has a large number of parts, each bearing a local piece of a picture. The goal is to figure out how to assemble them in a way that the global picture makes sense and reflects all of the individual small parts. The only difference is that we look for a linear ordering of distributions in our case.

1.2 Data mining

Generally, data mining is understood to be the process of discovering patterns in large data sets involving various methods from machine learning, statistics, and database systems. The goal of a data mining process is not a model itself but its interpretation in the form of a distilled knowledge.

One possible usage of compositional models in data mining stands in the process of model construction. For example, constructing compositional models from two data files collected in different cultural environments enables the user to compare the structures of the two models, revealing qualitative differences between the studied societies, and the comparison of the respective probability tables enables the researchers to describe the quantitative differences. The already mined data can serve also in the opposite direction. The supervised approach to model construction enables the researchers to influence the resulting models in the way that these models are easily comprehensible and interpretative. The user can have some knowledge about data, based on which the model is constructed.

1.3 Notation

In this paper, we consider only finite-valued variables, which are denoted by upper-case Latin characters. Groups of variables are denoted by bold-face characters: i.e., for example, $\mathbf{M} = \{X, Y, Z, W\}$. The set of values of variable X is denoted by $\mathbb{X}_X$. Similarly, we use $\mathbb{X}_\mathbf{M}$. Generally, we use the same notation as in the second paper by same authors in this proceedings [1].

Let us highlight that by a *state* of a group of variables we understand any combination of values of the respective variables.

Probability distributions are denoted by characters of Greek alphabet (κ, ν, π). To highlight that the given probability distribution is defined over variables a set of variables $\mathbf{K}$ we write $\kappa(\mathbf{K})$ A *marginal distribution* of $\kappa(\mathbf{K})$ defined for variables $\mathbf{L}$ is denoted as $\kappa^{\downarrow \mathbf{L}}$.

2 Compositional models

The key element of the theory of compositional models is the operator of composition. To be able to introduce these models, let us briefly recall its definition and a couple of its most important properties (to read more about basic properties of the operator of composition, we refer the reader to [2] and [3]).

For arbitrary two distributions $\kappa(\mathbf{K})$ and $\lambda(\mathbf{L})$, for which [1] $\kappa^{\downarrow \mathbf{K}\cap\mathbf{L}} \ll \lambda^{\downarrow \mathbf{K}\cap\mathbf{L}}$ is their composition given by the following formula[2]

$$(\kappa \triangleright \lambda)(x) = \frac{\kappa(x^{\downarrow \mathbf{K}})\lambda(x^{\downarrow \mathbf{L}})}{\lambda^{\downarrow \mathbf{K}\cap\mathbf{L}}(x^{\downarrow \mathbf{K}\cap\mathbf{L}})}. \tag{1}$$

Otherwise, the composition remains undefined.

The operator of composition is used to construct multidimensional compositional models. Composing two distributions, we can define a distribution of a higher dimensionality than any of the original ones.

By a compositional model of a multidimensional probability distribution we understand a sequence of low-dimensional distributions that assembled together using the operator of composition somehow models the original multidimensional distribution that would be difficult to handle otherwise.

Denoting the low-dimensional distributions $\kappa_1(\mathbf{K}_1)$, $\kappa_2(\mathbf{K}_2), \ldots, \kappa_n(\mathbf{K}_n)$, we get the compositional multidimensional model by the application of the operator of composition $\triangleright$ to this sequence

[1] $\kappa(\mathbf{M}) \ll \lambda(\mathbf{M})$ denoted that the distribution κ is absolutely continues with respect to distribution λ, which in our finite settings means that whenever κ is positive also $\lambda must be positive$.

[2] Define $\frac{0 \cdot 0}{0} = 0$.

from left to right: $\kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_n$. (Note that the operator is neither commutative nor associative.) This sequence of probabilistic distributions, if all compositions are defined, is called the *generating sequence* of the compositional model. The sequence of sets of variables the generating sequence is defined for – $\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_n$ – is called a compositional model *structure*. To visualize the structure we use a tool called a *persegram* - a table of markers where rows correspond to variables and columns to sets from the structure in the given ordering. A position in the table is marked if the variable is among variables of the respective variables sets. Markers for the first occurrence of each variable (i.e., the leftmost markers in rows) are box-markers, and for other occurrences there are bullets.

We say that a generating sequence is perfect if all elements are marginals of the resulting multidimensional distribution. It is worth noting that among all models, perfect models play an important role because they faithfully reflect the information contained in the individual distributions. This property is thus important from the point of view of potential applications: when the individual low-dimensional distributions κ_i represent pieces of local knowledge, then $\kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_n$ is a proper representative of global knowledge.

Note that compositional models can be also used for the representation of causal models [4]. In that case, interventions can be easily modeled by composing the model with a simple degenerated probability distribution.

3 Implementation

It is always beneficial for a theoretical work to experiment with a real problem. To evaluate various hypotheses and to support further theoretical research, it is necessary to have an experimental tool for calculations with compositional models. In this section, we would like to describe the basic ideas standing behind the implementations. As already written above, we have implemented a semi-supervised web application that enables a researcher to design the probabilistic compositional model (both causal and stochastic). We assume that the web architecture of the system will make it easily accessible and open for the wide range of audience.

The system is implemented in R environment [5] using Shiny web application framework [6] and data.table package [7].

3.1 Probability distribution

The key problem is the representation of a probability distribution in a computer memory. Because we restricted random variables to finitely valued discrete variables only, it seemed like a natural step to store probability distributions not as multidimensional arrays (hypercubes) where individual dimensions correspond to random variables, but as a listing of states in form of a table. Actually, we have been inspired by relational database theory [8]. We store a probability distribution as a database table - a *data.table* [7] in R.

Example 1 *Having a probability distribution over e.g. three random variables Y, Z, W - the corresponding table has four columns. First three columns correspond to random variables, the last column corresponds to an amount of probability dedicated to each row. Every row represents a unique state - a combination of values of the respective random variables. In case of a zero probability state, it does not have to be listed in the table.*

Y	Z	W	probability
0	0	1	0.3
0	1	1	0.4
0	1	0	0.1
1	0	1	0.2

Table 1: Representation of $\lambda(Y, Z, W)$ - table `lambda`

It turns out that this approach is convenient for basic computations needed when working with compositional models.

3.1.1 Marginalization

It appears that marginalization procedure is a simple example of aggregation in relational databases. Indeed, to calculate $\lambda^{\downarrow Y,Z}$ from $\lambda(X,Y,Z)$ defined in Example 1, it is enough to provide the following query:

```
SELECT Y, Z, SUM(probability) AS probability
        FROM lambda
        GROUP BY (Y,Z);
```

It results in the following table

Y	Z	probability
0	0	0.90
0	1	0.10

Table 2: $\lambda^{\downarrow Y,Z}$ - table `lambda_marginal`

3.1.2 Composition operator

In case of the operator of composition, the situation is quite similar. We split the fraction from (1) into a product of two terms:

$$(\kappa \triangleright \lambda)(x,y,z,w) = \frac{\kappa(y,z,w)\lambda(x,y,z)}{\lambda^{\downarrow Y,Z}(y,z)} = \big(\kappa(y,z,w)\big)\frac{\lambda(x,y,z)}{\lambda^{\downarrow Y,Z}(y,z)}$$

The term

$$\bar{\lambda} = \frac{\lambda(x,y,z)}{\lambda^{\downarrow Y,Z}(y,z)}$$

can be computed using the following SQL query:

```
    COPY lambda INTO lambda_bar;

    UPDATE lambda_bar
        SET probability = probability / SUM(probability)
        GROUP BY (Y,Z);
```

The, using $\bar{\lambda}$, one can proceed with a standard JOIN operator. To guarantee that the operator is defined, i.e. whether $\forall y \in \mathbb{X}_{\mathbf{K}\cap\mathbf{L}} \quad \kappa^{\downarrow \mathbf{K}\cap\mathbf{L}}(y) > 0 \implies \lambda^{\downarrow \mathbf{K}\cap\mathbf{L}}(y) > 0$ it is enough to use LEFT OUTER JOIN operator and then check whether the resulting table does not contain NULL value in columns corresponding to variables $K \setminus L$. If it is the case then the composition remains undefined and the compositional process is stopped.

In case of probability distributions $\kappa(Y,Z,W)$ and $\lambda(X,Y,Z)$ the query has the following form:

```
SELECT X, Y, Z, W, kappa.probability * lambda_bar.probability AS probability
        FROM kappa LEFT OUTER JOIN lambda
        USING (Y, Z);
```

Example 2 *To illustrate the operator of composition, we will compose probability distribution κ from Table 3b with distribution λ from Example 1. The updated $\bar{\lambda}$ distribution can be found in Table 3a and the result corresponding to distribution $(\kappa \triangleright \lambda)$ is in Table 3c.*

3.1.3 Point-wise multiplication

For other more advanced operations with compositional models (like anticipating operator [2]), we also need the so-called point-wise multiplication of two probability distributions. The point-wise multiplication can be easily implemented using CROSS JOIN operator. In the case of κ and λ from previous Examples, the result is in Table 3c and the query has the following form:

```
    SELECT X,Y,Z,W kappa.probability*lambda.probability AS probability
              FROM kappa CROSS JOIN lambda
              USING (Y,Z);
```

Y	Z	W	probability
0	0	1	1.0
0	1	1	0.8
0	1	0	0.2
1	0	1	1.0

(a) $\bar{\lambda}(Y, Z, W)$ - table `lambdabar`

X	Y	Z	probability
0	0	0	0.20
0	0	1	0.15
1	0	1	0.25
1	0	0	0.40

(b) $\kappa(X, Y, Z)$ - table `kappa`

X	Y	Z	W	probability
0	0	0	1	0.20
1	0	0	1	0.40
0	0	1	1	0.12
0	0	1	0	0.03
1	0	1	1	0.20
1	0	1	0	0.05

(c) Representation of $(\kappa \triangleright \lambda)(X, Y, Z, W)$ - table `composition`

Table 3: Process of table composition

X	Y	Z	W	Probability
0	0	0	1	0.060
1	0	0	1	0.120
0	0	1	1	0.060
0	0	1	0	0.015
1	0	1	1	0.100
1	0	1	0	0.025

Table 4: $\kappa \cdot \lambda$

3.2 Compositional model

Every compositional model is fully defined using its *generating sequence* - the sequence of low-dimensional probability distributions that composed together from left to right using operator of composition $\triangleright$ create a *multidimensional probability distribution.* To represent the compositional model in a computer memory, it is enough and desirable to keep its generating sequence. In the case of e.g. marginalization, conditioning, perfectization (conversion of a model to a perfect one representing the same multidimensional distribution), etc. of the model, all computations are made locally. E.g. algorithm to perform marginalization of a compositional model locally using its generating sequence can be found in [9]. In case of conditioning, it has been proven that the conditioning process is easy if the compositional model is perfect (see [2]) and decomposable [3]. Note that decomposability is a structural property - the model is decomposable if the sequence variables meet the running intersection property.

To simplify the calculations, we keep the structure of the model aside. The reason is simple. For example, the marginalization process of a compositional model employs several heuristics that depend on structure only [10].

So far, we have implemented the following methods to work with compositional models:

- *marginalization* - removal of given variables from the given model
- *perfectization* - conversion of the model to a perfect one
- to perform *interventions*
- conversion to a *decomposable model*
- *conditioning* by a variable value

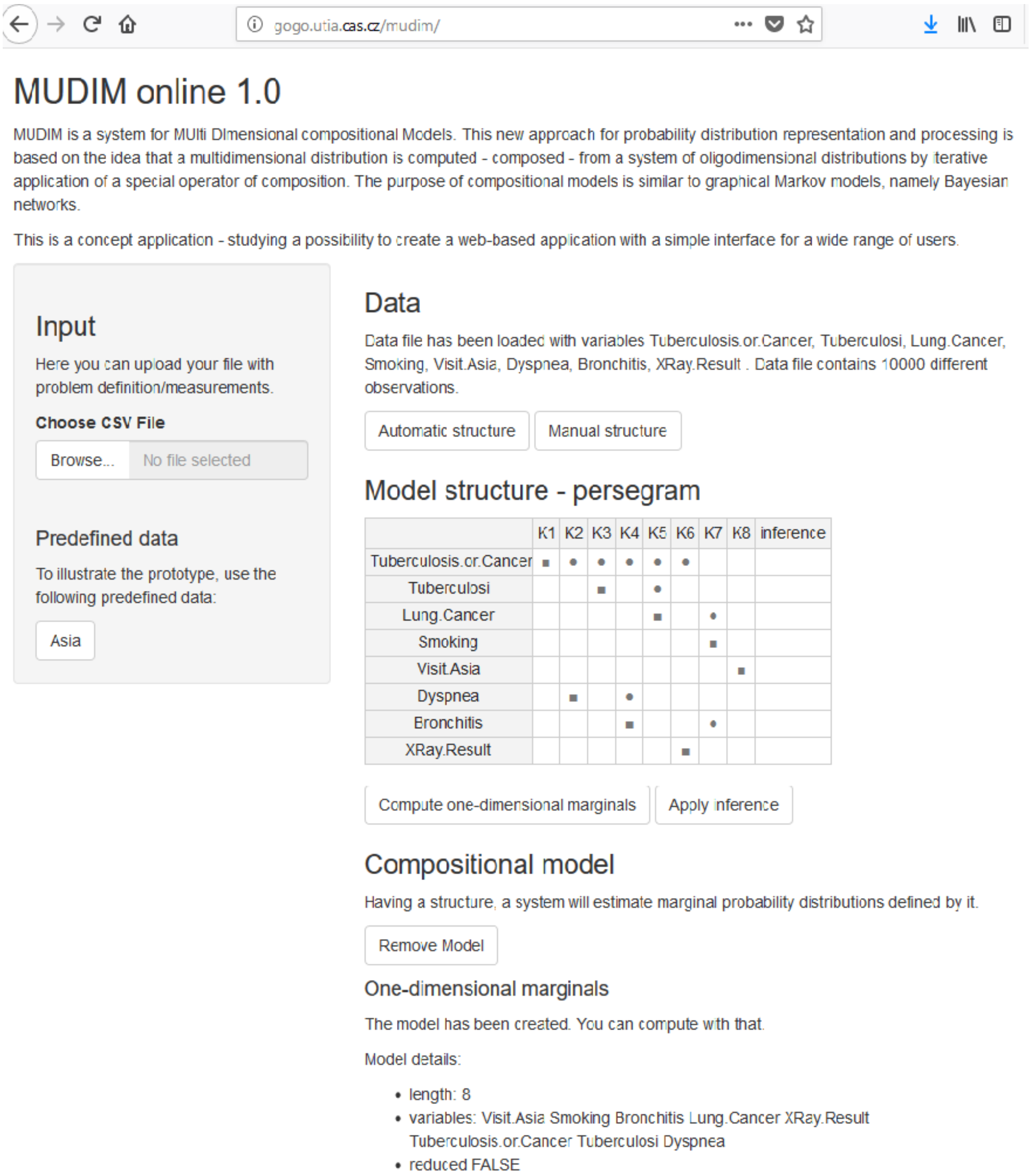
3.2.1 Learning

The first step in using the compositional model in case of data-mining is learning it from data. The learning process can be split into two parts. In the first part, a model structure is found. In this particular case, we have used hill climbing (HC) [11] algorithm. The main reason is that its implementation is a good trade-off between CPU requirements, the accuracy of the obtained model, and ease of implementation. Note that this method guarantees to obtain a minimal independence relations map and therefore it is especially appropriate to deal with high dimensional domains.

The second part of the compositional model learning process states in the estimating of the low-dimensional probabilistic distributions over a given set of variables (from the already learned model structure). In this case, we simply use frequencies of given states.

4 Web application

In this paper, we present a new semi-supervised web application called MUDIM online where acronym MUDIM stands for a system for MULti DImensional Models. Thanks to the web architecture of the system, the researchers will always have a possibility to influence the data-based model construction process from any place of the world without the need of any restriction. The application can be found at `http://gogo.utia.cas.cz/mudim`.



MUDIM online 1.0

MUDIM is a system for MUlti DImensional compositional Models. This new approach for probability distribution representation and processing is based on the idea that a multidimensional distribution is computed - composed - from a system of oligodimensional distributions by iterative application of a special operator of composition. The purpose of compositional models is similar to graphical Markov models, namely Bayesian networks.

This is a concept application - studying a possibility to create a web-based application with a simple interface for a wide range of users.

Input

Here you can upload your file with problem definition/measurements.

Predefined data

To illustrate the prototype, use the following predefined data:

Data

Data file has been loaded with variables Tuberculosis.or.Cancer, Tuberculosi, Lung.Cancer, Smoking, Visit.Asia, Dyspnea, Bronchitis, XRay.Result . Data file contains 10000 different observations.

Model structure - persegram

	K1	K2	K3	K4	K5	K6	K7	K8	inference
Tuberculosis.or.Cancer	■	●	●	●	●	●			
Tuberculosi			■		●				
Lung.Cancer					■		●		
Smoking							■		
Visit.Asia								■	
Dyspnea		■		●					
Bronchitis				■			●		
XRay.Result						■			

Compositional model

Having a structure, a system will estimate marginal probability distributions defined by it.

One-dimensional marginals

The model has been created. You can compute with that.

Model details:

- length: 8
- variables: Visit.Asia Smoking Bronchitis Lung.Cancer XRay.Result Tuberculosis.or.Cancer Tuberculosi Dyspnea
- reduced FALSE

Figure 1: The look of MUDIM online web application

Having a compositional model, a data miner can easily apply an intervention on variables of interest, fix values of other variables (conditioning), or to narrow the context of a problem (marginalization). To do that, the application expects a data file for observations/measurement at its input. The file has to be comma-separated values (CSV) file.

The basic interface of MUDIM online is visualized on Figure 1. The user can read that a data file has been loaded. The problem at hand has 8 random variables. To learn the structure of the model, the user can choose between automatic and manual mode. The structure is depicted using a persegram.

When the structure is finished, the model can be created. By creating the model we understand that corresponding low-dimensional probability distributions are estimated from data and aligned in a generating sequence in the order given by the structure. In case of the problem from Figure 1, the model has 8 low-dimensional probability distributions in its generating sequence. Reading the persegram, each distribution has dimension maximally 3. When the model is created the user is allowed to perform intervences, fix the value of a random variable by conditioning, display one-dimensional marginals (see Figure 2), etc.

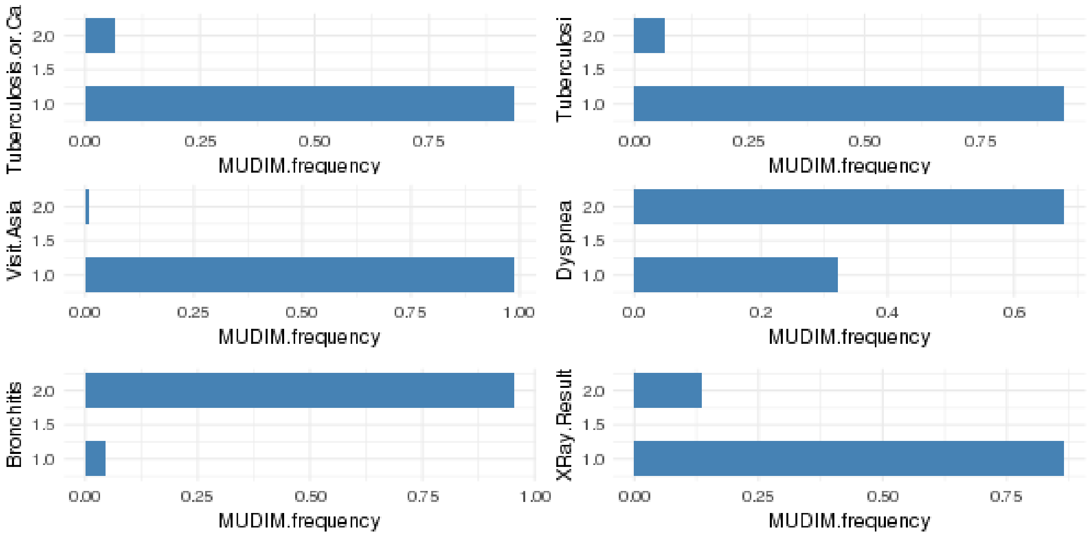


Figure 2: Visualisation of one-dimensionl marginals in MUDIM online

5 Future plans

There is a lot of work to do. First of all, we would like to implement all theoretical foundations known so far. We would like to implement more methods of learning the structure from data, information-theoretic notions like Kullback-Leibler divergence, multi-information, etc. We have several ideas on how the model learning process can be controlled to avoid overfitting of data - the system needs that for sure. Last but not least we hope that the application of this methodology to practical problems will open new problems that will be an inspiration for further theoretical research.

Acknowledgement

This work was supported by the Czech Science Foundation project 16-12010S. The authors would also like to thank the Czech Academy of Sciences and the Taiwanese Ministry of Science and Technology for their financial support of bilateral project “**Compositional Models for Data Mining**”.

References

[1] Radim Jiroušek, Václav Kratochvíl, and Tozng-Ru Lee. Compositional models for data mining: an example. In *Proceedings of the 21 Czech-Japan Seminar on Data Analysis and Decision*

Making, 2018.

[2] Radim Jiroušek. Foundations of compositional model theory. *International Journal of General Systems*, 40(6):623–678, 2011.

[3] Radim Jiroušek and Václav Kratochvíl. Foundations of compositional models: structural properties. *International Journal of General Systems*, 44(1):2–25, 2015.

[4] Radim Jiroušek. Brief introduction to causal compositional models. In *Causal Inference in Econometrics*, pages 199–211. Springer, 2016.

[5] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2018.

[6] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. *shiny: Web Application Framework for R*, 2018. R package version 1.1.0.

[7] Matt Dowle and Arun Srinivasan. *data.table: Extension of ‘data.frame‘*, 2018. R package version 1.11.4.

[8] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[9] Radim Jiroušek. Marginalization in composed probabilistic models. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 301–308. Morgan Kaufmann Publishers Inc., 2000.

[10] Vladislav Bína and Radim Jiroušek. Marginalization in multidimensional compositional models. *Kybernetika*, 42(4):405–422, 2006.

[11] José A. Gámez, Juan L. Mateo, and José M. Puerta. A fast hill-climbing algorithm for bayesian networks structure learning. In Khaled Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 585–597, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

Geometric Mean Approaches to Interval Priority Estimation and Their Comparisons

Maho Yamaguchi[1] and Masahiro Inuiguchi[2]

[1] *Graduate School of Engineering Science, Osaka University*
Toyonaka, Osaka 560-8531, Japan
yamaguchi@inulab.sys.es.osaka-u.ac.jp

[2] *Graduate School of Engineering Science, Osaka University*
Toyonaka, Osaka 560-8531, Japan
inuiguti@sys.es.osaka-u.ac.jp

Abstract

The interval AHP was proposed under the idea that the inconsistency of pairwise comparison matrix comes from human vague evaluation. Because the interval priority weights estimated by the conventional method proposed in interval AHP [3] are often unbalanced and too narrow to represent the vagueness of human judgment, several improved interval priority weight estimation methods have been proposed and shown their better performances by numerical experiments. β- and γ-relaxation methods [4] estimate interval weights by merging all sub-optimal interval priority weights with $100\beta\%$ and $100\gamma\%$ relaxation of minimal sum of widths and minimal weighted sum of widths, respectively. With proper selection of parameters β and γ, these relaxation methods perform much better than the conventional estimation method. However, proper parameter tuning may require certain computational expenses. As parameter-free methods, maximizing minimum range method (MMR method) and Maximizing minimum range method with weights λ (MMR$^\lambda$ method) were proposed. However, the estimation accuracies of MMR and MMR$^\lambda$ methods are worse than those of relaxation methods with proper parameter settings while they are better than that of the conventional method.

In numerical experiments evaluating the adequacy of preference relation executed in the previous paper [8], the crisp priority weights estimated by the classical AHP potentially performs well if a proper threshold of the difference of total scores is introduced for preference confirmation. In this paper, taking this result into consideration, we propose to utilize the geometric mean method of the classical AHP for the estimation of center values of interval priority weights. Then, the widths of interval priority weights are estimated by various interval priority weight estimation methods proposed in our previous studies [8]. The performances of the proposed methods using the geometric mean method are examined by numerical experiments. In the numerical experiments, we compare the proposed estimation methods to the previous methods from viewpoints of the estimation accuracy of the interval priority weights and the adequacy of the induced preference relation. As result, we found that the utilization of the geometric mean method for estimation of the center values of interval priority weights is advantageous. Moreover, we confirmed that by the utilization of the geometric mean method, the performances of parameter-free methods are remarkably improved so that we conclude that the parameter-free methods are useful because of their good performances without requiring any parameter tuning.

Acknowledgement
This work is supported by JSPS KAKENHI Grant Number JP17K18952.

References

[1] T. L. Saaty. *The Analytic Hierarchy Process.* McGraw-Hill, New York, 1980.

[2] T. L. Saaty and C. G. Vargas. Comparison of eigenvalue, logarithmic least squares and least squares methods in estimating ratios. *Mathematical Modelling*, vol. 5, no. 5, pp. 309–324, 1984.

[3] K. Sugihara and H. Tanaka. Interval evaluations in the analytic hierarchy process by possibility analysis. *Computational Intelligence*, vol. 17, no. 3, pp. 567–579, 2001.

[4] S. Innan and M. Inuiguchi. Modified interval weight estimation methods for Interval AHP and their comparisons. In *16th International Symposium on Advanced Intelligent Systems*, pages 1410–1417. 2015

[5] S. Innan and M. Inuiguchi. Improvement of interval weight estimation in interval AHP. In *Joint 10th International Conference on Soft Computing and Intelligent Systems and 19th International Symposium on Advanced Intelligent Systems*, pages 25–28. 2016

[6] Y. Kato. *Exposition of AHP: Fundamentals and applications* Minerva Shobo, Kyoto, 2013.

[7] M. Inuiguchi and S. Innan. Improving interval weight estimations in interval AHP by relaxations. *Advanced Computational Intelligence and Intelligent Informatics*, vol. 21, no. 7, pp. 1135–1143, 2017.

[8] M. Inuiguchi and S. Innan. Non-parametric interval weight estimation methods from a crisp pairwise comparison matrix. In *Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems*, #159, 2017.

[9] T. Entani and M. Inuiguchi. Pairwise comparison based interval analysis for group decision aiding with multiple criteria. *Fuzzy Sets and Systems*, vol. 274, pp. 79–96, 2015.

[10] H. Tanaka, K. Sugihara, and Y. Maeda. Non-additive measures by interval probability functions. *Information Sciences*, vol. 164, no. 1–4, pp. 209–227, 2004.

Compositional Models for Data Mining: an Example

Radim JIROUŠEK[1], Václav KRATOCHVÍL[1], and Tzong-Ru LEE[2]

[1] *Czech Academy of Sciences, Inst. Information Theory and Automation*
Pod Vodárenskou věží 4, CZ-182 08 Praha 8,
Czech Republic
radim@utia.cas.cz, velorex@utia.cas.cz

[2] *Department of Marketing, National Chung Hsing University*
145 Xingda Rd., South Dist., Taichung City 402,
Taiwan (R.O.C.)
trlee@dragon.nchu.edu.tw

Abstract

Like Bayesian networks, compositional models may also be used for data mining. Nevertheless, one can find several reasons why to prefer compositional models for this purpose. Perhaps the most important is the fact that compositional models are assembled from low-dimensional (unconditional) distributions so that computationally advantageous formulas are known for information theoretic characteristics computation. The other reason is that a decomposition is a natural way of complex tasks simplification. Therefore, the inverse process of composition is easily understandable for specialists from many fields of applications regardless of their level of mathematical education.

Keywords: Data mining, Mutual information, Compositional model, Conditional independence, Probability theory.

1 Introduction

The basic idea of compositional models is very simple: it is beyond human capabilities to describe global knowledge from an application area - one always works only with pieces of local knowledge. Such local knowledge can be, within probability theory, easily represented by low-dimensional distributions and a multidimensional distribution is (in a special way) composed from a number of such local pieces of knowledge – low-dimensional distributions. This analogy also explains why the compositional models are (relatively) easily understandable to specialists from the area of application - non-mathematicians. And it is also the reason why this technique can be, like Bayesian networks [1, 2], included among the methods of data-mining.

The goal of a data mining process [3] is not a model itself but its interpretation in the form of a distilled knowledge. Nevertheless, as we will see below, a greater part of knowledge is gained already during the process of model construction. The supervised approach to model construction enables the researchers to influence the resulting models in the way that these models are easily comprehensible and interpretable. Further, the user can have some knowledge about data, based on which the model is constructed. They may know that the data are not well stratified and some properties should be suppressed some others highlighted. Quite often, they want to adapt the constructed model to the purpose for which the model is constructed. Therefore, it is natural that we cannot give general instructions on how to proceed when constructing a model. It is the reason why we present in this paper just a simple example.

In this paper, we consider only finite-valued variables, which are denoted by upper-case Latin characters. Groups of variables are denoted by bold-face characters: i.e., for example, $\mathbf{M} = \{X, Y, Z, V, W\}$. The set of values of variable X is denoted by $\mathbb{X}_X$. Similarly, we use $\mathbb{X}_Y, \mathbb{X}_Z$, and so on.

By a *state* of a group of variables we understand any combination of values of the respective variables, i.e., for a group of three variables X, Z, V, a state is an element of a Cartesian product $\mathbb{X}_X \times \mathbb{X}_Z \times \mathbb{X}_V$. For the sake of simplicity, this Cartesian product is often denoted $\mathbb{X}_{\{X,Z,V\}}$. For a state $y \in \mathbb{X}_{\{X,Z,V\}}$ and a subset of the respective variables, say, for variables X, V, by $y^{\downarrow\{X,V\}}$

we denote a *projection* of y into $\mathbb{X}_{\{X,V\}}$, i.e., $y^{\downarrow\{X,V\}}$ is the state from $\mathbb{X}_{\{X,V\}}$ that is got from y by dropping out the value of variable Z.

Probability distributions are denoted by characters of Greek alphabet (κ, ν, π) (with possible indices). Recall that it means that $\kappa(Y,V) : \mathbb{X}_{\{Y,V\}} \longrightarrow [0,1]$, for which[1] $\sum_{x \in \mathbb{X}_{\{Y,V\}}} \kappa(x) = 1$.

Having a probability distribution $\kappa(X,Z,V)$, and a subset of variables $\mathbf{L} \subset \{X,Z,V\}$, $\kappa^{\downarrow\mathbf{L}}$ denote a *marginal distribution* of κ defined for each $x \in \mathbb{X}_{\mathbf{L}}$ by the formula

$$\kappa^{\downarrow\mathbf{L}}(x) = \sum_{y \in \mathbb{X}_{\mathbf{K}} : y^{\downarrow\mathbf{L}} = x} \kappa(y).$$

Note that we do not exclude situations when $\mathbf{L} = \emptyset$, for which we get $\kappa^{\downarrow\emptyset} = 1$.

Everybody knows that two variables X and Z are *independent* with respect to probability distribution $\pi(X,Z)$ if $\pi(X,Z) = \pi(X) \cdot \pi(Z)$. This is because, in this case[2],

$$\pi(X|Z) = \frac{\pi(X,Z)}{\pi(Z)} = \frac{\pi(X) \cdot \pi(Z)}{\pi(Z)} = \pi(X),$$

which expresses the fact that the knowledge of the value of variable Z does not bear any new information about the value of variable X, i.e., the conditional probability of X given Z equals the (un)conditional probability of X. For the purpose of data mining, a generalization of the notion of independence is very important. Therefore we introduce it in its general form.

Definition 1 *Consider a probability distribution* $\pi(\mathbf{N})$*, and three disjoint subsets of variables* $\mathbf{K}, \mathbf{L}, \mathbf{M}$ *(*$\mathbf{K} \cup \mathbf{L} \cup \mathbf{M} \subseteq \mathbf{N}$*). Let* $\mathbf{K}$ *and* $\mathbf{L}$ *be nonempty. We say that groups of variables* $\mathbf{K}$ *and* $\mathbf{L}$ *are* conditionally independent given $\mathbf{M}$ for distribution π *if*[3]

$$\pi^{\downarrow\mathbf{K}\cup\mathbf{L}\cup\mathbf{M}} \cdot \pi^{\downarrow\mathbf{M}} = \pi^{\downarrow\mathbf{K}\cup\mathbf{M}} \cdot \pi^{\downarrow\mathbf{L}\cup\mathbf{M}}.$$

In symbol, this property is expressed by $\mathbf{K} \perp\!\!\!\perp \mathbf{L}|\mathbf{M}\ [\pi]$.

Notice that in case of $\mathbf{M} = \emptyset$ we use only $\mathbf{K} \perp\!\!\!\perp \mathbf{L}\ [\pi]$ and speak about an unconditional independence (some authors call it marginal independence).

2 Decomposability

By a decomposition, we usually understand the result of a process that, with the goal of simplification, divides an original object into its sub-objects. Thus, for example, a problem is decomposed into two (or more) simpler sub-problems. General properties of such decomposition can be viewed on an example familiar to everybody: decomposition of a positive integer into prime numbers. In this case, an elementary decomposition is a decomposition of an integer into two factors, the product of which gives the original integer. For this example, we see that

- the result of the decomposition are two objects of the same type as the decomposed object – an integer is decomposed into two integers;
- both these sub-objects are simpler (smaller) than the original object – both factors are smaller than the original integer, we do not consider $1 \times n$ to be a decomposition of n;
- not all objects can be decomposed – prime numbers cannot be decomposed;
- there exists an inverse operation (we will call it a composition) yielding the original object from its decomposed parts – the composition of two integers is their product.

[1] Notice that symbol $\kappa(Y,V)$ is used to express the fact that probability distribution κ is defined for variables Y and V. $\kappa(x)$ for $x \in \mathbb{X}_{\{Y,V\}}$ is a probability of state $x \in \mathbb{X}_{\{Y,V\}}$.

[2] Naturally, this computation is valid only for positive distributions.

[3] This expression means that for all $x \in \mathbb{X}_{\mathbf{K}\cup\mathbf{L}\cup\mathbf{M}}$

$$\pi^{\downarrow\mathbf{K}\cup\mathbf{L}\cup\mathbf{M}}(x) \cdot \pi^{\downarrow\mathbf{M}}(x^{\downarrow\mathbf{M}}) = \pi^{\downarrow\mathbf{K}\cup\mathbf{M}}(x^{\downarrow\mathbf{K}\cup\mathbf{M}}) \cdot \pi^{\downarrow\mathbf{L}\cup\mathbf{M}}(x^{\downarrow\mathbf{L}\cup\mathbf{M}}).$$

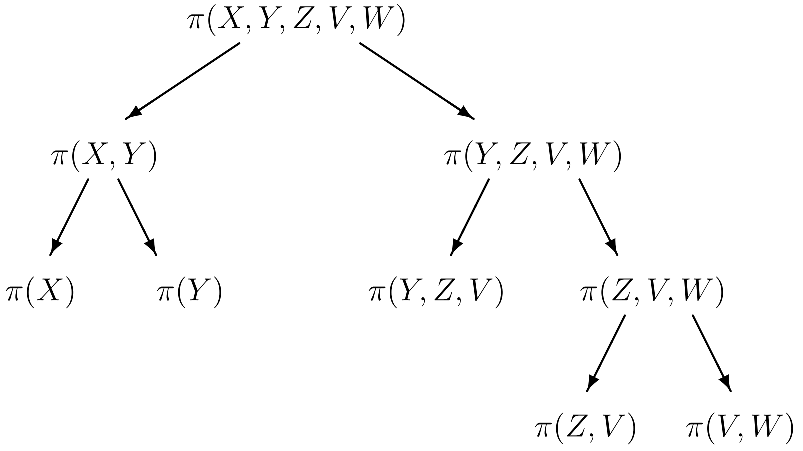


Figure 1: Hierarchical decomposition of $\pi(X,Y,Z,V,W)$.

What is a decomposition of a finite probability distribution? Consider a two-dimensional distribution $\pi(X,Y)$. Simpler sub-objects are just one-dimensional distributions: a distribution of a variable X and a distribution of a variable Y. To have a chance to reconstruct the original two-dimensional distribution π from these one-dimensional distributions, we have to consider marginals of π: $\pi^{\downarrow X}$ and $\pi^{\downarrow Y}$. Generally, the process of marginalization is unique, but, with the exception of a degenerate distribution, we cannot unambiguously reconstruct the original two-dimensional distribution from its one-dimensional marginals. To bypass this fact, we restrict the decomposition of two-dimensional distributions $\pi(X,Y)$ into their one-dimensional marginals only for the case of independence: $X \perp\!\!\!\perp Y\ [\pi]$. In this case, $\pi(X,Y)$ can easily be reconstructed from its marginals $\pi^{\downarrow X}$ and $\pi^{\downarrow Y}$: $\pi(X,Y) = \pi^{\downarrow X} \cdot \pi^{\downarrow Y}$, where "$\cdot$" denotes pointwise multiplication, i.e., $\pi(x,y) = \pi^{\downarrow X}(x) \cdot \pi^{\downarrow Y}(y)$ for all states $(x,y) \in \mathbb{X}_{\{X,Y\}}$.

Analogously, three-dimensional distribution $\pi(X,Y,Z)$ can be decomposed into two simpler probability distributions (marginals of $\pi(X,Y,Z)$) only if either a couple of variables (say X,Y) is independent of the remaining third variable (in this case Z), or, if two variables (say X and Z) are conditionally independent given the remaining third variable (in this case Y):

- $\{X,Y\} \perp\!\!\!\perp Z\ [\pi]$, then $\pi(X,Y,Z)$ can be reconstructed from $\pi^{\downarrow\{X,Y\}}$ and $\pi^{\downarrow Z}$,
- $X \perp\!\!\!\perp Z|Y\ [\pi]$, then $\pi(X,Y,Z)$ can be reconstructed from $\pi^{\downarrow\{X,Y\}}$ and $\pi^{\downarrow\{Y,Z\}}$.

This leads us to the following general definition.

Definition 2 *We say that a probability distribution* $\pi(\mathbf{M})$ *is* decomposed *into its marginals* $\pi^{\downarrow\mathbf{K}}$ *and* $\pi^{\downarrow\mathbf{L}}$ *if*

1. $\mathbf{K} \cup \mathbf{L} = \mathbf{M}$*;*

2. $\mathbf{K} \neq \mathbf{M}$*,* $\mathbf{L} \neq \mathbf{M}$*;*

3. $\pi(\mathbf{M}) \cdot \pi^{\downarrow\mathbf{K}\cap\mathbf{L}} = \pi^{\downarrow\mathbf{K}} \cdot \pi^{\downarrow\mathbf{L}}$.

Notice that the third condition is nothing else than $\mathbf{K}\setminus\mathbf{L} \perp\!\!\!\perp \mathbf{L}\setminus\mathbf{K}|\mathbf{K}\cap\mathbf{L}\ [\pi]$, and that the original distribution $\pi(\mathbf{M})$ can be uniquely reconstructed from the marginals $\pi^{\downarrow\mathbf{K}}$ and $\pi^{\downarrow\mathbf{L}}$.

Analogously to the decomposition of integers to prime numbers, even probability distributions can be hierarchically decomposed into a system of distributions that cannot be further decomposed. An example of such a hierarchical process represented by a corresponding tree structure can be seen in Figure 1, where distribution $\pi(X,Y,Z,V,W)$ is decomposed into a system of its marginal distributions: $\pi^{\downarrow X}$, $\pi^{\downarrow Y}$, $\pi^{\downarrow\{Y,Z,V\}}$, $\pi^{\downarrow\{Z,V\}}$, $\pi^{\downarrow\{V,W\}}$. Each decomposition was made possible by the fact that the respective conditional independence relation holds for distribution π. The decomposition process from Figure 1 was made possible by the assumption that the following system of conditional relations holds for distribution π (or, in other words, that the independence structure [4] of distribution π is the following):

- $X \perp\!\!\!\perp \{Z, V, W\}|Y\ [\pi]$;
- $X \perp\!\!\!\perp Y\ [\pi]$;
- $Y \perp\!\!\!\perp W|\{Z, V\}\ [\pi]$;
- $Z \perp\!\!\!\perp W|V\ [\pi]$.

Definition 3 *A probability distribution $\pi(\mathbf{N})$ is said to be* decomposable *if it can be decomposed into a system of its marginals $\pi^{\downarrow \mathbf{M}_1}, \pi^{\downarrow \mathbf{M}_2}, \ldots, \pi^{\downarrow \mathbf{M}_m}$, such that the variable sets $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_m$ can be reordered so that they meet the* Running Intersection Property *(RIP):*

$$\forall j = 2, 3, \ldots, m \ \ \exists k (1 \le k < j) \text{ for which } \mathbf{M}_j \cap (\mathbf{M}_1 \cup \ldots \cup \mathbf{M}_{j-1}) \subseteq \mathbf{M}_k.$$

3 Compositional models

This section introduces an operator of composition, originally introduced in [5], which realizes a process inverse to the process of decomposition discussed in previous section. For this, we need a notion of a *dominance*: $\pi(N) \ll \nu(N)$ if

$$\forall y \in \mathbb{X}_{\mathbf{N}} \ \ \pi(y) > 0 \implies \nu(y) > 0.$$

Definition 4 *For arbitrary two distributions $\kappa(\mathbf{K})$ and $\lambda(\mathbf{L})$, for which $\kappa^{\downarrow \mathbf{K} \cap \mathbf{L}} \ll \lambda^{\downarrow \mathbf{K} \cap \mathbf{L}}$, their* composition *is for each $x \in \mathbb{X}_{\mathbf{K} \cup \mathbf{L}}$ given by the following formula*[4]

$$(\kappa \rhd \lambda)(x) = \frac{\kappa(x^{\downarrow \mathbf{K}})\lambda(x^{\downarrow \mathbf{L}})}{\lambda^{\downarrow \mathbf{K} \cap \mathbf{L}}(x^{\downarrow \mathbf{K} \cap \mathbf{L}})}.$$

In case that $\kappa^{\downarrow \mathbf{K} \cap \mathbf{L}} \not\ll \lambda^{\downarrow \mathbf{K} \cap \mathbf{L}}$, the composition remains undefined.

The reader certainly noticed that the presented definition is slightly more general than just an inverse operation to decomposition discussed above. We do not require that both $\mathbf{K}$ and $\mathbf{L}$ are proper subsets of $\mathbf{K} \cup \mathbf{L}$. The main reason is that this generalization makes the formulation of some theoretical properties simpler. Moreover, abandoning this requirement appears advantageous when constructing compositional models and when reading a knowledge from the resulting models. As we will see in Section 5, it enables the user to specify the required relations of (conditional) independence, which would not be otherwise representable in a model.

This operator of composition enables us to set up *multidimensional compositional models*, i.e. multidimensional probability distributions assembled from sequences of low-dimensional distributions with the help of the operators of composition [6, 7, 8]. Considering a systems of low-dimensional distributions $\kappa_1(\mathbf{K}_1)$, $\kappa_2(\mathbf{K}_2), \ldots, \kappa_n(\mathbf{K}_n)$, the formula $\kappa_1 \rhd \kappa_2 \rhd \ldots \rhd \kappa_n$, if defined, specifies a distribution of variables $\mathbf{K}_1 \cup \mathbf{K}_2 \cup \ldots \cup \mathbf{K}_n$. However, because of the fact that the operator of composition is not associative, the order, in which the operators are performed in the expression $\kappa_1 \rhd \kappa_2 \rhd \ldots \rhd \kappa_n$ should be specified by parentheses. To simplify such expressions, we will omit the parentheses if the operators are to be performed from left to right. Therefore

$$\kappa_1 \rhd \kappa_2 \rhd \ldots \rhd \kappa_n = (\ldots((\kappa_1 \rhd \kappa_2) \rhd \kappa_3) \rhd \ldots \rhd \kappa_{n-1}) \rhd \kappa_n.$$

Moreover, without loss of generality, in what follows, we will always assume that κ_i is a distribution of variables $\mathbf{K}_i$ and that the composition will be defined in all the formulas wherever the operator appears.

To visualize the structure of a compositional model we use a tool called a *persegram*.

Definition 5 *A* persegram *of a compositional model $\kappa_1 \rhd \kappa_2 \rhd \ldots \rhd \kappa_n$ is a table in which rows correspond to variables from $\mathbf{K}_1 \cup \ldots \cup \mathbf{K}_n$ (in an arbitrary order) and columns correspond to distributions $\kappa_1, \ldots, \kappa_n$ in the respective ordering. A position in the table is marked if the variable is among the arguments of the respective distribution. Markers for the first occurrence of each variable (i.e., the leftmost markers in rows) are* box-markers*, and for other occurrences there are* bullets*.*

[4] Define $\frac{0 \cdot 0}{0} = 0$.

For an example, the reader is referred to Figure 2a in Section 5, in which the persegram of $\kappa_1(D,N) \triangleright \kappa_2(B,R) \triangleright \kappa_3(R,W) \triangleright \kappa_4(N,R) \triangleright \kappa_5(T,W)$ is depicted.

Persegrams were designed mainly for reading conditional independence relations holding for compositional models. For this, we have to learn what are the trails and avoiding trails in a persegram.

Definition 6 *A sequence of markers* $\mathfrak{m}_0, \dots, \mathfrak{m}_t$ *in the persegram of a compositional model* $\kappa_1 \triangleright \kappa_2 \triangleright \dots \triangleright \kappa_n$ *is called an* **M**-avoiding trail ($\mathbf{M} \subseteq \mathbf{K}_1 \cup \mathbf{K}_2 \cup \dots \cup \mathbf{K}_n$) that connects m_0 and m_t *if it meets the following five conditions:*

1. *neither* $\mathfrak{m}_0$ *nor* $\mathfrak{m}_t$ *corresponds to a variable from* **M***;*
2. *for each* $s = 1, \dots, t$*, the couple* $(\mathfrak{m}_{s-1}, \mathfrak{m}_s)$ *is either in the same row (i.e., a horizontal connection) or in the same column (a vertical connection);*
3. *each vertical connection must be adjacent to a box-marker (i.e., at least one of the markers in the vertical connection is a box-marker) - the so-called regular vertical connection;*
4. *no horizontal connection corresponds to a variable from* **M***;*
5. *vertical and horizontal connections regularly alternate with the following possible exception: at most, two vertical connections may be in direct succession if their common adjacent marker is a box-marker of a variable from* **M**.

If an **M***-avoiding trail connects two markers corresponding to variables* X *and* Y*, we say that these* variables are connected by an **M**-avoiding trail*. This situation is denoted by* $X \leftrightsquigarrow_{\mathbf{M}} Y \ [\kappa_1 \triangleright \kappa_2 \triangleright \dots \triangleright \kappa_n]$*. Symbol* $X \not\leftrightsquigarrow_{\mathbf{M}} Y \ [\kappa_1 \triangleright \kappa_2 \triangleright \dots \triangleright \kappa_n]$ *denote the situation when there does not exist an* **M***-avoiding trail connecting variables* X *and* Y *in the corresponding persegram. If* $\mathbf{M} = \emptyset$ *we speak about a simple trail, and use simplified symbol* $X \leftrightsquigarrow Y$*.*

For a simple trail (i.e. $\emptyset$-avoiding trail) connecting variables B and T see Figure 2b. A relationship between the existence of avoiding trails and the conditional independence of variables in a compositional model is expressed in the following assertion, which was originally proven in [9] (an alternative proof was published in [10]).

Theorem 7 *Consider a compositional model* $\kappa_1, \kappa_2, \dots, \kappa_n$ *and the corresponding persegram. Let* X *and* Y *be two different variables from* $\mathbf{K}_1 \cup \mathbf{K}_2 \cup \dots \cup \mathbf{K}_n$*, and* $\mathbf{M} \subseteq \mathbf{K}_1 \cup \mathbf{K}_2 \cup \dots \cup \mathbf{K}_n \setminus \{X, Y\}$*. Then*

$$X \not\leftrightsquigarrow_{\mathbf{M}} Y \ [\kappa_1, \kappa_2, \dots, \kappa_n] \implies X \perp\!\!\!\perp Y | \mathbf{M} \ [\kappa_1, \kappa_2, \dots, \kappa_n].$$

4 Information-theoretic notions

In this section we consider a probability distribution $\pi(\mathbf{N})$, and three disjoint subsets $\mathbf{K}, \mathbf{L}, \mathbf{M} \subset \mathbf{N}$, such that $\mathbf{K} \cup \mathbf{L} \cup \mathbf{M} = \mathbf{N}$. Moreover, we assume that $\mathbf{K}$ and $\mathbf{L}$ are nonempty.

The basic notion, from which all the remaining ones are derived, is the famous Shannon entropy defined

$$H(\pi) = -\sum_{x \in \mathbb{X}_{\mathbf{N}} : \pi(x) > 0} \pi(x) \log_2 \pi(x).$$

This concept measures an uncertainty connected with the probability distribution. Its value is always nonnegative, less or equal $\log_2 |\mathbb{X}_{\mathbf{N}}|$. It equals zero if and only if the distribution is degenerated and expresses certainty. In other words, $H(\pi)$ equals zero, if and only if there exists a state $x^* \in \mathbb{X}_{\mathbf{N}}$, for which $\pi(x^*) = 1$. The entropy achieves its maximum only for a uniform distribution, i.e.,

$$H(\pi) = \log_2 |\mathbb{X}_{\mathbf{N}}| \iff \pi(x) = \frac{1}{|\mathbb{X}_{\mathbf{N}}|} \quad \text{for all } x \in \mathbb{X}_{\mathbf{N}}.$$

To measure the strength of dependence between the groups of random variables we employ a notion of *mutual information* defined by the formula

$$MI_{\pi}(\mathbf{K}; \mathbf{L}) = \sum_{x \in \mathbb{X}_{\mathbf{K} \cup \mathbf{L}} : \pi^{\downarrow \mathbf{K} \cup \mathbf{L}}(x) > 0} \pi^{\downarrow \mathbf{K} \cup \mathbf{L}}(x) \log_2 \left(\frac{\pi^{\downarrow \mathbf{K} \cup \mathbf{L}}(x)}{\pi^{\downarrow \mathbf{K}}(x^{\downarrow \mathbf{K}}) \cdot \pi^{\downarrow \mathbf{L}}(x^{\downarrow \mathbf{L}})} \right).$$

The higher this value, the stronger dependence exists between two disjoint groups of variables: $\mathbf{K}$ and $\mathbf{L}$. If the reader likes, this property can also be expressed in another way. The higher this value, the more information about variables $\mathbf{K}$ we get when learning values of variables $\mathbf{L}$ (or equivalently, because $MI_\pi(\mathbf{K};\mathbf{L}) = MI_\pi(\mathbf{L};\mathbf{K})$, the more information about variables $\mathbf{L}$ we get when learning values of variables $\mathbf{K}$).

Let us summarize the most important properties of mutual information supporting the fact that it is used as the measure of the *strength* of the dependence.

- $0 \le MI_\pi(\mathbf{K};\mathbf{L}) \le \min(H(\pi^{\downarrow\mathbf{K}}), H(\pi^{\downarrow\mathbf{L}}))$.
- $MI_\pi(\mathbf{K};\mathbf{L}) = 0 \iff \mathbf{K} \perp\!\!\!\perp \mathbf{L}\ [\pi]$.
- $MI_\pi(\mathbf{K};\mathbf{L}) = H(\pi^{\downarrow\mathbf{K}})$ if and only if variables $\mathbf{K}$ are functionally dependent on variables $\mathbf{L}$. It means that in this case for the conditional distribution $\pi^{\mathbf{K}|\mathbf{L}}$ it holds that

$$\forall y \in \mathbb{X}_{\mathbf{L}}\ \exists x \in \mathbb{X}_{\mathbf{K}} \text{ such that } \pi^{\mathbf{K}|\mathbf{L}}(x|y) = 1.$$

In many practical situations, it is useful to normalize the measure of mutual information, to get a measure achieving values from the interval $[0,1]$. This value suggested by A. Perez [11], who called it *information measure of dependence*, is in this text denoted ID:

$$ID_\pi(\mathbf{K};\mathbf{L}) = \frac{MI_\pi(\mathbf{K};\mathbf{L})}{\min(H(\pi^{\downarrow\mathbf{K}}), H(\pi^{\downarrow\mathbf{L}}))}.$$

It may help the reader to understand the notion of mutual information, if we show that it is actually the measure of similarity of two distributions. In probability theory, several measures of similarity for distributions have been introduced. One of them, having its origin in information theory, is a Kullback-Leibler divergence defined for $\pi(\mathbf{N})$ and $\nu(\mathbf{N})$ by the formula

$$Div(\pi \shortparallel \nu) = \begin{cases} \sum\limits_{x \in \mathbb{X}_N : \pi(x) > 0} \pi(x) \log_2\left(\frac{\pi(x)}{\nu(x)}\right), & \text{if } \pi \ll \nu; \\ +\infty, & \text{otherwise.} \end{cases}$$

It is known that Kullback-Leibler divergence is always nonnegative and equals 0 if and only if $\pi = \nu$ (see [12, 13]). Its only disadvantage is that it is not symmetric, i.e., generally $Div(\pi \shortparallel \nu) \neq Div(\nu \shortparallel \pi)$. Nevertheless, since it is very easy to show that $\pi^{\downarrow\mathbf{K}\cup\mathbf{L}} \ll \pi^{\downarrow\mathbf{K}} \cdot \pi^{\downarrow\mathbf{L}}$ we see that

$$MI_\pi(\mathbf{K};\mathbf{L}) = Div(\pi^{\downarrow\mathbf{K}\cup\mathbf{L}} \shortparallel \pi^{\downarrow\mathbf{K}} \cdot \pi^{\downarrow\mathbf{L}})$$

is always finite, and, as we already said above, equals zero if and only if $\pi^{\downarrow\mathbf{K}\cup\mathbf{L}} = \pi^{\downarrow\mathbf{K}} \cdot \pi^{\downarrow\mathbf{L}}$, which is nothing else than $\mathbf{K} \perp\!\!\!\perp \mathbf{L}\ [\pi]$.

As the reader can expect, not only there is a relationship between independence and mutual information, but there is also an analogous relationship between conditional independence and *conditional mutual information*, defined by

$$MI_\pi(\mathbf{K};\mathbf{L}|\mathbf{M}) = \sum_{x \in \mathbb{X}_{\mathbf{N}} : \pi(x) > 0} \pi(x) \log_2\left(\frac{\pi^{\mathbf{K}\cup\mathbf{L}|\mathbf{M}}(x)}{\pi^{\mathbf{K}|\mathbf{M}}(x^{\downarrow\mathbf{K}\cup\mathbf{M}}) \cdot \pi^{\mathbf{L}|\mathbf{M}}(x^{\downarrow\mathbf{L}\cup\mathbf{M}})}\right).$$

(Notice that $MI_\pi(\mathbf{K};\mathbf{L}|\emptyset) = MI_\pi(\mathbf{K};\mathbf{L})$.)

Again, the higher the value of conditional mutual information the stronger the conditional dependence between the respective groups of variables. Since we have not introduced the notion of conditional entropy, in this case, we can precisely formulate only a part of the properties holding for conditional mutual information.

- $MI_\pi(\mathbf{K};\mathbf{L}|\mathbf{M}) \ge 0$.
- $MI_\pi(\mathbf{K};\mathbf{L}|\mathbf{M}) = 0 \iff \mathbf{K} \perp\!\!\!\perp \mathbf{L}|\mathbf{M}\ [\pi]$.

5 Data mining example

This is the main section of the paper presenting a supervised model construction, during which we gain a knowledge from data. We consider six variables $\mathbf{M} = \{B, D, N, R, T, W\}$ with $\mathbb{X}_B = \{1, 2, 3\}$ and $\mathbb{X}_D = \mathbb{X}_N = \mathbb{X}_R = \mathbb{X}_T = \mathbb{X}_W = \{1, 2\}$. We are about to construct a compositional model for these variables from a data file containing 1000 records. Taking into account the fact that the cardinality of the considered state space is $|\mathbb{X}_\mathbf{M}| = 3 \times 2^5 = 96$, we can hardly expect to get any reasonable (i.e., interpretable) knowledge from the respective frequency table depicted in Table 1.

Table 1: Frequencies of states from $\mathbb{X}_{\{B,D,N,R,T,W\}}$.

	$R = 0$				$R = 1$			
	$T = 0$		$T = 1$		$T = 0$		$T = 1$	
	$W = 0$	$W = 1$	$W = 0$	$W = 1$	$W = 0$	$W = 1$	$W = 0$	$W = 1$
$B = 1,\ D = 1,\ N = 1$	0	8	4	15	2	9	23	3
$B = 1,\ D = 1,\ N = 2$	0	0	0	0	0	1	3	0
$B = 1,\ D = 2,\ N = 1$	0	0	0	0	1	3	2	0
$B = 1,\ D = 2,\ N = 2$	0	147	12	66	5	9	1	0
$B = 2,\ D = 1,\ N = 1$	0	2	0	10	10	34	70	0
$B = 2,\ D = 1,\ N = 2$	0	10	0	7	3	8	1	2
$B = 2,\ D = 2,\ N = 1$	0	0	0	0	1	6	13	0
$B = 2,\ D = 2,\ N = 2$	0	61	4	31	14	45	22	1
$B = 3,\ D = 1,\ N = 1$	0	0	0	4	20	40	78	4
$B = 3,\ D = 1,\ N = 2$	0	4	0	2	1	9	3	0
$B = 3,\ D = 2,\ N = 1$	0	0	1	0	3	5	13	0
$B = 3,\ D = 2,\ N = 2$	0	13	1	7	23	57	20	0

It is not a bad idea to start with computing the value of entropy for all considered variables:

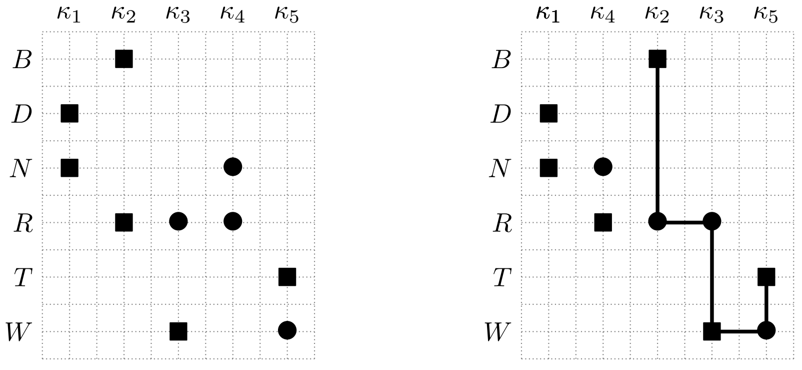
$$\begin{array}{lll} H(B) = 1.58, & H(D) = 0.98, & H(N) = 0.96, \\ H(R) = 0.99, & H(T) = 0.99, & H(W) = 0.93. \end{array}$$

From this, we do not get any knowledge about the relationship among the considered variables, but we get some information as for how to proceed further. Since the entropy of all binary variables is close to 1, it means that a minimum of entropies for any pair of variables is close to one, either. Therefore, when considering a strength of dependence between two variables, the value of mutual information MI and the value of information measure of dependence ID do not significantly differ from each other. Therefore we compute only values of mutual information. But keep in mind that when the considered variables achieve different numbers of values, there may be substantial differences between the values of the entropy of individual variables. In such a case, considering the information measure of dependence is preferable.

From the point of view of model construction, we are interested in couples of variables, which are closely (strongly) connected, and in couples of independent variables. Therefore, when computing values of mutual information for all pairs of variables, we sort the couples according to the value of mutual information. In the present example, we get

$$\begin{array}{ll} \left\{\begin{array}{l} MI(D;N) = 0.4356, \\ MI(B;R) = 0.2871, \\ MI(R;W) = 0.2578, \\ MI(N;R) = 0.2070, \\ MI(T;W) = 0.1813, \end{array}\right. & \begin{array}{l} MI(N;T) = 0.0709 \\ MI(D;W) = 0.0627 \\ MI(B;N) = 0.0619 \\ MI(D;T) = 0.0421 \end{array} \\ \quad MI(N;W) = 0.1546 & \quad MI(B;D) = 0.0342 \\ \quad MI(D;R) = 0.0958 & \left\{\begin{array}{l} MI(R;T) = 0.0019, \\ MI(B;T) = 0.0007. \end{array}\right. \\ \quad MI(B;W) = 0.0814 & \end{array}$$

The head of this sequence contains the couples of closely connected variables, the tail of this sequence suggests which pairs of variables may be considered independent. The first five couples are grouped together because these first five couples cover the whole $\mathbf{M}$. Therefore, let us start building compositional models from two-dimensional distributions defined for these couples of variables. To get their best ordering in a model, the multi-information of the whole model should be taken into account. The higher multi-information, the better model because it incorporates more information from data.



(a) $\pi_1 = \kappa_1 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4 \triangleright \kappa_5$ (b) $\pi_2 = \kappa_1 \triangleright \kappa_4 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_5$

Figure 2: Persegrams of models π_1 and π_2.

Consider estimates of the first five two-dimensional distributions and denote them respectively: $\kappa_1(D,N)$, $\kappa_2(B,R)$, $\kappa_3(R,W)$, $\kappa_4(N,R)$, $\kappa_5(T,W)$. If considering model $\pi_1 = \kappa_1 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4 \triangleright \kappa_5$ (see persegram in Figure 2a) we can immediately see that $\pi_1 = \kappa_1 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_5$. Distribution κ_4 may be deleted from the model because both the respective markers in the persegram are bullets. This model is decomposable (the reader may easily check RIP) and perfect (data file does not contain missing values, and therefore the estimates of marginals are consistent). Therefore[5],

$$\begin{aligned} IC(\pi_1) = IC(\kappa_1) + IC(\kappa_2) - IC(\kappa_2^{\downarrow\emptyset}) + IC(\kappa_3) - IC(\kappa_3^{\downarrow R}) \\ +IC(\kappa_5) - IC(\kappa_5^{\downarrow W}) = \sum_{i=1,2,3,5} IC(\kappa_i) = 1.1618, \end{aligned}$$

because $IC(\kappa_1) = MI(D;N), IC(\kappa_2) = MI(B;R)$, and so on, and because the multi-information of probability distribution $\kappa(\mathbf{K})$ for $|\mathbf{K}| < 2$ equals zero. However, it is evident that also $\pi_2 = \kappa_1 \triangleright \kappa_4 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_5$ is a decomposable model, for which

$$IC(\pi_2) = \sum_{i=1}^{5} IC(\kappa_i) = 1.3687.$$

In fact, this model is the best possible among those assembled from distributions $\kappa_1, \kappa_2, \kappa_3, \kappa_4, \kappa_5$, in case the amount of multi-information is taken as the only criterion of optimality. This is because this model utilizes all the information contained in the distributions from which it is assembled. However, this model does not reflect the other information we obtained from computing the mutual information for all couples of variables: the two smallest values of mutual information suggest that variables T and R, and variables T and B are independent. And, as the reader can deduce from the persegram corresponding to π_2 (see persegram in Figure 2b), one can find simple trails connecting all couples of variables, i.e., also $B \leftrightsquigarrow T\ [\pi_2]$ and $R \leftrightsquigarrow T\ [\pi_2]$. Therefore, the independence relations $B \perp\!\!\!\perp T\ [\pi_2]$ and $R \perp\!\!\!\perp T\ [\pi_2]$ are not guaranteed by the model structure.

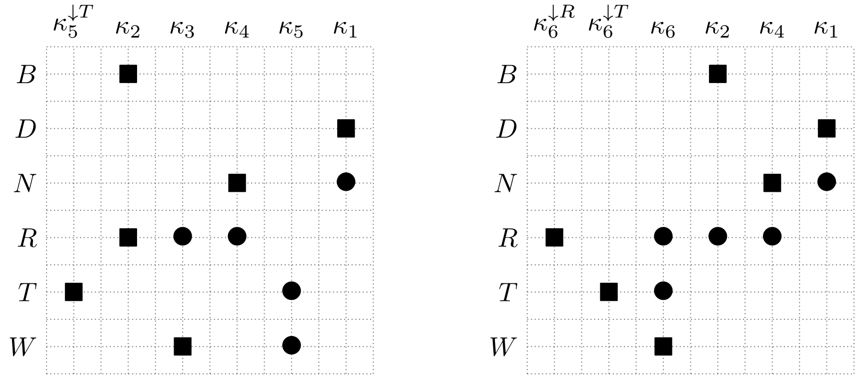
To incorporate this knowledge into the model, one can consider, e.g., model $\pi_3 = \kappa_5^{\downarrow T} \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4 \triangleright \kappa_5 \triangleright \kappa_1$. However, as the reader can see from the persegram in Figure 3a, $\pi_3 = \kappa_5^{\downarrow T} \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4 \triangleright \kappa_1$, and therefore

$$IC(\pi_3) = \sum_{i=1}^{4} IC(\kappa_i) = 1.1874.$$

The decrease of the multi-information is due to the fact that π_3 dos not incorporate the information from κ_5.

Thus, it may seem that one can incorporate the knowledge about the two independence relations into the model only at the cost of a decrease of multi-information, i.e., at the cost of the loss of

[5] Such a simple formula holds only for decomposable models.



	$\kappa_5^{\downarrow T}$	κ_2	κ_3	κ_4	κ_5	κ_1
B		■				
D						■
N				■		●
R		■	●	●		
T	■				●	
W			■		●	

(a) $\pi_3 = \kappa_5^{\downarrow T} \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4 \triangleright \kappa_5 \triangleright \kappa_1$

	$\kappa_6^{\downarrow R}$	$\kappa_6^{\downarrow T}$	κ_6	κ_2	κ_4	κ_1
B				■		
D						■
N					■	●
R	■		●	●	●	
T		■	●			
W			■			

(b) $\pi_5 = \kappa_6^{\downarrow R} \triangleright \kappa_6^{\downarrow T} \triangleright \kappa_6 \triangleright \kappa_2 \triangleright \kappa_4 \triangleright \kappa_1$

Figure 3: Persegrams of models π_3 and π_5.

information. To get out of this trap, let us start studying the way, how variable T is connected with all others. Let us compute from the data the conditional mutual information of T and B given the remaining variables, and similarly, the conditional mutual information of T and R given the remaining variables. We get

$$\begin{aligned} MI(T;B|D) &= 0.002, & MI(T;R|D) &= 0.001,\\ MI(T;B|N) &= 0.006, & MI(T;R|N) &= 0.013,\\ MI(T;B|W) &= 0.024, & MI(T;R|W) &= 0.084. \end{aligned}$$

How to explain the fact that T and R are independent but not conditionally independent? A straightforward explanation is that T and R are independent and jointly influence other variables. In case we know the meaning of the variables, we should choose the one, which is, in our knowledge, directly influenced by T and R (or T and B). Otherwise, we choose the one indicated by the highest value of conditional mutual information: $MI(T;R|W)$. It makes us believe that two independent variables T and R influence W, and the only way how to incorporate this knowledge into the model is to start considering a three-dimensional distribution: let $\kappa_6(R,T,W)$ be the corresponding estimate got from data. Naturally, this three-dimensional distribution is a bearer of all the information expressed by both κ_3 and κ_5, which can be now dropped off from the further consideration. Naturally, κ_6 contains more information than κ_3 and κ_5. It describes the combined influence of T and R on W, which cannot be expressed by two two-dimensional distributions.[6]

After adding κ_6 and deleting κ_3 and κ_5, the remaining distributions $\kappa_1, \kappa_2, \kappa_4, \kappa_6$ can easily be ordered to meet RIP: e.g., $\pi_4 = \kappa_6 \triangleright \kappa_2 \triangleright \kappa_4 \triangleright \kappa_1$ is a decomposable model expressing all the knowledge we consider. Nevertheless, the above discussed independence of variables is not visible from the respective persegram, it is only encoded in the distribution κ_6. Therefore, we can prefer model $\pi_5 = \kappa_6^{\downarrow B} \triangleright \kappa_6^{\downarrow T} \triangleright \kappa_6 \triangleright \kappa_2 \triangleright \kappa_4 \triangleright \kappa_1$, from the persegram of which in Figure 3b the considered independence relations are obvious.

What are the differences between the models π_4 and π_5? Model π_4 is decomposable, and therefore more advantageous when used for computations. On the other hand, model π_5 explicitly manifests the independence $T \perp\!\!\!\perp \{R,B\}|W\ [\pi_5]$. When computing the multi-information of these models we get

$$IC(\pi_4) = \sum_{i=6,2,4,1} IC(\kappa_i) = 0.5234 + 0.2871 + 0.2070 + 0.4356 = 1.4531,$$

and

$$IC(\pi_5) = \sum_{i=6,2,4,1} IC(\kappa_i) - IC(\kappa_6^{\downarrow\{R,T\}}) = IC(\pi_5) - MI(T,R) = 1.4512.$$

[6] To illustrate the fact that a three-dimensional distribution may bear more information than a collection if its two-dimensional marginals, consider the following simple example. Children have usually more fun if the weather is warm. Similarly, they prefer sunny days to days with precipitation. However, in winter, the precipitation in very cold days usually means snowing, which is a great fun for children. And this type of knowledge cannot be expressed just by describing two separate relations: day temperature and children fun, and precipitation and children fun.

The imperceptible decrease of the value of multi-information when transforming π_4 into π_5 is due to small changes necessary for introducing the independence of T and R.

Model π_5 seems to meet all the requirements made for data-based models. Nevertheless, especially when considering supervised approaches, one should not miss the realization of important subsequent steps belonging to a process of model verification rather than to the process of model construction. Let us illustrate these steps by verifying model π_5. Consider the respective persegram in Figure 3b, which enables us to list all (conditional) independence relations holding for the model:

1. $B \perp\!\!\!\perp D|\mathbf{M}$ for $\mathbf{M}$ containing either N or R,
2. $B \perp\!\!\!\perp N|\mathbf{M}$ for $R \in \mathbf{M}$,
3. $B \perp\!\!\!\perp T|\mathbf{M}$ for $R \in \mathbf{M}$, or $W \notin \mathbf{M}$,
4. $B \perp\!\!\!\perp W|\mathbf{M}$ for $R \in \mathbf{M}$,
5. $D \perp\!\!\!\perp R|\mathbf{M}$ for $N \in \mathbf{M}$,
6. $D \perp\!\!\!\perp T|\mathbf{M}$ for $N \in \mathbf{M}$, or $R \in \mathbf{M}$, or $W \notin \mathbf{M}$,
7. $D \perp\!\!\!\perp W|\mathbf{M}$ for $\mathbf{M}$ containing either N or R,
8. $N \perp\!\!\!\perp T|\mathbf{M}$ for $R \in \mathbf{M}$, or $W \notin \mathbf{M}$,
9. $N \perp\!\!\!\perp W|\mathbf{M}$ for $R \in \mathbf{M}$,
10. $R \perp\!\!\!\perp T|\mathbf{M}$ for $W \notin \mathbf{M}$.

From this list, the eights relation covering also the unconditional independence $N \perp\!\!\!\perp T$ is in contradiction with $MI(N;T) = 0.0709$. To set this imperfectness right, we substitute $\kappa_4(N,R)$ by $\kappa_7(N,R,T)$, and consider model $\pi_6 = \kappa_6^{\downarrow B} \triangleright \kappa_6^{\downarrow T} \triangleright \kappa_6 \triangleright \kappa_2 \triangleright \kappa_7 \triangleright \kappa_1$. For this model we have

$$\begin{aligned} IC(\pi_6) &= \sum_{i=6,2,7,1} IC(\kappa_i) - IC(\kappa_6^{\downarrow\{R,T\}}) - IC(\kappa_7^{\downarrow\{R,T\}}) \\ &= 0.5234 + 0.2871 + 0.3236 + 0.4356 - 2 \times 0.0019 = 1.5659. \end{aligned}$$

To accept a model the user should verify that

- the independence relations deduced from the corresponding persegram do not contradict the intuition of the supervising user,
- the independence relations deduced from the corresponding persegram are not in contradiction with the values of (conditional) mutual information values computed from data,
- the marginals from which the resulting model is set up do not differ substantially from the corresponding estimates from data.

To follow these instructions let us transform model π_6 into a form that all the low-dimensional distributions, from which the model is composed, are marginals of the model itself:

$$\begin{aligned} &\nu_1(R) = \kappa_6^{\downarrow R}(R), \\ &\nu_2(T) = \kappa_6^{\downarrow T}(T), \\ &\nu_3(R,T,W) = \nu_1(R) \triangleright \nu_2(T) \triangleright \kappa_6(R,T,W), \\ &\nu_4(B,R) = \nu_1(R) \triangleright \kappa_2(B,R) = \kappa_2(B,R), \\ &\nu_5(N,R,T) = \nu_1(R) \triangleright \nu_2(T) \triangleright \kappa_7(N,R,T), \\ &\nu_6(D,N) = \nu_5^{\downarrow N}(N) \triangleright \kappa_1(D,N). \end{aligned}$$

Thus, $\pi_6 = \nu_1 \triangleright \nu_2 \triangleright \nu_3 \triangleright \nu_4 \triangleright \nu_5 \triangleright \nu_6$, all ν_i (for $i = 1,\ldots,6$) are marginals of π_6. The respective probability distributions generating this model are depicted in Table 2, and the respective persegram is in Figure 4. From this persegram the following list of conditional independence

Table 2: Probability distributions $\nu_1 - \nu_6$.

$\nu_1(R)$			
$\nu_1(1) = .437$	$\nu_1(2) = .563$		
$\nu_2(T)$			
$\nu_2(1) = .566$	$\nu_2(2) = .434$		
$\nu_3(R,T,W)$			
$\nu_3 1,1,1) = .000$	$\nu_3(1,1,2) = .247$	$\nu_3(1,2,1) = .024$	$\nu_3(1,2,2) = .166$
$\nu_3(2,1,1) = .085$	$\nu_3(2,1,2) = .233$	$\nu_3(2,2,1) = .235$	$\nu_3(2,2,2) = .010$
$\nu_4(B,R)$			
$\nu_4(1,1) = .280$	$\nu_4(2,1) = .125$	$\nu_4(3,1) = .032$	
$\nu_4(1,2) = .057$	$\nu_4(2,2) = .230$	$\nu_4(3,2) = .276$	
$\nu_5(N,R,T)$			
$\nu_5(1,1,1) = .010$	$\nu_5(1,1,2) = .036$	$\nu_5(1,2,1) = .136$	$\nu_5(1,2,2) = .197$
$\nu_5(2,1,1) = .238$	$\nu_5(2,1,2) = .153$	$\nu_5(2,2,1) = .182$	$\nu_5(2,2,2) = .048$
$\nu_6(D,N)$			
$\nu_6(1,1) = .45$	$\nu_6(1,2) = .05$	$\nu_6(2,1) = .05$	$\nu_6(2,2) = .45$

relations can be deduced:

$$\begin{array}{ll}
B \perp\!\!\!\perp D|\mathbf{M} & \text{for } \mathbf{M} \text{ containing either } N \text{ or } R,\\
B \perp\!\!\!\perp N|\mathbf{M} & \text{for } R \in \mathbf{M},\\
B \perp\!\!\!\perp T|\mathbf{M} & \text{for } \mathbf{M} = \emptyset, \text{ or } R \in \mathbf{M},\\
B \perp\!\!\!\perp W|\mathbf{M} & \text{for } R \in \mathbf{M},\\
D \perp\!\!\!\perp R|\mathbf{M} & \text{for } N \in \mathbf{M},\\
D \perp\!\!\!\perp T|\mathbf{M} & \text{for } N \in \mathbf{M},\\
D \perp\!\!\!\perp W|\mathbf{M} & \text{for } N \in \mathbf{M}, \text{ or } \{R,T\} \subseteq \mathbf{M},\\
N \perp\!\!\!\perp W|\mathbf{M} & \text{for } \{R,T\} \subseteq \mathbf{M},\\
R \perp\!\!\!\perp T|\mathbf{M} & \text{for } \mathbf{M} = \emptyset, \text{ or } \mathbf{M} = \{B\},
\end{array}$$

neither of which is in contradiction with anything what has been said about the modeled distribution up to now. Distributions ν_1, ν_2 and ν_4 are the original estimates from data. The remaining distributions ν_3, ν_5 and ν_6 are slightly different from the originally estimated distributions. This is due to the modification realized in the process of computation of distributions ν_i. Nevertheless, the deviations from the original data-based estimates are very small, as it can also be seen from the values of Kullback-Leibler divergence

$$\begin{aligned}
Div(\kappa_6 \parallel \nu_3) &= 0.00192,\\
Div(\kappa_7 \parallel \nu_5) &= 0.00192,\\
Div(\kappa_1 \parallel \nu_6) &= 0.00002.
\end{aligned}$$

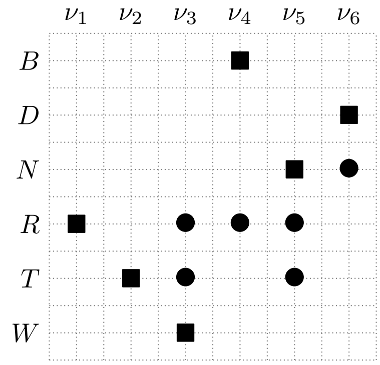


Figure 4: Persegram of model $\pi_6 = \nu_1 \triangleright \nu_2 \triangleright \nu_3 \triangleright \nu_4 \triangleright \nu_5 \triangleright \nu_6$.

(Notice, it is not a pure incidence that $Div(\kappa_6 \| \nu_3) = MI(R,T)$; it can be deduced from other properties the information-theoretic characteristics.) Thus we may say that π_6 is a reasonable model of the distribution generating the data.

Acknowledgement
The authors would like to thank the Czech Academy of Sciences and the Taiwanese Ministry of Science and Technology for their financial support of the bilateral project "**Compositional Models for Data Mining**", in the framework of which the described research was performed.

References

[1] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[2] David Heckerman. Bayesian networks for data mining. *Data mining and knowledge discovery*, 1(1):79–119, 1997.

[3] David J Hand. Principles of data mining. *Drug safety*, 30(7):621–622, 2007.

[4] Milan Studený. *Probabilistic conditional independence structures.* Springer Science & Business Media, 2006.

[5] Radim Jiroušek. Composition of probability measures on finite spaces. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 274–281. Morgan Kaufmann Publishers Inc., 1997.

[6] Radim Jiroušek. Foundations of compositional model theory. *International Journal of General Systems*, 40(6):623–678, 2011.

[7] Francesco M Malvestuto. Equivalence of compositional expressions and independence relations in compositional models. *Kybernetika*, 50(3):322–362, 2014.

[8] Francesco M Malvestuto. Marginalization in models generated by compositional expressions. *Kybernetika*, 51(4):541–570, 2015.

[9] Radim Jiroušek. Persegrams of compositional models revisited: conditional independence. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, Malaga*, pages 915–922, 2008.

[10] Radim Jiroušek and Václav Kratochvíl. Foundations of compositional models: structural properties. *International Journal of General Systems*, 44(1):2–25, 2015.

[11] Albert Perez. Personal communication. 1970–1980.

[12] Solomon Kullback and R A Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:76–86, 1951.

[13] Solomon Kullback. An information-theoretic derivation of certain limit relations for a stationary markov chain. *J. SIAM Control*, 4:454–459, 1966.

Modified DTW Dissimilarity for Analyzing Time-Series Data

Tomoharu NAKASHIMA

Graduate School of Humanities and Sustainable System Sciences
Osaka Prefecture University
Gakuen-cho 1-1, Sakai, Osaka 599-8531
Japan
tomoharu.nakashima@kis.osakafu-u.ac.jp

Keywords: Time-series data, nearest neighbor, regression

Abstract

Time series data analysis is one of the important methods in the real world while it is difficult to achieve. Prediction and classification tasks use a training set that is used to guess the corresponding real value and the category of an unknown time series data pattern. Nearest neighbor approach is used to find the most relevant time series pattern (reference time series pattern) is the most well-known method. The distance measure in find the reference time series pattern include Euclidean distance and dynamic time warping (DTW). The DTW is exeperimentally better than the Euclidean distance in the sense that the shape of the pattern is well considered. The DTW also accommodates time series patterns with different length between each other.

In the DTW, the cost function of any two data points is defined as its distance. The dynamical calculation is performed to obtain the dissimilarity measure between the two time series data up to the calculated point. The original version of the DTW considers all data points equally. However, for some real-world problems there is a possibility that some data time points should be treated more importantly than the other time points. This paper proposes a weighted dynamic time warping method (w-DTW) where a weight vector is introduced in order to represent the importance degree of each data point. In the proposed w-DTW, a weight vector is first prepared that contains a set of real-valued importance. The number of the elements is set to be the one with the longer time series. The cost function is the weighted distance which is the multiplication of the weight and its distance. The overall distance between the two time series data is obtained when the calculation reaches the end point of the shorter time series data.

The proposed w-DTW needs an appropriate weight vector for appropriately calculating the distance between time series. However, it is difficult to obtain the appropriate weight vector by hand. Thus, this paper suggests to use a metaheuristic optimization algorithm to obtain it. Specifically, we propose to apply a firefly algorithm as the metaheuristics. In the firefly algorithm, each firefly holds a set of real values which represents the weight in the w-DTW. The brightness of a firefly is related to the prediction performance of the w-DTW with the associated weight vector. The darker firefly moves towards to other brighter ones by updating the related weight vectors.

In the computational experiments, we apply the firefly algorithm to obtain the optimal weight vector of the w-DTW for a real world problem. As the real world problem, the prediction task is employed where the amount of the rice crop yield of the year is predicted from the precipitation amount per month. The experimental results reveal the important months of the year in the context of the rice crop yield prediction.

Author Index