Proceedings of ISCAMI 2009 Conference

edited by

Radko Mesiar and Daniel Ševčovič

Acta Universitatis Matthiae Belii ser. Mathematics, 16 (2010), 5–16.
Received: 23 June 2009, Accepted: 2 February 2010.

INTERPRETATION OF THE MMPI-2 TEST BASED ON FUZZY SET TECHNIQUES

IVETA BEBČÁKOVÁ, JANA TALAŠOVÁ, AND PAVEL ŠKOBRTAL

Abstract. MMPI-2 (Minnesota Multiphasic Personality Inventory) is a psychological test for detecting pathological features of personality. After answering all the items of the test each patient is assigned a codetype describing his/her mental health. The diagnosed profile of a patient is verified by the comparison of his/her data to the prototypic profile of the given codetype. This paper introduces a mathematical model for codetype determination and codetype verification. The model has two parts. The first solves the problem of codetype determination by using a fuzzy expert system to formally express the linguistic description of the original method. In the second part, each prototypic profile is described by an n-tuple of fuzzy numbers. This allows us to effectively find the degree of agreement between the profiles and data obtained from the patient. The proposed mathematical model is realized in the MATLAB Fuzzy Logic toolbox.

1. Introduction

MMPI-2 (Minnesota Multiphasic Personality Inventory) is one of the most frequently used tests for characterization of personality features and psychic disorders. The first version of the test, MMPI, was developed by psychologist S. R. Hathaway and psychiatrist J. C. McKinley [6] of the Minnesota University. Their goal was to develop an instrument to describe patient's personality more effectively than what was allowed by the psychiatric interview with the patient, [1]. At the same time it was desirable to replace a great number of tests, focusing on single features, by a single test capable of full characterization. The fruit of their labor was an extensive testing method with applications far beyond the clinical practice. Today, a revised version of the test, MMPI-2 [3], is used. MMPI-2 is an important screening method for detecting pathological personality features, which is used in clinical practice, as well as in entrance interviews for universities, military, police, or leading positions [7].

2000 *Mathematics Subject Classification.* 03E72, 91E45, 93A30.
Key words and phrases. MMPI-2, Psychometry, Fuzzy expert systems, Base of rules, Linguistic variable.

Use of the MMPI-2 is very demanding. The examiner needs to possess knowledge of theory and use of psychological tests; he/she should have a Master degree in personal psychology and psychopathology, [7]. Furthermore, correct interpretation of the test requires experience with the MMPI-2 and a special training. For this reason, a software with transparent results providing solid basis for the clinic deliberation would be an enormous asset.

1.1. **Quantitative interpretation of MMPI-2.** An important part of the testing process is quantitative interpretation, [5]. Answers to questionnaire questions are used to saturate a large number of scales. Their rough point values are then transformed into linear T-scores. Based on values of these, a codetype of the patient is determined.

The basis for the MMPI-2 interpretation is a determination of codetype, if possible. Each codetype is defined by T-scores of ten clinical scales (1-Hypochondriasis, 2-Depression, 3-Hysteria, 4-Psychopathic deviate, 5-Masculinity-Feminity, 6-Paranoia, 7-Psychasthenia, 8-Schizophrenia, 9-Hypomania, 0-Social introversion). Value of each T-score comes from the interval $[0, 120]$. Values higher than 65 are considered significantly elevated. According to number and type of increased clinical scales we define 55 different codetypes. Codetypes with one significantly elevated clinical scale are designated "Spike" (ten possible types), while two significantly elevated scales represent a "Two Point" (45 possible types). For a codetype to be well defined, there has to be at least five point difference between the T-scores of the highest scales and remaining T-scores. If this is not satisfied, there is a possibility of triad, for example, and it is not possible to use codetypes.

After finding the codetype, the agreement between patient's data and the respective prototypic profile is checked. In this testing, T-scores of all scales need to be considered. Each of 55 prototypic profiles is defined by specific values of all scales. To have a perfect match between the patient and a given prototypic profile, T-scores of patient's scales must not differ from T-scores of the profile by more than ten points.

For finding the T-scores and determining the codetype, the MMPI-2 software was developed [7]. This software finds the codetype only from the two highest T-scores and rest of the data is not involved in the process. This leads to loss of information and it is wasteful of the full MMPI-2 potential. Furthermore, the software does not strictly adhere to the five-point-difference condition and therefore may return an erroneous result.

In this paper, we present a mathematical model, which can help to find several codetypes best fitting the patient. The codetypes are determined in two steps. In the first step, the model searches for codetypes using the MMPI methodology with slightly modified conditions. In the second step, the additional suitable codetypes are found by comparing the patient's data to the prototypic profiles. Similar approach has already been employed in [2], where the second part of the model

employed a base of rules, which caused several problems. For example, prototypic profiles were not detected with unit overlap and the method was was not universal and tended to prefer certain profiles. In this paper we introduce a different treatment of the second part of the model, which addresses the aforementioned issues.

The Czech version of the MMPI-2 does not work with all of the scales. It uses and saves values of only 79 of them. The mathematical model will consider this simplified version of the MMPI-2.

2. Preliminaries

The codetype determination requiring full satisfaction of all 79 conditions of a prototypic profile is problematic. Classification based on such a crisp mathematical model may not work, because only rarely a patient satisfies fully a prototypic profile. It will be shown that in a situation like this, as well as in many areas of social sciences and psychology, it is effective to use the so called fuzzy approach.

Fuzzy set theory [4, 8] gives us a tool to model the vagueness phenomenon. It allows us to describe mathematically linguistic values and linguistically defined rules. This is the reason why in this special case, where we look for a mathematical model of a linguistically described methodology, description by fuzzy sets is very helpful.

Let U be a nonempty set. A fuzzy set A on U is defined by the mapping $A : U \to [0, 1]$. For each $x \in U$ the value $A(x)$ is called a membership degree of the element x in the fuzzy set A, $A(\cdot)$ is a membership function of the fuzzy set A.

A height of a fuzzy set A on U is a real number $\mathrm{hgt}(A) = \sup_{x \in U}\{A(x)\}$. An intersection of fuzzy sets A, B on U is a fuzzy set $A \bigcap B$ on U with a membership function $(A \bigcap B)(x) = \min\{A(x), B(x)\}$ for any $x \in U$.

A fuzzy number A is a fuzzy set on $\mathbb{R}$ which fulfills the following conditions: the kernel of the fuzzy set A, $\mathrm{Ker}A = \{x \in \mathbb{R} | A(x) = 1\}$, is a non-empty set, the α-cuts of the fuzzy set A, $A_\alpha = \{x \in \mathbb{R} | A(x) \geq \alpha\}$, are closed intervals for all $\alpha \in (0, 1]$, the support of A, $\mathrm{Supp}A = \{x \in \mathbb{R} | A(x) > 0\}$, is bounded.

The family of all fuzzy numbers on $\mathbb{R}$ is denoted by $F_N(\mathbb{R})$. If $\mathrm{Supp}A \subseteq [a, b]$ then A is referred to as a fuzzy number on the interval $[a, b]$. The family of all fuzzy numbers on the interval $[a, b]$ is denoted by $F_N([a, b])$. A linear fuzzy number on the interval $[a, b]$ that is determined by four points $(x_1, 0), (x_2, 1), (x_3, 1), (x_4, 0)$, $a \leq x_1 \leq x_2 \leq x_3 \leq x_4 \leq b$, is a fuzzy number A with the membership function

depending on parameters x_1, x_2, x_3, x_4, as follows

$$\forall x \in [a,b] : A(x, x_1, x_2, x_3, x_4) = \begin{cases} 0, & \text{for } x < x_1; \\ \frac{x-x_1}{x_2-x_1}, & \text{for } x_1 \leq x < x_2; \\ 1, & \text{for } x_2 \leq x \leq x_3; \\ \frac{x_4-x}{x_4-x_3}, & \text{for } x_3 < x \leq x_4; \\ 0, & \text{for } x_4 < x. \end{cases}$$

This linear fuzzy number A will be denoted by $A \sim (x_1, x_2, x_3, x_4)$.

A linguistic variable is a quintuple $(X, T(X), U, G, M)$, where X is a name of the variable, $T(X)$ is a set of its linguistic values (linguistic terms), U is an universe, which the mathematical meanings of the linguistic terms are modelled on, G is a syntactical rule for generating the linguistic terms, and M is a semantic rule, which to every linguistic term $\mathcal{A}$ assigns its meaning $M(\mathcal{A})$ as a fuzzy set on U. If the set of linguistic terms is given explicitly, then the linguistic variable is denoted by $(X, T(X), U, M)$.

Let $(X_j, T(X_j), U_j, M_j)$, $j = 1, 2, \ldots, m$, and $(Y, T(Y), V, N)$ be linguistic variables. Let $\mathcal{A}_{ij} \in T(X_j)$ and $M(\mathcal{A}_{ij}) \in F_N(U_j)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. Let $\mathcal{B}_i \in T(Y)$ and $M(\mathcal{B}_i) \in F_N(V)$, $i = 1, 2, \ldots, n$. Then the following scheme F

$$\begin{array}{c} \text{If } X_1 \text{ is } \mathcal{A}_{11} \text{ and } \ldots \text{ and } X_m \text{ is } \mathcal{A}_{1m}, \text{ then } Y \text{ is } \mathcal{B}_1 \\ \text{If } X_1 \text{ is } \mathcal{A}_{21} \text{ and } \ldots \text{ and } X_m \text{ is } \mathcal{A}_{2m}, \text{ then } Y \text{ is } \mathcal{B}_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \text{If } X_1 \text{ is } \mathcal{A}_{n1} \text{ and } \ldots \text{ and } X_m \text{ is } \mathcal{A}_{nm}, \text{ then } Y \text{ is } \mathcal{B}_n \end{array}$$

is called a linguistically defined function (base of rules).

The process of calculating linguistic values of an output variable for the given linguistic values of input variables by means of such a rule base is called an approximate reasoning. There are several methods of approximate reasoning. The most popular and the most widely used one is the Mamdani algorithm.

Let F be the base of rules defined above and let us assume the observed values to be

$$X_1 \text{ is } \mathcal{A}'_1 \text{ and } X_2 \text{ is } \mathcal{A}'_2 \text{ and } \ldots \text{ and } X_m \text{ is } \mathcal{A}'_m,$$

then by entering the observed values into the base of rules F, according to the Mamdani algorithm, we obtain the output value $Y = \mathcal{B}'$, where the $\mathcal{B}'$ is the linguistic approximation [4] of a fuzzy set B^M. The membership function of the fuzzy set B^M is defined for all $y \in V$ as follows $B^M(y) = \max\{B_1^M(y), \ldots, B_n^M(y)\}$, where $B_i^M(y) = \min\{h_i, B_i(y)\}$, $h_i = \min\{\text{hgt}(A_{i1} \bigcap A'_1), \ \ldots, \ \text{hgt}(A_{im} \bigcap A'_m)\}$, for $i = 1, \ldots, n$.

3. Designed mathematical model

The quantitative interpretation of the MMPI-2 is performed in two steps. First, based on values of clinical scales, a patient's codetype is determined. This is

followed by the verification, where the relevant prototypic profile is compared with the patient's data.

The proposed mathematical model respects this structure of MMPI-2. In the first step, the model finds the three clinical scales with the highest T-scores, and with help of the linguistically described function decides on a codetype. In the second step, the model works with values of all 79 scales and calculates the overlap between the linear T-scores of the patient and the prototypic profile of the codetype found in the previous step. Simultaneously the model searches for other prototypic profiles, which agree well with patient's data.

3.1. **Codetype determination.** Two conditions are important for correct determination of the codetype. First, T-scores of significantly elevated scales must be higher than 65. Second, values of the highest scales must be at least five points higher than values of all remaining scales. In practice, it is often difficult to strictly fulfill this conditions. It has shown to be more effective to use the fuzzy approach and define these conditions linguistically. Furthermore, use of the fuzzy set theory was instrumental in finding more variants of the codetype, which can be presented to the evaluator.

Prior to further processing, the scales need to be ordered from the highest T-score to the lowest. Based on the above mentioned requirements, we then define linguistic variables as:

(1) $\langle$The First Scale Elevation,
$\{$Insignificant, Significant$\}, [0, 120], M_1\rangle$,
(2) $\langle$The Second Scale Elevation,
$\{$Insignificant, Significant$\}, [0, 120], M_1\rangle$,
(3) $\langle$The Third Scale Elevation,
$\{$Insignificant, Significant$\}, [0, 120], M_1\rangle$,
(4) $\langle$The Difference between the First Two Scales,
$\{$Small, Big Enough$\}, [0, 120], M_2\rangle$,
(5) $\langle$The Difference between the 2^{nd} and the 3^{rd} Scale,
$\{$Small, Big Enough$\}, [0, 120], M_2\rangle$,
(6) $\langle$Codetype Shape,
$\{$Spike, Two Point, Potential Triad, Within-Normal-Limits$\}$,
$\{1, 2, 3, 4\}, N\rangle$,

where
M_1(Insignificant) $= IE \sim (0, 0, 63, 65)$, M_1(Significant) $= SE \sim (63, 65, 120, 120)$, M_2(Small) $= SM \sim (0, 0, 0, 5)$, M_2(Big Enough) $= BE \sim (0, 5, 120, 120)$, N(Spike) $= S \sim (1, 1, 1, 1)$, N(Two Point) $= 2P \sim (2, 2, 2, 2)$, N(Potential Triad)$= PT \sim (3, 3, 3, 3)$, N(Within-Normal-Limits) $= WNL \sim (4, 4, 4, 4)$. Some of defined variables are illustrated in Fig. 1 and 2.

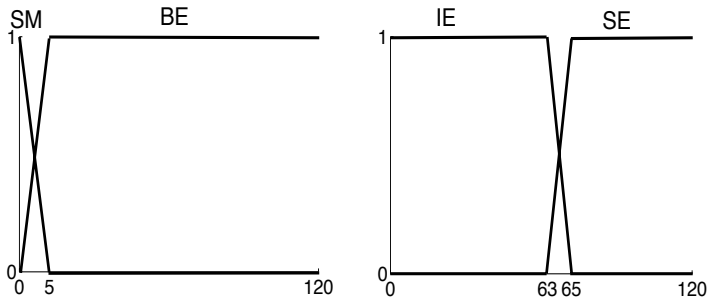

FIGURE 1. Input linguistic variables
Left: *The Difference between the First Two Scales* and its two linguistic values *Small* and *Big Enough* modelled by fuzzy numbers SM and BE.
Right: *The First Scale Elevation* and its two linguistic values *Insignificant* and *Significant* modelled by fuzzy numbers IE and SE.

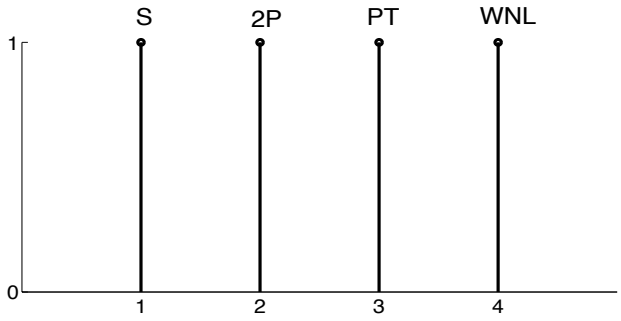

FIGURE 2. Output linguistic variable *Codetype Shape* and its four linguistic values *Spike, Two Point, Potential Triad* and *Within-normal-limits* modelled by fuzzy numbers S,$2P$,PT and WNL.

With help of these six linguistic variables and four rules we construct a base of rules F:

rule 1: If The First Scale Elevation is Significant and The Second Scale Elevation is Insignificant and The Difference between the First Two Scales is Big Enough, then the Codetype Shape is a Spike.

rule 2: If The First Scale Elevation is Significant and The Second Scale Elevation is Significant and The Difference between the 2^{nd} and the 3^{rd} Scale is Big Enough, then the Codetype Shape is Two Point.

rule 3: If The First Scale Elevation is Significant and The Second Scale Elevation is Significant and The Third Scale Elevation is Significant and The Difference between the 2^{nd} and the 3^{rd} Scale is Small, then the Codetype Shape is Potential Triad.

rule 4: If The First Scale Elevation is Insignificant, then the Codetype Shape is Within-Normal-Limits.

The base of rules F has five input linguistic variables - the three highest T-scores of clinical scales and the two differences between them - and one output linguistic variable, which determines the shape of the codetype.

Together with the Mamdani approximate reasoning algorithm, the linguistic function F forms an expert system for determination of the codetype shape. With values of clinical scales as an input, the model produces a fuzzy set B^M that helps the evaluator to determine possible codetype shapes. The membership degree of an element of the set $\{1, 2, 3, 4\}$ in fuzzy set B^M, representing a particular codetype shape, equals to the degree of satisfaction of the respective rule. See, for example, Fig. 3. To determine the complete codetype of the patient, we need to combine the information about the codetype shape with knowledge of the initial ordering of clinical scales. For example, if the codetype shape is Spike and the designation of the highest scale is 8-Schizophrenia, then the codetype is Spike 8.

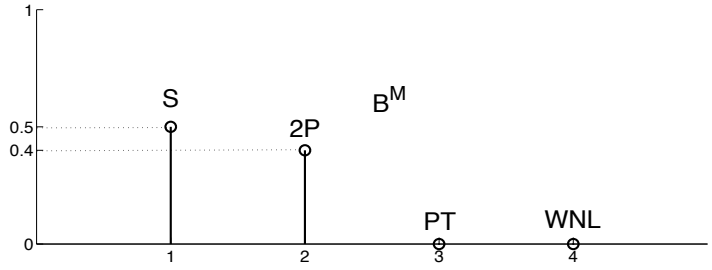

FIGURE 3. The fuzzy set B^M as obtained by entering input values [67 64 62 3 2] into the designed expert system. The degrees of satisfaction express the possibility that the corresponding codetype shape is a Spike (possibility 50%) or a Two Point (possibility 40%).

3.2. **Codetype verification.** Each of all 55 codetypes is described in detail by a so called prototypic profile. Codetype verification is based on the calculation of the degree of agreement between the patient's data and the prototypic profiles corresponding to the codetypes, which were determined in the first part of the model. Besides the verification, the model also searches for other prototypic profiles with a good overlap. Each profile is described by a vector of 79 real numbers representing values of the 79 scales with the T-scores ranging from 0 to 120. For a patient's profile to match a prototypic profile, all the patient's T-scores must be within 10 point distance from the prototypic values.

In the second part of the mathematical model we replaced all crisp numbers t_{ij} describing the prototypic profiles by linear fuzzy numbers $T_{ij} \sim (t_{ij} - 20, t_{ij} - 10, t_{ij} + 10, t_{ij} + 20)$, $i = 1, 2, \ldots, 55$, $j = 1, 2, \ldots, 79$. The example is illustrated in the Fig. 4. The kernel of the designed fuzzy number corresponds to the requirements of the methodic, i.e. if the patient's T-score is within 10 point distance from the prescribed value, there is a perfect match and the membership degree is equal to 1. The support of the fuzzy number was set at twice the length of the kernel, i.e. if the distance of the patient's T-score and the prototypic value is bigger than 20 points, then there is no match at all and the membership degree is zero.

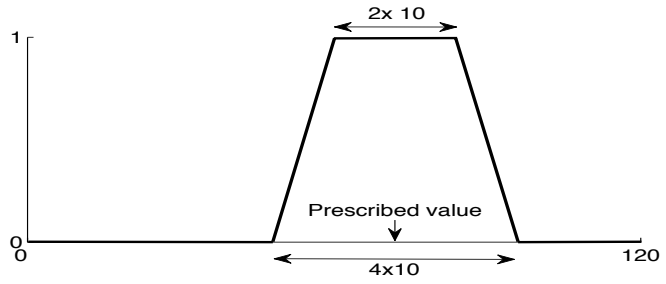

FIGURE 4. Fuzzy number replacing the crisp prototypic value of a scale. The membership degree corresponds to the degree of agreement between the patient's T-score and the prescribed value.

Each i-th, $i = 1, 2, \ldots, 55$, prototypic profile is then described by 79 of these fuzzy numbers. Entering the patient's T-scores $t'_1, t'_2, \ldots, t'_{79}$ into the designed fuzzy numbers, we obtain 79 membership degrees $T_{i1}(t'_1), T_{i2}(t'_2), \ldots, T_{i,79}(t'_{79})$. The degree of agreement between the patient's data and the i-th prototypic profile is

calculated as an arithmetic mean of these membership degrees:

$$h_i = \frac{1}{79} \sum_{j=1}^{79} T_{ij}(t'_j), \ i = 1, 2, \ldots, 55. \tag{1}$$

During the development of the model we tried various aggregation operators. However, the common aggregation operators used for modelling the operation of logical conjunction, such as minimum, proved unfeasible, because a patient rarely satisfies the full range of conditions. On the other hand, the arithmetic mean proved itself to be the most convenient in this case. The degree of agreement between the given data and the prototypic profile here represents the average satisfaction of all 79 prescribed conditions. Compared to minimum, for example, the arithmetic mean provides better information about the satisfaction of given conditions. In the future, the aggregation operator can be readjusted to the requirements of the examiner and the arithmetic mean can be replaced by an other aggregation operator, for example weighted mean or OWA, [9].

Applying the aforementioned approach we are able to test all the 55 prototypic profiles. The result can be modelled by a fuzzy set H on the set U, $U = \{1, 2, \ldots, 55\}$, where each integer between 1 and 55 corresponds to one prototypic profile and the membership degrees $H(i)$, $i = 1, 2, \ldots, 55$, represent the overlap of the respective prototypic profiles with the profile of the patient. The example is illustrated in Fig. 5.

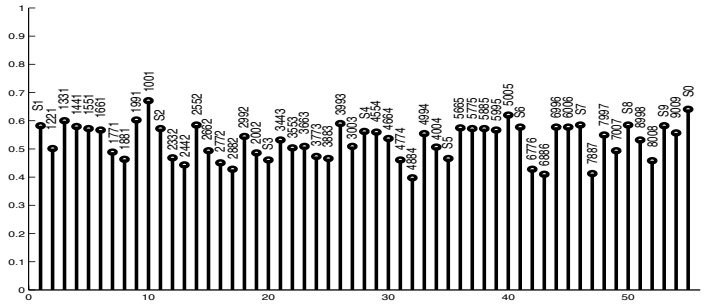

FIGURE 5. The fuzzy set H as obtained by comparing 79 given values with the prescribed prototypic profiles. The degrees of satisfaction represent the overlap between the prototypic profiles and the patient's profile.

4. The implementation of the mathematical model in MATLAB

Both parts of the proposed mathematical model were realized in MATLAB. At first, we have used the Fuzzy Logic Toolbox to create the base of rules and to set the proper approximate reasoning algorithm. Then to each one of the 55 prototypic profiles we have assigned a 79-tuple of fuzzy numbers, as was described in the previous section.

An example of the output can be seen in Fig. 6. The output of the utility is in the form of three figures and linguistic description of the situation. The first figure presents values of clinical scales as obtained from the patient - the patient's profile. The second figure presents possible codetypes, together with their respective degrees of satisfaction. The third figure shows all prototypic profiles and their overlap with the patient's profile. The evaluator can therefore decide, whether the found codetypes are in good agreement with all available patient's data. The linguistic output presents possible codetypes and three prototypic profiles with the best agreement. In addition it comments on a possibility of a triad or scales within normal limits.

In Fig. 6 we demonstrate performance of the implementation. According to clinical scales values, codetype 6-9 was determined. The result is in agreement with the original software. However, during the prototypic profile analysis, the codetype 6-9 didn't show sufficient agreement, as the degree of overlap was only 0.48. The three most faithful profiles were those of codetypes 6-8/8-6, 8-9/9-8, and 7-8/8-7, with 6-8/8-6 showing the best overlap. This suggests that for further deliberation, codetypes 6-8/8-6 should be considered in addition to 6-9.

5. Conclusion

In the paper we have created a mathematical model which can help an examiner with quantitative interpretation of the results given by the MMPI-2 test. For determination of a MMPI codetype we have employed a fuzzy expert system to formally express the linguistically described method of data analysis. To verify the overlap between the prototypic profiles of the found codetypes and the patient's data, all the prototypic profiles were described by 79-tuples of special fuzzy numbers. This allowed us to effectively find the degree of agreement between the respective prototypic profiles and data obtained from the patient. The model contains several free parameters which allow for further fine tuning needed before a practical application

The created fuzzy model was implemented in MATLAB. The created utility, employing the fuzzy approach, can analyze the data while avoiding the shortcomings of the existing software "MMPI-2". In this way the utility can serve as a valuable tool for a human psychiatrist or psychologist in the tuning process.

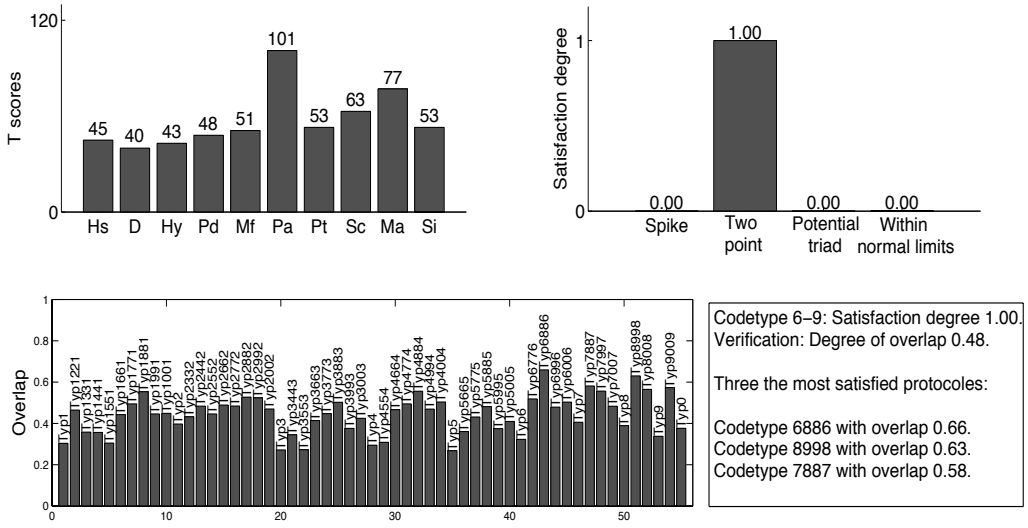

FIGURE 6. Three figures and linguistic description as returned by the MATLAB implementation of the model.

REFERENCES

[1] Archer, R. P., MMPI-A: Assessing adolescent psychopathology, Third edition, Lawrence Erlbaum Associates Publishers, 2005.

[2] Bebčáková, I., Talašová, J., Škobrtal, P., *Fuzzy Approach to Quantitative Interpretation of MMPI-2.* Journal of Applied Mathematics, Vol. 2, Number 1, 2009, 65-75, ISSN 1337-6365

[3] Butcher, J. N., Dahlstrom, W. G., Graham, J. R., Tellegen, A., and Kaemmer, B., MMPI-2: Manual for administration and scoring. University of Minnesota Press, Minneapolis, 1989.

[4] Dubois, D., Prade, H.(Eds.), Fundamentals of fuzzy sets. Kluwer Academic Publishers, Boston/London/Dordrecht, 2000.

[5] Greene, R. L., The MMPI-2 An interpretive manual, Second edition, Allyn & Bacon, 1999.

[6] Hathaway, S. R. and Mckinley, J. C., *A multiphasic personality schedule (Minnesota): I. Construction of the schedule*, Journal of Psychology, 10, 249-254, 1940.

[7] Hathaway, S. R., Mckinley, J. C. and Netík, K., Minnesota Multiphasic Personality Inventory - 2, First czech edition, Testcentrum, Praha, 2002.

[8] Talašová, J., Fuzzy metody vícektriteriálního hodnocení a rozhodování, Univerzita Palackého v Olomouci, Olomouc, 2003.

[9] Torra, V. and Narukawa, Y., Modeling decisions: Information fusion and aggregation operators, Springer-Verlag, Berlin Heidelberg, 2007.

(I. Bebčáková) Department of Mathematical Analysis and Applications of Mathematics, Faculty of Science, Palacký University Olomouc, tř. 17. listopadu 12, 771 46 Olomouc, Czech Republic

E-mail address: bebcakova.iveta@post.cz

(J. Talašová) Department of Mathematical Analysis and Applications of Mathematics, Faculty of Science, Palacký University Olomouc, tř. 17. listopadu 12, 771 46 Olomouc, Czech Republic

E-mail address: talasova@inf.upol.cz

(P. Škobrtal) Department of Psychology and Pathopsychology, Faculty of Education, Palacký University Olomouc, Žižkovo nám. 5, 771 40 Olomouc, Czech Republic

E-mail address: pavel.skobrtal@email.cz

Acta Universitatis Matthiae Belii ser. Mathematics, 16 (2010), 17–23.
Received: 15 June 2009, Accepted: 2 February 2010.

VALUATION OF THE AMERICAN-STYLE OF ASIAN OPTION BY A SOLUTION TO AN INTEGRAL EQUATION

TOMÁŠ BOKES

Abstract. We extend the model for valuation of American-style of Asian options introduced by Hansen, Jørgensen (2000) in [3] by including a nontrivial dividend rate q. We use the theory of conditioned expectations to calculate the formula of the American-style Asian floating strike option with a general average of the underlying asset. We determine an integral equation formula for the value of this type of an option with continuous geometric average and approximate formula for the continuous arithmetic average.

1. Introduction

Evolution of trading systems influences the development of the market of financial derivatives. First, the simple derivatives (as forwards and vanilla options) were used to hedge the risk of a portfolio. Progress in valuation of these simple financial instruments pushed traders into inventing less predictable and more complex derivatives. Using financial derivatives with more complicated pay-offs brings into attention also new mathematical problems.

Asian options belong to a group of path-dependent options, i.e. part of exotic options. Here the pay-off depends on the spot value of the underlying during the whole or some part(s) of the life span of the option. Asian options depend on the (arithmetic or geometric) average of the spot price of the underlying.

Asian options can be used as a tool for hedging the high volatility of the price of assets or goods. The price of an underlying varies during the life span of the option, the holder of the Asian option can be secured for the case when the price jumps to the unpleasant region (too high for call holder or too low for put holder) his loss will be reduced.

Asian options can be divided into two subgroups when considering the type of their pay-off function. The average strike Asian option and the fixed strike Asian

2000 *Mathematics Subject Classification.* 35K15, 35K55, 90A09, 91B28.
Key words and phrases. option pricing, martingales, exotic options, Asian options, American option.

option with the pay-off function for the call option

$$V_T(S, A) = (S - A)^+ \tag{1}$$

and

$$V_T(S, A) = (A - X)^+, \tag{2}$$

respectively.

2. A probabilistic model for pricing of American-Style of Asian options

In this section we provide a formula for the valuation of the early exercise boundary of an American-style Asian option paying nontrivial dividends. We follow the derivation introduced by Hansen, Jørgensen in [3]. Their formula for a floating strike option was derived using the theory of martingales and conditioned expected values. We extend the formula to Asian options on underlying paying non-zero dividend rate.

This model is based on the stochastic behavior of the underlying in time. It is assumed that it is driven by stochastic process satisfying the following stochastic differential equation

$$dS_t = (r - q)S_t\ dt + \sigma S_t\ dW_t^Q \quad \text{on the time interval } [0, T], \tag{3}$$

starting almost surely from the initial price $S_0 > 0$, where the constant parameter $r > 0$ denotes the risk-free interest rate, $q \geq 0$ is a dividend rate, σ is the volatility of stock returns and W_t^Q is a standard Brownian motion with respect to the standard risk-neutral probability measure Q. A solution of equation (3) corresponds to the geometric Brownian motion

$$S_t = S_0 e^{(r-q-\frac{1}{2}\sigma^2)t+\sigma W_t^Q}, \tag{4}$$

for $0 \leq t \leq T$.

The bond (risk-free) market is driven by the differential equation

$$dB_t = rB_t\ dt, \tag{5}$$

with $B_0 = 1$, i.e. $B_t = e^{rt}$.

As we have already mentioned above we shall derive the value of an American-style Asian option with floating strike. If we define the optimal stopping time as T^*, the pay-off of the option is set by

$$V_{T^*} = \Big(\rho(S_{T^*} - A_{T^*})\Big)^+, \tag{6}$$

where V_t is the value of the option at time t, A_t is a continuous average of the stock value during the interval $[0, t]$ and $\rho = 1$ for a call option and $\rho = -1$ for a

put option. We may consider either the continuous arithmetic average

$$A_t = \frac{1}{t}\int_0^t S_u \; du, \tag{7}$$

or the continuous geometric average

$$\ln A_t = \frac{1}{t}\int_0^t \ln S_u \; du \tag{8}$$

or the weighted arithmetic average

$$A_t = \frac{1}{t}\int_0^t a(t-u)S_u \; du, \tag{9}$$

where the kernel function $a(.) \geq 0$ with the property $\int_0^\infty a(\zeta)\; d\zeta < \infty$ is usually defined as $a(s) = e^{-\lambda s}$ for $\lambda > 0$.

3. Valuation

We recall that derivation of the more simple type option was introduced in [3]. According to Hansen and Jørgensen, American-style contingent claims can be priced by the conditioned expectations approach. The option prices are evaluated by considering all possible stopping times in the interval $[t, T]$

$$V(t,S,A) = \operatorname*{ess\,sup}_{s\in\mathcal{T}_{[t,T]}} \mathbb{E}_t^Q\Big[e^{-r(s-t)}\Big(\rho(S_s - A_s)\Big)^+\Big|S_t = S, A_t = A\Big], \tag{10}$$

where $\mathcal{T}_{[t,T]}$ denotes the set of all stopping times in the interval $[t, T]$ and $E_t^Q[X] = E^Q[X|\mathcal{F}_t]$ is the conditioned expectation with information of time t (the information is represented by the filtration $\mathcal{F}_t$ of the σ-algebra $\mathcal{F}$, where the Brownian motion is supported).

To simplify the formula we change the probability measure by the martingale

$$\eta_t = e^{-(r-q)t}\frac{S_t}{S_0} = e^{-\frac{1}{2}\sigma^2 t + \sigma W_t^Q} \tag{11}$$

the new probability measure $\mathcal{Q}$ is defined by

$$d\mathcal{Q} = \eta_T \; dQ. \tag{12}$$

According to Girsanov's theorem, the process

$$W_t^{\mathcal{Q}} = W_t^Q - \sigma t \tag{13}$$

is a standard Brownian motion with respect to the measure $\mathcal{Q}$. The value of the stock under this measure is defined by

$$S_t = S_0 e^{(r-q+\frac{1}{2}\sigma^2)t + \sigma W_t^{\mathcal{Q}}}. \tag{14}$$

All assets priced under this measure are $\mathcal{Q}$-martingales when discounted by the stock price. According to this fact, we can reduce the dimension of stochastic variables. We introduce a variable $\xi_t = \frac{A_t}{S_t}$ and so we can derive

$$\begin{aligned}
V(t,S,A) &= \operatorname*{ess\,sup}_{s\in\mathcal{T}_{[t,T]}} \mathbb{E}_t^{Q}\Big[e^{-r(s-t)}\Big(\rho(S_s-A_s)\Big)^+\Big|S_t=S,A_t=A\Big] \\
&= \operatorname*{ess\,sup}_{s\in\mathcal{T}_{[t,T]}} \mathbb{E}_t^{\mathcal{Q}}\Big[\frac{\eta_t}{\eta_T}e^{-r(s-t)}\Big(\rho(S_s-A_s)\Big)^+\Big|S_t=S,A_t=A\Big] \\
&= \operatorname*{ess\,sup}_{s\in\mathcal{T}_{[t,T]}} \mathbb{E}_t^{\mathcal{Q}}\Big[\frac{e^{r(t-s)}}{e^{(r-q)t}}S_t\Big(\rho(S_s-A_s)\Big)^+\ \mathbb{E}_s^{\mathcal{Q}}\big[\frac{e^{(r-q)T}}{S_T}\big]\Big|S_t=S,A_t=A\Big] \\
&= \operatorname*{ess\,sup}_{s\in\mathcal{T}_{[t,T]}} \mathbb{E}_t^{\mathcal{Q}}\Big[e^{qt-rs}S_t\Big(\rho(S_s-A_s)\Big)^+\ \frac{e^{(r-q)s}}{S_s}\Big|S_t=S,A_t=A\Big] \\
&= \operatorname*{ess\,sup}_{s\in\mathcal{T}_{[t,T]}} \mathbb{E}_t^{\mathcal{Q}}\Big[e^{-q(s-t)}S_t\Big(\rho\Big(1-\frac{A_s}{S_s}\Big)\Big)^+\Big|S_t=S,A_t=A\Big] \\
&= \operatorname*{ess\,sup}_{s\in\mathcal{T}_{[t,T]}} e^{-q(s-t)}S\ \mathbb{E}_t^{\mathcal{Q}}\Big[\Big(\rho(1-\xi_s)\Big)^+\Big|S_t=S,A_t=A\Big].
\end{aligned}$$

The last expression can be rewritten in terms of the new variable $\xi = \frac{A}{S}$ as follows:

$$\widetilde{V}(t,\xi) = e^{-qt}\frac{V(t,S,A)}{S} = e^{-qT_t^*}\ \mathbb{E}_t^{\mathcal{Q}}\Big[\Big(\rho(1-\xi_{T_t^*})\Big)^+\Big], \tag{15}$$

where $T_t^* = \inf\{s\in[t,T]|\xi_s=\xi_s^*\}$ and the function $t\mapsto\xi_t^*$ describes the early exercise boundary.

The stopping region $\mathcal{S}$ and continuation region $\mathcal{C}$ for the call and put options are defined by

$$\mathcal{S}_{call} = \mathcal{C}_{put} = \{0\le t\le T, 0\le\xi<\xi_t^*\}, \tag{16}$$

$$\mathcal{C}_{call} = \mathcal{S}_{put} = \{0\le t\le T, \xi_t^*<\xi<\infty\}. \tag{17}$$

Now we solve the problem (with one stochastic variable) formulated in (15). In what follows, we generalize the result by Hansen, Jørgensen (2000) from [3] for the case of a nontrivial dividend rate $q\ge 0$.

Theorem 3.1. *The value of the floating strike Asian option on stock underlying with dividend rate $q\ge 0$ is given by*

$$\widetilde{V}(t,\xi_t) = \widetilde{v}(t,\xi_t) + \widetilde{e}(t,\xi_t), \tag{18}$$

where

$$\widetilde{v}(t,\xi_t) \equiv \mathbb{E}_t^{\mathcal{Q}}\Big[e^{-qT}\Big(\rho(1-\xi_T)\Big)^+\Big] \tag{19}$$

and

$$\widetilde{e}(t,\xi_t) \equiv \mathbb{E}_t^{\mathcal{Q}}\Big[\int_t^T \rho e^{-qu}\xi_u \mathbf{1}_{\mathcal{S}}(u,\xi_u)\Big(\frac{dA_u}{A_u} - (r - q\xi_u^{-1})du\Big)\Big], \tag{20}$$

with average given by the function A_t and stopping region $\mathcal{S}$. Here the function $1_{\mathcal{S}}(\cdot)$ is the indicator function of the set $\mathcal{S}$, ρ sets the call option by the value 1 and the put option by the value -1.

In the proof of THEOREM 3.1 we will use the following lemma.

Lemma 3.2. *The auxiliary variable $\xi_t = \frac{A_t}{S_t}$ satisfies the following stochastic differential equation:*

$$d\xi_t = \xi_t \frac{dA_t}{A_t} - (r-q)\xi_t\ dt - \sigma\xi_t\ dW_t^{\mathcal{Q}}. \tag{21}$$

Proof of LEMMA 3.2. We express the differential $d\xi_t = d\Big(\frac{A_t}{S_t}\Big)$ as

$$\begin{aligned} d\xi_t &= \frac{1}{S_t}dA_t - \frac{A_t}{S_t^2}dS_t + \frac{A_t}{S_t^3}(dS_t)^2 \\ &= \xi_t\frac{dA_t}{A_t} - (r-q)\xi_t\ dt - \sigma\xi_t\ dW_t^{\mathcal{Q}}, \end{aligned}$$

and the proof of lemma follows. □

Notice that, when comparing to the original expression with a zero dividend rate, $q = 0$, the only difference is that the parameter r is replaced by $r - q$. The value of $\frac{dA_t}{A_t}$ depends on the method of averaging of the underlying used in the valuation. The expression for the arithmetic averaging has form

$$\frac{dA_t^a}{A_t^a} = \frac{1}{t}\Big(\frac{1}{\xi_t^a} - 1\Big)\ dt. \tag{22}$$

As far as, the geometric average is concerned, we have

$$\frac{dA_t^g}{A_t^g} = -\frac{1}{t}\ln\xi_t^g\ dt \tag{23}$$

and for the weighted arithmetic averaging

$$\frac{dA_t^{wa}}{A_t^{wa}} = \frac{1}{t}\Big(\frac{a(0) + \int_0^t a'(t-u)\frac{S_u}{S_t}\ du}{\xi_t^{wa}} - 1\Big)\ dt, \tag{24}$$

where a' is the derivative of the function a. The last equation is unusable in its general form. Neverthless, if we set $a(s) = e^{-\lambda s}$, it becomes

$$\frac{dA_t^{wa}}{A_t^{wa}} = \frac{1}{t}\Big(\frac{1}{\xi_t^{wa}} - (1+\lambda t)\Big)\ dt. \tag{25}$$

Proof of THEOREM 3.1. We follow the proof of the original theorem including necessary modifications related to the form of averaging and the fact that $q \geq 0$.

First, we suppose that (t, ξ) belongs to the continuation region $\mathcal{C}$. The option is held and so we use Itô's lemma to calculate the differential

$$\begin{aligned} d\widetilde{V} &= \frac{\partial \widetilde{V}}{\partial \xi} d\xi + \frac{1}{2}\frac{\partial^2 \widetilde{V}}{\partial \xi^2}(d\xi)^2 + \frac{\partial \widetilde{V}}{\partial t} dt \\ &= \xi \frac{\partial \widetilde{V}}{\partial \xi}\frac{dA}{A} + \Big[-(r-q)\xi \frac{\partial \widetilde{V}}{\partial \xi} + \frac{1}{2}\sigma^2 \xi^2 \frac{\partial^2 \widetilde{V}}{\partial \xi^2} + \frac{\partial \widetilde{V}}{\partial t}\Big] dt - \sigma \xi \frac{\partial \widetilde{V}}{\partial \xi} dW^{\mathcal{Q}} \\ &= -\sigma \xi \frac{\partial \widetilde{V}}{\partial \xi} dW^{\mathcal{Q}}, \end{aligned}$$

where the last equality holds true, because $\widetilde{V}$ is $\mathcal{Q}$-martingale.

Now we suppose that (t, ξ) belongs to the stopping region $\mathcal{S}$. The value of the option is defined by

$$\widetilde{V}(t, \xi_t) = \rho e^{-qt}(1 - \xi_t).$$

So the differential $d\widetilde{V}$ has form

$$\begin{aligned} d\widetilde{V} &= -\rho q e^{-qt}(1-\xi)dt - \rho e^{-qt} d\xi \\ &= -\rho e^{-qt}\xi \frac{dA}{A} + \rho e^{-qt}(r\xi - q)dt + \rho e^{-qt}\sigma \xi dW^{\mathcal{Q}}. \end{aligned}$$

For both regions we have an equation

$$d\widetilde{V}(t, \xi_t) = -\rho e^{-qt} \mathbf{1}_{\mathcal{S}}(t, \xi_t)\Big(\xi_t \frac{dA_t}{A_t} - (r\xi_t - q)dt\Big) + dM_t^{\mathcal{Q}}, \tag{26}$$

where $M_t^{\mathcal{Q}}$ is a $\mathcal{Q}$-martingale. Integrating (26) from t to T and taking expectation we have

$$\begin{aligned} \mathbb{E}_t^{\mathcal{Q}}\Big[\widetilde{V}(T, \xi_T)\Big] - \widetilde{V}(t, \xi_t) &= -\mathbb{E}_t^{\mathcal{Q}}\Big[\int_t^T \rho e^{-qu}\xi_u \mathbf{1}_{\mathcal{S}}(u, \xi_u)\Big(\frac{dA_u}{A_u} - (r - \frac{q}{\xi_u})du\Big)\Big] \\ &\quad + \underbrace{\mathbb{E}_t^{\mathcal{Q}}\Big[\int_t^T dM_u^{\mathcal{Q}}\Big]}_{=0}, \end{aligned}$$

$$\begin{aligned} \widetilde{V}(t, \xi_t) &= \underbrace{\mathbb{E}_t^{\mathcal{Q}}\Big[e^{-qT}\Big(\rho(1-\xi_T)\Big)^+\Big]}_{=\widetilde{v}(t,\xi_t)} \\ &\quad + \underbrace{\mathbb{E}_t^{\mathcal{Q}}\Big[\int_t^T \rho e^{-qu}\xi_u 1_{\mathcal{S}}(\xi_u)\Big(\frac{dA_u}{A_u} - (r - \frac{q}{\xi_u})du\Big)\Big]}_{=\widetilde{e}(t,\xi_t)}. \end{aligned}$$

this completes the proof of THEOREM 3.1. □

Conclusions

In this paper we extended the Hansen and Jørgensen's formula for valuation of the floating strike American-style Asian option by assuming a non-zero dividend rate q. The theory of the martingales and conditioned expected values was used in the calculation of an integral equation for the position of the early exercise boundary. We also present the formula for the weighted arithmetic average with time dependent weights. The presented formula can be used in the comparison of the value of the early exercise boundary to the projected SOR method for Asian option due Kwok, Dai in [1] as well as integral transformation method described in [7].

The numerical experiments and asymptotic analysis of the early exercise boundary will be the subject of the forthcoming paper being prepared.

Acknowledgments

The work has been supported by bilateral Slovak-Bulgarian project APVV SK-BG-0034-08.

References

[1] M. Dai and Y. K. Kwok, *Characterization of optimal stopping regions of American Asian and lookback options*, Mathematical Finance, Vol. 16, No. 1 (January 2006), p. 63-82.

[2] J. N. Dewynne, S. D. Howison, I. Rupf and P. Wilmott, *Some mathematical results in the pricing of American options*, European Journal of Applied Mathematics (1993), Vol. 4, p. 381-398.

[3] A. T. Hansen and P. L. Jørgensen, *Analytical Valuation of American-style Asian Options*, Management Science 46(8) 2000, p. 1116-1136.

[4] J. C. Hull, Options Futures and Other Derivative Securities, 2^{nd} Edition, Prentice Hall, New Jersey, 1993.

[5] Y. K. Kwok, Mathematical Models of Financial Derivatives, 2^{nd} Edition, Springer Berlin Heildelberg, 2008.

[6] D. Revuz and M. Yor, Continuous Martingales and Brownian motion, Springer-Verlag, Berlin, 2005.

[7] D. Ševčovič, *Transformation methods for evaluating approximations to the optimal exercise boundary for linear and nonlinear Black-Scholes equations*, Nonlinear Models in Mathematical Finance New Research Trends in Option Pricing, Matthias Ehrhardt (ed.), Nova Science Publishers, Inc., 2008, p. 153-198.

[8] J. M. Steel, Stochastic Calculus and Financial Applications, Springer-Verlag, New York, 2001.

[9] P. Wilmott, S. Howison and J. Dewynne, The Mathematics of Financial Derivatives, Cambridge University Press, New York, 1995.

Department of Applied Mathematics and Statistics, Faculty of Mathematics, Physics and Informatics, Comenius University, 842 48 Bratislava, Slovak Republic

E-mail address: bokes@fmph.uniba.sk

Acta Universitatis Matthiae Belii ser. Mathematics, 16 (2009), 25–34.
Received: 15 July 2009, Accepted: 2 February 2010.

GENERATED FUZZY IMPLICATIONS AND KNOWN CLASSES OF IMPLICATIONS

VLADISLAV BIBA AND DANA HLINĚNÁ

ABSTRACT. In MV-logic we use a mapping $I : [0,1]^2 \to [0,1]$, called a fuzzy implication, which is a monotonous extension of classical implication on the unit interval. In this paper we deal with one of possible extensions of classical implication. Our implications are generated. Some properties of these implications have already been given in [7]. Well-known classes of implications are (S,N)-implications and R-implications. Some connections between class of our generated implications on one side, and (S,N) and R-implications on the other side will be given. The aim of this paper is to study their properties and to investigate connections between mentioned classes.

1. PRELIMINARIES

We briefly recall definitions and properties of the most important connectives in MV-logic.

Definition 1.1. *A unary operator $n : [0,1] \to [0,1]$ is called a fuzzy negation if, for any $x, y \in [0,1]$,*

- $x < y \Rightarrow n(y) \leq n(x)$,
- $n(0) = 1, n(1) = 0$.

The negation n is called a *strict negation* if and only if the mapping n is continuous and strictly decreasing. A strict negation is strong if it is an involution.

Example 1.2. *The following are some examples of fuzzy negations:*

- $N_s(x) = 1-x$ — *strong negation, standard negation,*
- $n(x) = 1 - x^2$ — *strict, not strong negation,*
- $n(x) = \sqrt{1-x^2}$ — *strong negation,*
- $N_{G_1}(1) = 0, N_{G_1}(x) = 1$ *if* $x < 1$ — *non-continuous, greatest, Gödel negation,*
- $N_{G_2}(0) = 1, N_{G_2}(x) = 0$ *if* $x > 0$ — *non-continuous, smallest, dual Gödel negation.*

Note that the dual negation based on a negation n is given by $n^d(x) = 1-n(1-x)$.

2000 *Mathematics Subject Classification.* 03B52, 03E72, 39B99.
Key words and phrases. Generated implication, (S,N)-implication, R-implication.

Definition 1.3. *A non-decreasing mapping $C : [0,1]^2 \to [0,1]$ is called a conjunctor if, for any $x, y \in [0,1]$, it holds*

- $C(x,y) = 0$ *whenever* $x = 0$ *or* $y = 0$,
- $C(1,1) = 1$.

Commonly used conjunctors in MV-logic are the triangular norms.

Definition 1.4. *A triangular norm (t-norm for short) is a binary operation on the unit interval $[0,1]$, i.e., a function $T : [0,1]^2 \to [0,1]$ such that for all $x, y, z \in [0,1]$, the following four axioms are satisfied:*
(T1) Commutativity $T(x,y) = T(y,x)$,
(T2) Associativity $T(x, T(y,z)) = T(T(x,y), z)$,
(T3) Monotonicity $T(x,y) \leq T(x,z)$ *whenever* $y \leq z$,
(T4) Boundary Condition $T(x,1) = x$.

Remark 1.5. *Note that the dual operator to the conjunctor C, defined by $D(x,y) = 1 - C(1-x, 1-y)$ is called the disjunctor. Equivalently, a disjunctor can be defined as a non-decreasing mapping $D : [0,1]^2 \to [0,1]$ such that $D(x,y) = 1$ whenever $x = 1$ or $y = 1$ and $D(0,0) = 0$. Commonly used disjunctors in MV-logic are the triangular conorms. A triangular conorm (also called a $t-$conorm) is a binary operation S on the unit interval $[0,1]$ which, for all $x, y, z \in [0,1]$, satisfies $(T1) - (T3)$ and $(S4)$ $S(x,0) = x$. The original definition of $t-$conorms given in* [8] *is completly equivalent to the previous axiomatic definition, where the $t-$conorm is based on a given $t-$norm T by formula*

$$S(x,y) = 1 - T(1-x, 1-y).$$

For more information, see [4].

In the literature, we can find several different definitions of fuzzy implications. In this paper we will use the following one, which is equivalent to the definition introduced by Fodor and Roubens in [3]. The readers can obtain more information in [2] and [5].

Definition 1.6. *A function $I : [0,1]^2 \to [0,1]$ is called a fuzzy implication if it satisfies the following conditons:*

(I1) *I is decreasing in its first variable,*
(I2) *I is increasing in its second variable,*
(I3) $I(1,0) = 0$, $I(0,0) = I(1,1) = 1$.

Now, we recall definitions of some important properties of implications, which we will investigate.

Definition 1.7. *A fuzzy implication $I : [0,1]^2 \to [0,1]$ satisfies:*

(NP) *the left neutrality property, or is called left neutral, if*

$$I(1, y) = y;\ \ y \in [0, 1],$$

(EP) *the exchange principle if*

$$I(x, I(y, z)) = I(y, I(x, z)) \text{ for all } x, y, z \in [0, 1],$$

(IP) *the identity principle if*

$$I(x, x) = 1;\ \ x \in [0, 1],$$

(OP) *the ordering property if*

$$x \leq y \iff I(x, y) = 1;\ \ x, y \in [0, 1],$$

(CP) *the contrapositive symmetry with respect to a given negation n if*

$$I(x, y) = I(n(y), n(x));\ \ x, y \in [0, 1].$$

Definition 1.8. *Let $I : [0, 1]^2 \to [0, 1]$ be a fuzzy implication. The function N_I defined by $N_I(x) = I(x, 0)$ for all $x \in [0, 1]$, is called the natural negation of I.*

One of well-known classes of implications is represented by (S, N)-implications, which are based on given t-conorm and negation N.

Definition 1.9. *A function $I : [0, 1]^2 \to [0, 1]$ is called an $(S, N)-$implication if there exist a t-conorm S and fuzzy negation N such that*

$$I(x, y) = S(N(x), y),\ \ x, y \in [0, 1].$$

If N is a strong negation, then I is called a strong implication.

The following characterization of $(S, N)-$implications is from [1].

Theorem 1.10. *(Baczyński and Jayaram [1], Theorem 5.1) For a function $I : [0, 1]^2 \to [0, 1]$, the following statements are equivalent:*

- *I is an (S, N)-implication generated from some t-conorm and some continuous (strict, strong) fuzzy negation N.*
- *I satisfies (I2), (EP) and N_I is a continuous (strict, strong) fuzzy negation.*

Another way of extending the classical binary implication operator to the unit interval $[0, 1]$ uses the *residuation* I with respect to a left-continuous triangular norm T

$$I(x, y) = \max\{z \in [0, 1]; T(x, z) \leq y\}.$$

The following characterization of $R-$implications is from [3].

Theorem 1.11. *(Fodor and Roubens [3], Theorem 1.14) For a function $I : [0, 1]^2 \to [0, 1]$, the following statements are equivalent:*

- *I is an R-implication based on some left-continuous t-norm T.*

- *I satisfies (I2), (OP), (EP), and $I(x,.)$ is a right-continuous for any $x \in [0,1]$.*

Our constructions of implications will make use extensions of the classical inverse of function. One way of extending is described in next definitions.

Definition 1.12. *Let $\varphi : [0,1] \to [0,\infty]$ be a non-decreasing function. The function $\varphi^{(-1)}$ which is defined by*

$$\varphi^{(-1)}(x) = \sup\{z \in [0,1]; \varphi(z) < x\},$$

is called the pseudo-inverse of the function φ, with the convention $\sup \emptyset = 0$.

Definition 1.13. *Let $f : [0,1] \to [0,\infty]$ be a non-increasing function. The function $f^{(-1)}$ which is defined by*

$$f^{(-1)}(x) = \sup\{z \in [0,1]; f(z) > x\},$$

is called the pseudo-inverse of the function f, with the convention $\sup \emptyset = 0$.

One of main contributions of our paper are, in fact, corollaries of the following technical result.

Proposition 1.14. *Let c be a positive real number. Then the pseudo-inverse of a positive multiple of any monotone function $f : [0,1] \to [0,\infty]$ satisfies*

$$(c \cdot f)^{(-1)}(x) = f^{(-1)}\left(\frac{x}{c}\right).$$

Proof. Let f be a non-decreasing function. Then

$$f^{(-1)}(x) = \sup\{z \in [0,1]; f(z) < x\}$$

and

$$(c \cdot f)^{(-1)}(x) = \sup\{z \in [0,1]; c \cdot f(z) < x\} =$$
$$= \sup\left\{z \in [0,1]; f(z) < \frac{x}{c}\right\} = f^{(-1)}\left(\frac{x}{c}\right).$$

Now, the proof for the case of non-increasing function is analogous. □

2. New generated implications

It is well-known that it is possible to generate t-norms from one variable functions. It means it is enough to consider one variable function instead of two variable function. Moreover, we can generate implications in a similar way as t-norms. One of these possibilities is described in the next theorem and example.

Theorem 2.1. *Let $f : [0,1] \to [0,\infty]$ be a strictly decreasing function such that $f(1) = 0$. Then the function $I_f(x,y) : [0,1]^2 \to [0,1]$ which is given by*

$$I_f(x,y) = \begin{cases} 1 & \text{if } x \leq y, \\ f^{(-1)}(f(y^+) - f(x)) & \text{otherwise,} \end{cases}$$

where $f(y^+) = \lim_{y \to y^+} f(y)$ and $f(1^+) = f(1)$ is a fuzzy implication.

Proof. We proceed by the points of the Definition 1.6.

(I1)
- Let $x_1, x_2, y \in [0,1]$ and $x_1 \leq x_2$ and $x_1 \geq y$. Function f is decreasing and therefore $f(x_1) \geq f(x_2)$ and $f(y^+) - f(x_1) \leq f(y^+) - f(x_2)$. Pseudoinverse $f^{(-1)}$ of function f is decreasing, too, and $f^{(-1)}(f(y^+) - f(x_1)) \geq f^{(-1)}(f(y^+) - f(x_2))$. Therefore $I_f(x_1, y) \geq I_f(x_2, y)$ and it means that the function I_f is decreasing in its first variable.
- If $x_1 \leq y \leq x_2$, then $I_f(x_1, y) = 1$ and $I_f(x_2, y) \leq 1$.
- If $x_1 \leq x_2 \leq y$, then $I_f(x_1, y) = I_f(x_2, y) = 1$.

(I2)
- Let $x, y_1, y_2 \in [0,1]$ and $y_1 \leq y_2$ and $x \geq y_2$. Function f is decreasing and therefore $f(y_1^+) \geq f(y_2^+)$ and $f(y_1^+) - f(x) \geq f(y_2^+) - f(x)$. Pseudoinverse $f^{(-1)}$ of function f is decreasing too and $f^{(-1)}\left(f(y_1^+) - f(x)\right) \leq f^{(-1)}\left(f(y_2^+) - f(x)\right)$. Therefore $I_f(x, y_1) \leq I_f(x, y_2)$ and this means that the function I_f is increasing in its second variable.
- If $y_1 \leq x \leq y_2$, then $I_f(x, y_2) = 1$ and $I_f(x, y_1) \leq 1$.
- If $x \leq y_1 \leq y_2$, then $I_f(x, y_1) = I_f(x, y_2) = 1$.

(I3) From the formula for function I_f we get $I_f(0,0) = I_f(1,1) = 1$ and for $I_f(1,0)$ we have

$$I_f(1,0) = f^{(-1)}\left(f(0^+) - f(1)\right) = f^{(-1)}(f(0^+)) = \sup\{z \in [0,1] f(z) > f(0)\} = 0.$$

□

For illustration we introduce some examples of generated implications.

Example 2.2. *Let $f_1, f_2, f_3 : [0,1] \to [0,\infty]$ be functions defined as follows:*

- $f_1(x) = \begin{cases} 1 - x & \text{if } x \leq 0.5, \\ 0.5 - 0.5x & \text{otherwise,} \end{cases}$
- $f_2(x) = \frac{1}{x} - 1$,
- $f_3(x) = -\ln(x)$.

Note, that all three functions are decreasing. For $f_1^{(-1)}, f_2^{(-1)}, f_3^{(-1)}$, we get:

- $f_1^{(-1)}(x) = \begin{cases} 1 - 2x & \text{if } x \leq 0.25, \\ 0.5 & \text{if } 0.25 < x \leq 0.5, \\ 1 - x & \text{otherwise,} \end{cases}$

- $f_2^{(-1)}(x) = \min\left\{\frac{1}{1+x}, 1\right\}$,
- $f_3^{(-1)}(x) = \min\{e^{-x}, 1\}$.

For our functions f_1, f_2, f_3 we get

- $I_{f_1}(x,y) = \begin{cases} 1 & \text{if } x \le y, \\ 1-2x+2y & \text{if } x \le 0.5, y < 0.5, x-y \le 0.25, x > y, \\ 0.5 & \text{if } x \le 0.5, y < 0.5, x-y > 0.25, \\ 0.5 & \text{if } x > 0.5, y < 0.5, x \le 2y, \\ 0.5+y-0.5x & \text{if } x > 0.5, y < 0.5, x > 2y, \\ 1-x+y & \text{if } x > 0.5, y \ge 0.5, \end{cases}$
- $I_{f_2}(x,y) = \begin{cases} 1 & \text{if } x \le y, \\ \frac{1}{\frac{1}{y}-\frac{1}{x}+1} & \text{otherwise}, \end{cases}$
- $I_{f_3}(x,y) = \begin{cases} 1 & \text{if } x \le y, \\ \frac{y}{x} & \text{otherwise.} \end{cases}$

Remark 2.3. *All three implications satisfy IP, NP and OP. Note that I_{f_3} is the well-known Goguen implication.*

3. Properties of I_f implications

In this section we investigate the properties of I_f implications. We turn our attention to relations with (S, N) and R implications and our implications. Directly from Definition 1.7 and the following equivalence for strictly decreasing function f

$$f^{(-1)}(x_0) = 1 \iff x_0 \le \lim_{x \to 1^-} f(x) = f(1^-),$$

we get the condition for NP. The part concerning OP is explained in subsequent example.

Proposition 3.1. *Let $f : [0,1] \to [0,\infty]$ be a strictly decreasing function such that $f(1) = 0$. Then I_f satisfies IP and NP. Moreover, f is continuous in $x = 1$ if and only if I_f satisfies OP.*

The meaning of continuity of the function f in $x = 1$ is introduced in the next example.

Example 3.2. *Let us have function $f : [0,1] \to [0,1]$ given by*

$$f(x) = \begin{cases} 1 - \frac{x}{2} & x \in [0,1[, \\ 0 & x = 1. \end{cases}$$

Pseudoinverse $f^{(-1)} : [0,1] \to [0,1]$ *will be given by*

$$f^{(-1)}(x) = \begin{cases} 1 & x \leq 0.5, \\ 2 - 2x & x \in]0.5, 1]. \end{cases}$$

Implication $I_f : [0,1]^2 \to [0,1]$ *will be given by*

$$I_f(x,y) = \begin{cases} y & x = 1, \\ 1 & \textit{otherwise.} \end{cases}$$

For this implication it holds $I_f(0.5, 0.4) = 1$. *Therefore* I_f *doesn't have OP. It is due to the fact that* $f^{(-1)}(x) = 1$ *for some* $x > 0$, *which is a consequence of violation of continuity of* f *at* $x = 1$. *From continuity in* $x = 1$ *we have* $f^{(-1)}(x) = 1$ *only for* $x = 0$ *and from strictly decreasing function* f *we have* $f(y^+) - f(x) = 0$ *only for* $x = y$, *where* $x, y \in [0,1]$. *It means that continuity in* $x = 1$ *is equivalent with OP for implication* I_f.

Continuity of strictly decreasing function f implies that $f \circ f^{(-1)}(x) = x$. Therefore we get for EP and CP the propositions.

Proposition 3.3. *Let* $f : [0,1] \to [0,\infty]$ *be a continuous strictly decreasing function such that* $f(1) = 0$. *Then the implication* I_f *satisfies EP.*

Proof. Since f is continuous function on $[0,1]$ and by definition $f(1^+) = f(1)$, we have $f(z^+) = f(z)$ $\forall z \in [0,1]$. Also $\forall z \in [0,1] : f^{(-1)} \circ f(z) = z$ and $\forall z \in [0, f(0)] : f \circ f^{(-1)}(z) = z$.

- Let $x > z$ and $y > z$. Then

  $$f(I_f(y,z)) = f(f^{(-1)}(f(z) - f(y))) = f(z) - f(y),$$

  and

  $$I_f(x, I_f(y,z)) = \begin{cases} 1 & x \leq I_f(y,z), \\ f^{(-1)}(f(z) - f(y) - f(x)) & \text{otherwise.} \end{cases}$$

  Analogously

  $$I_f(y, I_f(x,z)) = \begin{cases} 1 & y \leq I_f(x,z), \\ f^{(-1)}(f(z) - f(y) - f(x)) & \text{otherwise.} \end{cases}$$

  If $x \leq I_f(y,z)$, then $f(x) \geq f(f^{(-1)}(f(z) - f(y)))$ and then $f(x) \geq f(z) - f(y)$ or equivalently $f(y) \geq f(z) - f(x)$. The last inequality implies $y \leq I_f(x,z)$ and it means $I_f(x, I_f(y,z)) = I_f(y, I_f(x,z))$
- Let $x > z$ and $y \leq z$. It is clear that $I_f(x, I_f(y,z)) = I_f(x,1) = 1$. Since f is decreasing function f, we have $f(y) \geq f(z) - f(x)$ and this implies $y \leq f^{(-1)}(f(z) - f(x))$. From previous inequality and the formula for I_f we get $I_f(y, I_f(x,z)) = I_f(y, f^{(-1)}(f(z) - f(x)) = 1$.

- The proof is very similar to the previous point for $x \leq z$ and $y > z$.
- Let $x \leq z$ and $y \leq z$. Then obviously $I_f(x, I_f(y, z)) = I_f(y, I_f(x, z)) = 1$.

□

Remark 3.4. *We study the properties of implications I_f under which they are $(S, N)-$ or $R-$ implications. Because there are relations between $(S, N)-$ implications and EP and continuity of N_{I_f}, the previous proposition leads us to dealing with continuous function f. Continuity of function f implies continuity of natural negation based on I_f. Moreover for continuous and bounded strictly decreasing function f such that $f(1) = 0$ and $f(0) = c$ the natural negation N_{I_f} is strong.*

Proposition 3.5. *Let $f : [0, 1] \to [0, c]$ be a continuous bounded decreasing function such that $f(1) = 0$. The I_f possess CP only with respect to its natural negation $N_{I_f}(x) = f^{-1}(f(0) - f(x))$.*

Proof. Let $f : [0, 1] \to [0, c]$ be a continuous bounded decreasing function, such that $f(1) = 0$ and $N_{I_f}(x) = f^{-1}(f(0)-f(x))$. Since we deal with classical inverse function, we have

$$\forall x \in [0, 1]; f(N_{I_f}(x)) = f(0) - f(x),$$

and therefore

$$\forall x, y \in [0, 1]^2; f(N_{I_f}(x)) - f(N_{I_f}(y)) = f(y) - f(x).$$

Since f is continuous, we get $f(y^+) = f(y)$. Since N_{I_f} is strictly decreasing, the conditions $x < y$ and $N_{I_f}(x) > N_{I_f}(y)$ are equivalent and

$$I_f(x, y) = \begin{cases} f^{-1}(f(y) - f(x)) & x > y, \\ 1 & \text{othervise.} \end{cases}$$

Therefore I_f possess CP. Let I_f possess CP w.r. to $n(x)$. We have:

$$I_f(x, 0) = \begin{cases} 1 & x = 0, \\ f^{-1}(f(0) - f(x)) & \text{otherwise,} \end{cases}$$

and

$$I_f(1, n(x)) = \begin{cases} 1 & n(x) = 1, \\ f^{-1}(f(n(x))) & \text{otherwise.} \end{cases}$$

Since $I_f(1, n(x)) = I_f(x, 0)$, we have that $n(x) = f^{-1}(f(n(x))) = f^{-1}(f(0) - f(x)) = N_{I_f}(x)$ for all $x > 0$ and $n(0) = 1$. □

Remark 3.6. *This proposition is a corollary of Proposition 2.5.28 of* [2]*, since continuity of function f implies that we have $R-$implication (Theorem 1.16 from* [3]*).*

It is well known that generators of continuous Archimedean t-norms are unique up to a positive multiplicative constant, and this is also true for the f generators of I_f implications. The next theorem is a corollary of Proposition 1.14.

Theorem 3.7. *Let c be a positive constant and $f : [0,1] \to [0,\infty]$ be a strictly decreasing function. Then the implications I_f and $I_{c\cdot f}$ which are based on functions f and $c \cdot f$ are identical.*

Proof.
- Let $x, y \in [0,1], x \leq y$ and c be a positive real number. From Theorem 2.1 we get $I_{c\cdot f}(x,y) = I_f(x,y) = 1$.
- Let $x, y \in [0,1], x > y$ and c be a positive real number. Then from Theorem 2.1 and Proposition 1.14 we get

$$I_{c\cdot f}(x,y) = (c\cdot f)^{(-1)}\left((c\cdot f)(y^+) - (c\cdot f)(x)\right) =$$

$$= f^{(-1)}\left(\frac{(c\cdot f)(y^+) - (c\cdot f)(x)}{c}\right) = f^{(-1)}\left(f(y^+) - f(x)\right) = I_f(x,y).$$

□

The last theorem of this sections is corollary of previous propositions and Theorems 1.10 and 1.11.

Theorem 3.8. *Let $f : [0,1] \to [0,\infty]$ be a continuous strictly decreasing function such that $f(1) = 0$. Then I_f is an R-implication given by some left-continuous t-norm, and more if $f(0) < \infty$ then I_f is an (S,N)-implication, too.*

The full characterization of $f-$ generated fuzzy implications is yet unknown, and is significant enough to merit attention. Our future endeavors will be along these lines. Note that similar problems relating $QL-$implications and $R-$ and $(S,N)-$ implications were recently studied in [6].

Supported by Project 1ET100300517 of the Program "Information Society" and by Project MSM0021630529 of the Ministry of Education.

References

[1] Baczyński, M. and Balasubramaniam, J., *On the characterizations of (S,N)-implications*, Fuzzy Sets and Systems 158 (2007), 1713–1727.
[2] Baczyński, M. and Balasubramaniam, J., Fuzzy implications, Studies in Fuzziness and Soft Computing, Vol. 231, Springer, Berlin (2008)
[3] Fodor, J. C. and Roubens, M., Fuzzy Preference Modeling and Multicriteria Decision Support, Kluwer Academic Publishers, Dordrecht 1994.
[4] Klement, E. P., Mesiar, R. and Pap, E., Triangular Norms, Kluwer, Dordrecht 2000.
[5] Mas, M., Monserrat, M., Torrens and J., Trillas, E., *A survey on fuzzy implication functions*, IEEE Transactions on Fuzzy Systems 15 (6) 1107–1121, (2007).

[6] Shi, Y., Van Gasse, B., Ruan, D. and Kerre, E. E., *On the first place antitonicity in QL-implications*, Fuzzy Sets and Systems 159(22): 2988-3013 (2008)
[7] Smutná, D., *On many valued conjunctions and implications*, Journal of Electrical Engineering, 10/s, vol. 50, 1999, 8–10
[8] Schweizer, B. and Sklar, A., Probabilistic Metric Spaces, North Holland, New York, 1983

Dept. of Mathematics, FEEC, BUT Brno, Technická 8, 616 00, Czech rep.
E-mail address, V. Biba: xbibav00@stud.feec.vutbr.cz
E-mail address, D. Hliněná: hlinena@feec.vutbr.cz

Acta Universitatis Matthiae Belii ser. Mathematics, 16 (2010), 35–51.
Received: 26 June 2009, Accepted: 2 February 2010.

FUZZME: A NEW SOFTWARE FOR MULTIPLE-CRITERIA FUZZY EVALUATION

PAVEL HOLEČEK AND JANA TALAŠOVÁ

Abstract. This paper is focused on an introduction of a new software product, which is called FuzzME. This software was developed as a tool for creating fuzzy models of multiple-criteria evaluation and decision making. The type of evaluations employed in the fuzzy models fully corresponds with the paradigm of the fuzzy set theory; the evaluations express the (fuzzy) degrees of fulfillment of corresponding goals. The FuzzME software works with both quantitative and qualitative criteria. The basic structure of evaluation is described by a goals tree. Within the goals tree, aggregation of partial fuzzy evaluations is done either by one of fuzzified aggregation operators or by a fuzzy expert system. The FuzzME software takes advantage of linguistic fuzzy modeling to the maximum extent.
This paper also contains a short summary of other available software product for fuzzy multiple-criteria evaluation.
In this paper, the possibilities of FuzzME are demonstrated on a sample problem - evaluation of a new employee.

1. Introduction

There are many situations which require use of multiple-criteria evaluation models. Such models can be utilized e.q. for evaluation of universities, rating of clients of a bank or for evaluation of new employees. In the chapter 4, the last situation will be used as an example and its solution with FuzzME software will be described more in detail.

In the evaluation models, some of the input data are set expertly (e.g. evaluations of alternatives according to qualitative criteria, partial evaluating functions for quantitative criteria, a choice of a suitable type of aggregation, criteria weights, or eventually, a rule base describing the relation between criteria values and the overall evaluation). Because uncertainty is the typical feature of any expert information, the fuzzy set theory is a suitable mathematical tool for creating such models. For the practical use of the fuzzy models of multiple-criteria

2000 *Mathematics Subject Classification.* Primary 90B50, 91B06; Secondary 03E72.
Key words and phrases. Fuzzy expert system, Fuzzy OWA operator, Fuzzy weighted average, Multiple-criteria fuzzy evaluation, Normalized fuzzy weights, Software.

evaluation, their user-friendly software implementation is necessary. But a good theoretical basis of the used models is crucial, too. The clear and well-elaborated theory of multiple-criteria fuzzy evaluation makes it possible to create an understandable methodics for the software user. And a good methodics is essential for correct application of any software to solving real problems.

There is a large number of papers and books dealing with the theory and methods of multiple-criteria evaluation that make use of the fuzzy approach (e.g. [1], [2], [3], [4]).

The most commonly used software for multiple criteria evaluation and decision making based on fuzzy models is FuzzyTECH [5] even if it was not its main purpose (its main application area is fuzzy control). FuzzyTECH is a general software product that makes it possible to create and use fuzzy expert systems. It also includes neural networks algorithms for deriving fuzzy rule bases from data. Interesting applications of this software to evaluation and decision making in the area of business and finance were published in [6].

In 2000 a Czech software company, TESCO SW Inc., developed a software product whose name is NEFRIT. It uses fuzzy methods for multiple criteria evaluation and decision making. The fuzzy model of evaluation applied there is described in detail in [7] and in the book [8]. The demo version of this software is enclosed to the book [8]. NEFRIT can work with expert fuzzy evaluations of alternatives according to qualitative criteria. The values of quantitative criteria can be either crisp or fuzzy. Evaluating functions for quantitative criteria represent membership functions of partial fuzzy goals. For aggregation, the method of weighted average of partial fuzzy evaluations is used. The weights (crisp, not fuzzy) express shares of particular partial evaluations in the aggregated evaluation. Fuzzy evaluations on all levels of the goals tree express fuzzy degrees of fulfillment of the corresponding goals. Publicly available version of NEFRIT does not make it possible to use a fuzzy expert system for evaluation. This software was originally developed for the Czech National Bank (decision making about granting a credit). Further, it was used e.g. by the Czech Tennis Association, the Czech Basketball Association and in other institutions. Nowadays it is tested by the Supreme Audit Office of the Czech Republic. The successor of NEFRIT, in terms of the used theoretical basis, is the FuzzME software.

The FuzzME software (**Fuzz**y models of **M**ultiple-criteria **E**valuation), presented in this paper, is based on the theoretical concept of evaluation which is very close to the original Zadeh's ideas. Similarly to his paper [1], the evaluations of alternatives according to particular criteria represent their degrees of fulfillment of the corresponding partial goals. Besides evaluations expressed by real numbers in $[0, 1]$, fuzzy evaluations modeled by fuzzy numbers on the same interval are employed in the software. They represent, analogously, the fuzzy

degrees of fulfillment of the partial goals which are connected to the criteria. Resulting fuzzy evaluations, which are obtained by aggregation, have a similar clear interpretation. This theoretical approach to (fuzzy) evaluation was published in the book [8] and in the paper [7] and is used also in NEFRIT.

In contrast with NEFRIT, the aggregation is not limited only to simple weighted average method. The FuzzME software also enables to use the fuzzy OWA operator for the aggregation or to define evaluating function by a fuzzy rule base.

For the aggregation of the partial evaluations by the method of weighted average, fuzzy weights can be used (in contrast to NEFRIT which works only with crisp weights). The theory of normalized fuzzy weights, ways of their setting (including a method for removing potential inconsistence) and algorithm for calculation of the fuzzy weighted average are taken from [9].

Another fuzzy aggregation operator, available in the FuzzME software, is a fuzzified OWA operator. Again, it works with normalized fuzzy weights. The fuzzy OWA operator and the used algorithm for its calculation are described in [10].

In the FuzzME software, multiple-criteria evaluating functions can also be defined by means of fuzzy rule bases. Three algorithms are offered for the approximate reasoning - the standard Mamdani algorithm and two modified Sugeno algorithms (Sugeno-WA and Sugeno-WOWA). The advantage of this software is that all of these types of aggregation can be arbitrarily combined in the same goals tree.

There are also software products for multiple-criteria decision making based on other mathematical methods but they are usually designed for solving a particular assignment. Fuzzy toolboxes of general mathematical systems such as Matlab can be used for multiple-criteria decision making, too. But our investigation by means of Internet did not result software fully comparable to FuzzME. Its universality and comprehensiveness make it unique.

2. Preliminaries

A fuzzy set A on a universal set X is characterized by its membership function $A : X \to [0, 1]$. $Ker\ A$ denotes a kernel of A, $Ker\ A = \{x \in X \mid A(x) = 1\}$. For any $\alpha \in [0, 1]$, A_α denotes an α-cut of A, $A_\alpha = \{x \in X \mid A(x) \geq \alpha\}$. A support of A is defined as $Supp\ A = \{x \in X \mid A(x) > 0\}$. The symbol $hgt\ A$ denotes a height of the fuzzy set A, $hgt\ A = \sup\{A(x) \mid x \in X\}$. An intersection and a union of the fuzzy sets A and B on X are defined for all $x \in X$ by the following formulas: $(A \cap B)(x) = \min\{A(x), B(x)\}$, $(A \cup B)(x) = \max\{A(x), B(x)\}$.

A fuzzy number is a fuzzy set C on the set of all real numbers $\Re$ which satisfies the following conditions: a) the kernel of C, $Ker\ C$, is not empty, b) the α-cuts of C, C_α, are closed intervals for all $\alpha \in (0, 1]$, c) the support of C, $Supp\ C$, is bounded. A fuzzy number C is called to be defined on $[a, b]$, if $Supp\ C \subseteq [a, b]$.

Real numbers $c^1 \leq c^2 \leq c^3 \leq c^4$ are called significant values of the fuzzy number C if the following holds: $[c^1, c^4] = Cl(Supp\ C), [c^2, c^3] = Ker\ C$, where $Cl(Supp\ C)$ denotes a closure of $Supp\ C$.

Any fuzzy number C can be characterized by a pair of functions $\underline{c} : [0,1] \to \Re$, $\overline{c} : [0,1] \to \Re$ which are defined by the following formulas: $C_\alpha = [\underline{c}(\alpha), \overline{c}(\alpha)]$ for all $\alpha \in (0,1]$, and $Cl(Supp\ C) = [\underline{c}(0), \overline{c}(0)]$. The fuzzy number C is called to be linear if both the functions $\underline{c}$, $\overline{c}$ are linear. A linear fuzzy number is fully determined by its significant values because $\underline{c}(\alpha) = (c_2 - c_1) \cdot \alpha + c_1$, $\overline{c}(\alpha) = (c_3 - c_4) \cdot \alpha + c_4$. For that reason, we can denote it as $C = (c^1, c^2, c^3, c^4)$.

An ordering of fuzzy numbers is defined as follows: a fuzzy number C is greater than or equal to a fuzzy number D, if $C_\alpha \geq D_\alpha$ for all $\alpha \in (0,1]$.

A fuzzy scale makes it possible to represent a closed interval of real numbers by a finite set of fuzzy numbers. Let $T_1, T_2, ..., T_s$ be fuzzy numbers defined on $[a,b]$, forming a fuzzy partition on the interval, i.e., for all $x \in [a,b]$ the following holds

$$\sum_{i=1}^{s} T_i(x) = 1, \tag{1}$$

then the set of the fuzzy numbers can be linearly ordered (see [8]). If the fuzzy numbers $T_1, T_2, ..., T_s$ are defined on $[a,b]$, form a fuzzy partition on the interval and are numbered according to their linear ordering, then they are said to form a fuzzy scale on $[a,b]$.

An uncertain division of the whole into m parts can be modeled by normalized fuzzy weights. Fuzzy numbers $V_1, ..., V_m$ defined on $[0,1]$ are normalized fuzzy weights if for any $i \in \{1, ..., m\}$ and any $\alpha \in (0,1]$ it holds that for any $v_i \in V_{i\alpha}$ there exist $v_j \in V_{j\alpha},\ j = 1, ..., m,\ j \neq i$, such that

$$v_i + \sum_{j=1, j\neq i}^{m} v_j = 1. \tag{2}$$

3. The FuzzME software

The mathematical models of the FuzzME software are based primarily on the theory and methods of multiple-criteria evaluation that were published in [8] and [7]. The theory of normalized fuzzy weights, the definition of fuzzy weighted average, and the algorithm for its computation were taken from [9], [11] and [12]. The fuzzified OWA operator and the algorithm for its calculation published in [10] are also used in the software.

In the FuzzME software, the basic structure of the fuzzy model of multiple-criteria evaluation is expressed by a goals tree. The root of the tree represents the overall goal of evaluation and each branch corresponds to a partial goal. The

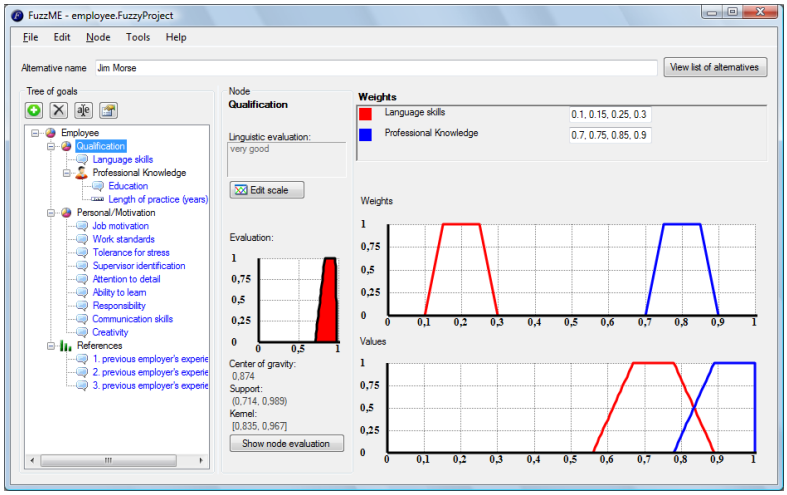

FIGURE 1. The main window of the software

goals at the ends of branches are connected either with quantitative or qualitative criteria.

When an alternative is evaluated, evaluations with respect to criteria connected with the terminal branches are calculated first. Independently of the criterion type, each of the evaluations is described by a fuzzy number defined on the interval $[0, 1]$. It expresses the fuzzy degree of fulfillment of the corresponding partial goal.

These partial fuzzy evaluations are then aggregated according to the defined type of the tree node. Three types of aggregation are available: a fuzzy weighted average (fuzzy WA), an ordered fuzzy weighted average (fuzzy OWA) or aggregation by means of a fuzzy expert system. For aggregation by fuzzy weighted average or ordered fuzzy weighted average, normalized fuzzy weights must be set. The weights express uncertain shares of the partial evaluations in the aggregated one. For the fuzzy expert system, the fuzzy rule base must be defined and an inference algorithm must be chosen (the Mamdani algorithm, the Sugeno-WA or the Sugeno-WOWA algorithm of approximate reasoning).

The overall evaluation reflects the degree of fulfillment of the main goal. A verbal description of the overall evaluation can be obtained by means of the implemented linguistic approximation algorithm.

The overall evaluations can be compared within the frame of a given set of alternatives. By this comparison the best of the alternatives can be chosen. That is why the FuzzME software can be also used as a decision support system.

The import and export of data is supported by the software, too. The FuzzME software is available in the Czech and English versions.

3.1. **Goals tree.** Goals trees represent the basic structure of fuzzy models of multiple-criteria evaluation in the FuzzME software. When a goals tree is designed, the main goal is consecutively divided into goals of progressively lower levels. The process of division is stopped when such goals are reached whose fulfillment can be assessed by means of some known characteristics of alternatives (i.e. quantitative or qualitative criteria).

The design of a tree structure in the goals-tree editor is the first step in forming a fuzzy evaluation model in FuzzME. In the next step, the type of each node in the tree must be specified. For the nodes at the ends of tree branches the user defines if the node is connected with a quantitative or qualitative criterion. For the other nodes he/she sets the type of aggregation - fuzzy weighted average, ordered fuzzy weighted average or fuzzy expert system.

3.2. **Criteria of evaluation.** In the models of evaluation created by the FuzzME software, qualitative and quantitative criteria can be combined arbitrarily.

3.2.1. *Qualitative criteria.* According to qualitative criteria, alternatives are evaluated verbally, by means of values of linguistic variables of special kinds - linguistic scales, extended linguistic scales and linguistic scales with intermediate values.

A linguistic variable is defined as a quintuple $(\mathcal{V}, \mathcal{T}(\mathcal{V}), X, G, M)$, where $\mathcal{V}$ is a name of the variable, $\mathcal{T}(\mathcal{V})$ is a set of its linguistic values, X is a universal set on which the meanings of the linguistic values are defined, G is a syntactic rule for generating values in $\mathcal{T}(\mathcal{V})$, and M is a semantic rule which maps each linguistic value $\mathcal{C} \in \mathcal{T}(\mathcal{V})$ to its mathematical meaning, $C = M(\mathcal{C})$, which is a fuzzy set on X.

A linguistic scale on $[a, b]$ is a special case of the linguistic variable $(\mathcal{V}, \mathcal{T}(\mathcal{V}), X, G, M)$, where $X = [a, b]$, $\mathcal{T}(\mathcal{V}) = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_s\}$ and the meanings of the linguistic values $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_s$ are modeled by fuzzy numbers $T_1, T_2, \ldots, T_s$ which form a fuzzy scale on $[a, b]$. As the set of linguistic values of the scale is defined explicitly, it is not necessary to include the grammar G into the scale notation.

In the FuzzME software, the user defines a linguistic scale for each qualitative criterion in the fuzzy-scale editor. For example, the linguistic scale *communication skills of an employee* can contain linguistic values *inadequate*, *adequate*, *satisfying*, *good* and *very good*. The evaluating linguistic scale is usually defined on $[0, 1]$; in other cases, it has to be transformed to this interval.

The extended linguistic scale contains, besides elementary terms of the original scale, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_s$, also derived terms in the form $\mathcal{T}_i$ *to* $\mathcal{T}_j$, where $i < j, i, j \in \{1, 2, \ldots, s\}$. For example, the user can evaluate *communication skills of an employee* by the linguistic term *satisfying to very good*. The meaning of the linguistic value $\mathcal{T}_i$ *to* $\mathcal{T}_j$ is modeled by $T_i \cup_L T_{i+1} \cup_L \cdots \cup_L T_j$, where $\cup_L$ denotes the

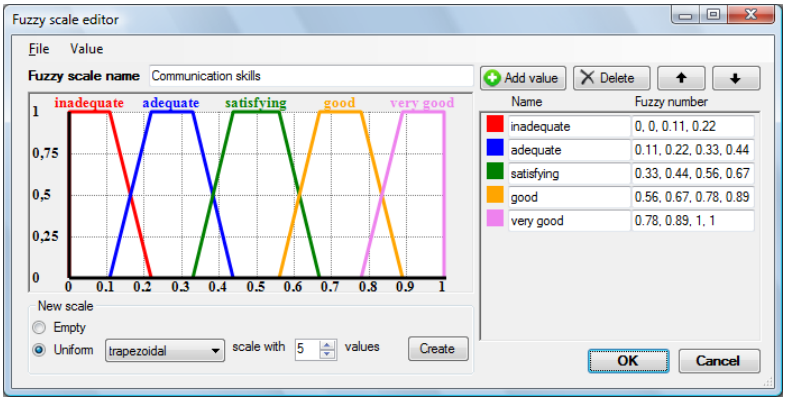

FIGURE 2. Linguistic scale editor

union of fuzzy sets based on the Lukasiewicz disjunction; e.g. $(T_i \cup_L T_{i+1})(x) = \min\{1, T_i(x) + T_{i+1}(x)\}$ for all $x \in \Re$.

The linguistic scale with intermediate values is the original linguistic scale enriched with derived terms *between* $\mathcal{T}_i$ *and* $\mathcal{T}_{i+1}$, $i \in \{1, 2, \ldots, s-1\}$. The meaning of the derived term *between* $\mathcal{T}_i$ *and* $\mathcal{T}_{i+1}$ is modeled by the arithmetic average of the fuzzy numbers T_i and T_{i+1}.

In the FuzzME software, the user evaluates a given alternative according to a qualitative criterion by selecting a proper linguistic evaluation from a drop-down list box. He/she can choose the value from a standard linguistic scale, extended scale or scale with intermediate values.

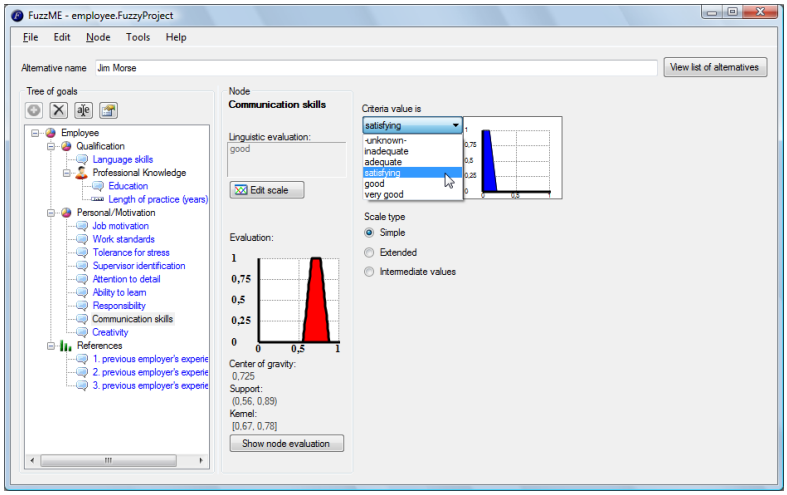

FIGURE 3. Choosing the value of a qualitative criterion

The three mentioned structures of linguistic values are also applied when resulting fuzzy evaluations are approximated linguistically.

3.2.2. *Quantitative criteria.* The evaluation of an alternative with respect to a quantitative criterion is calculated from the measured value of the criterion by means of the evaluating function expertly defined for the criterion. The evaluating function is the membership function of the corresponding partial goal. The FuzzME software admits both crisp and fuzzy values of quantitative criteria. The fuzzy values represent inaccurate measurements or expert estimations of the criteria values. In the case of a fuzzy value, the corresponding partial fuzzy evaluation is calculated by the extension principle.

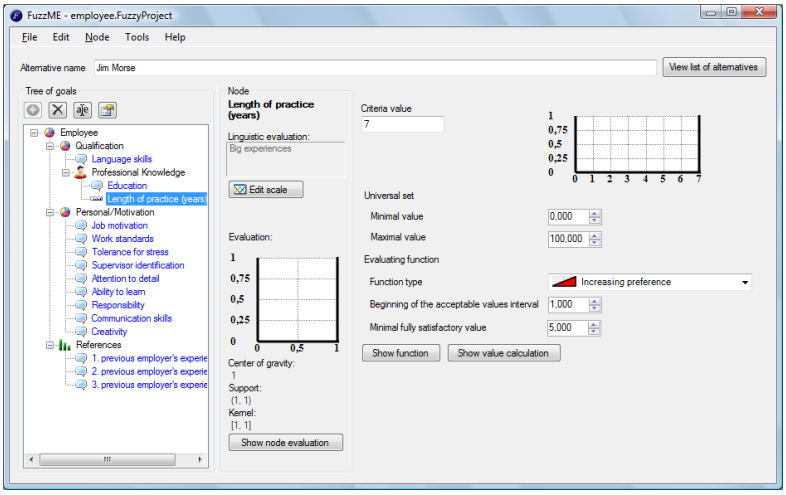

FIGURE 4. A quantitative criterion

In the FuzzME software, the evaluating function of a quantitative criterion is formally set by means of a fuzzy number. For example, if the evaluating function is defined by a linear fuzzy number $F = (f_1, f_2, f_3, f_4)$, then f_1 is the lower limit of all at least partly acceptable values of the criterion, f_2 is the lower limit of its fully satisfactory values, f_3 is the upper limit of the fully satisfactory values, and f_4 is the upper limit of the acceptable values.

For example, when a company wants to hire a new employee, the candidates are evaluated according to the length of their practice. Evaluating function for this quantitative criterion can be defined by a linear fuzzy number with significant values 2, 5, 100, 100. In that case, less than 2 years of practice are not satisfying at all. For the length of practice from 2 to 5 years the satisfaction of the company is growing linearly. More than 5 years of practice is fully satisfactory from the company??s point of view. Values greater than 100 are not supposed to occur.

This way we can define a monotonous evaluating function, which is the most common in the evaluating models, by a fuzzy number.

In the FuzzME software, this process is simplified for the user. It is necessary to chooses just the type of the evaluating function (increasing preference, decreasing preference or preference of a selected value) and set only some of the significant values.

3.3. **Methods of aggregation of partial evaluations.** The calculated partial fuzzy evaluations are then consecutively aggregated according to the structure of the goals tree. With respect to the defined type of the tree node, the fuzzy weighted average method, the ordered fuzzy weighted average method or the fuzzy expert system method is used for the aggregation. Each of the aggregation methods is suitable for a different situation:

The fuzzy weighted average is used if the goal corresponding with the node of interest is fully decomposed into disjunctive goals of the lower level. The normalized fuzzy weights represent uncertain shares of these lower-level goals in the goal corresponding with the considered node.

Again, the ordered fuzzy weighted average requires that the goal corresponding with the given node is decomposed into disjunctive goals of the lower level. In contrast to the fuzzy weighted average, the usage of this aggregation operator supposes special user's requirements concerning the structure of partial fuzzy evaluations. The normalized fuzzy weights again represent uncertain shares of the partial evaluations in the aggregated one. But the normalized fuzzy weights are not linked to the individual partial goals; the correspondence between the weights and the partial evaluations is given by the ordering of partial evaluations of the alternative of interest. It means, evaluations with respect to the same partial goal can have different weights for different alternatives.

If the relationship between the evaluations of the lower level and the evaluation corresponding with the given node is more complex (if neither of the two previous methods can be used), and if expert knowledge about the relationship is available, then the aggregation function is described by a fuzzy rule base of a fuzzy expert system. The approximate reasoning is used to calculate the resulting evaluation. In particular, evaluating function described by a fuzzy expert system is used if the fulfillment of a goal at the end of a tree branch depends on several mutually dependent criteria (i.e., if combinations of criteria values bring synergic or disynergic effects to the resulting multiple-criteria evaluation).

3.3.1. *Aggregation by the fuzzy weighted average method.* If the fuzzy weighted average is used for aggregation of partial fuzzy evaluations, then the uncertain weights of the corresponding partial goals, which express their shares in the superior goal, must be set. To define consistent uncertain weights, a special structure of fuzzy numbers, normalized fuzzy weights, must be used.

In the FuzzME software, both real and fuzzy normalized weights can be used. Normalized real weights, i.e., real numbers $v_1, ..., v_m$, $v_j \geq 0$, $j = 1, ..., m$, $\sum_{j=1}^{m} v_j = 1$, represent a special case of the normalized fuzzy weights.

The fuzzy weighted average of the partial fuzzy evaluations, i.e., of fuzzy numbers $U_1, ..., U_m$ defined on $[0, 1]$, with the normalized fuzzy weights $V_1, \ldots, V_m$, is a fuzzy number U on $[0, 1]$ whose membership function is defined for any $u \in [0, 1]$ as follows

$$U(u) = \max\{\min\{V_1(v_1), ..., V_m(v_m), U_1(u_1), ..., U_m(u_m)\} \mid \sum_{i=1}^{m} v_i u_i = u,\ \sum_{i=1}^{m} v_i = 1,\ v_i, u_i \in [0, 1], i = 1, ..., m\}. \tag{3}$$

For an expert who sets the fuzzy weights, it is not so easy to satisfy the condition of normality. That is why the FuzzME software allows to set only an approximation to the normalized fuzzy weights - fuzzy numbers $W_1, ..., W_m$ on $[0, 1]$ satisfying the following weaker condition

$$\exists w_i \in Ker\, W_i,\ i = 1, ..., n : \sum_{i=1}^{n} w_i = 1. \tag{4}$$

The software removes the potential inconsistence in $W_1, ..., W_m$ and derives the normalized fuzzy weights $V_1, ..., V_m$ from them.

The structure of normalized fuzzy weights and the fuzzy weighted average operation are studied in detail in [9], [11] and [12]. Conditions for verifying normality of fuzzy weights, an algorithm for normalization of fuzzy weights satisfying the condition (4), and an algorithm for calculating fuzzy weighted average, which are all used in the FuzzME software, can be found there. Let us notice, that the used algorithm of fuzzy weighted average calculation is very effective.

3.3.2. *Aggregation by the ordered fuzzy weighted average.* The fuzzy OWA operator is used in case that the evaluator has special requirements concerning the structure of the partial evaluation. For example, he/she does not want any partial goal to be satisfied poorly. Then the weight of the minimum partial evaluation of any alternative equals 1, and the weights of all its other partial evaluations equal 0. The aggregated fuzzy evaluations then represent the guaranteed fuzzy degrees of fulfillment of all the partial goals (the fuzzy MINIMAX method). Another example of the fuzzy OWA operator usage could be the evaluation of subjects who can choose in which of the three areas they will be mostly involved. The evaluation algorithm should take into account their right of choice. Then, e.g., the results in the area where the subject performs best contribute to the overall evaluation by about one half, results from the second area by one third and results from the area in which the subject was least involved contribute to the overall

evaluation only by one sixth. A practical application of such a fuzzy evaluation model could be the overall evaluation of the academic staff with respect to their results in the areas of research, education, and management of education and science.

The ordered fuzzy weighted average represents a fuzzification of the crisp OWA operator by means of the extension principle. Uncertain weights are modeled by normalized fuzzy weights as in the case of fuzzy weighted average.

The following notation will be used to define the ordered fuzzy weighted average: if $(x_1, ..., x_m)$ is a vector of real numbers, then $(x^{(1)}, ..., x^{(m)})$ is a vector in which for all $j \in \{1, \ldots, m\}$, $x^{(j)}$ is the j-th greatest number of $x_1, ..., x_m$.

The ordered fuzzy weighted average of the partial fuzzy evaluations, i.e., of fuzzy numbers $U_1, ..., U_m$ defined on $[0, 1]$, with the normalized fuzzy weights $V_1, \ldots, V_m$, is a fuzzy number U on $[0, 1]$ whose membership function is defined for any $u \in [0, 1]$ as follows

$$U(u) = \max\{\min\{V_1(v_1), ..., V_m(v_m), U_1(u_1), ..., U_m(u_m)\} \mid \sum_{i=1}^{m} v_i u^{(i)} = u,\ \sum_{i=1}^{m} v_i = 1,\ v_i, u_i \in [0, 1], i = 1, ..., m\}. \tag{5}$$

The algorithm used to calculate the ordered fuzzy weighted average in the FuzzME software was taken from [10], where fuzzification of the OWA operator is described in detail. The used algorithm is an analogy to the one used for the fuzzy weighted average.

3.3.3. *Aggregation by the fuzzy expert system.* The fuzzy expert system is used if the relationship between the criteria (or the partial evaluations) and the overall evaluation is complicated. Theoretically, it is possible to model, with an arbitrary precision, any Borel measurable function by means of a fuzzy rule base (properties of Mamdani and Sugeno fuzzy controllers, see e.g. [13]) In reality, the quality of the approximation is limited by the expert's knowledge of the relationship.

If the fuzzy rule base models the relation between values of criteria and the fulfillment of the corresponding partial goal, then the evaluation function is of the following form

$$\begin{aligned} &\text{If } \mathcal{C}_1 \text{ is } \mathcal{A}_{1,1} \text{ and } \ldots \text{ and } \mathcal{C}_m \text{ is } \mathcal{A}_{1,m}, \text{then } \mathcal{E} \text{ is } \mathcal{U}_1 \\ &\text{If } \mathcal{C}_1 \text{ is } \mathcal{A}_{2,1} \text{ and } \ldots \text{ and } \mathcal{C}_m \text{ is } \mathcal{A}_{2,m}, \text{then } \mathcal{E} \text{ is } \mathcal{U}_2 \\ &\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\text{If } \mathcal{C}_1 \text{ is } \mathcal{A}_{n,1} \text{ and } \ldots \text{ and } \mathcal{C}_m \text{ is } \mathcal{A}_{n,m}, \text{then } \mathcal{E} \text{ is } \mathcal{U}_n \end{aligned} \tag{6}$$

where for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, $(\mathcal{C}_j, \mathcal{T}(\mathcal{C}_j), V_j, M_j)$ are linguistic scales representing the criteria, $\mathcal{A}_{i,j} \in \mathcal{T}(\mathcal{C}_j)$ are their linguistic values, $(\mathcal{E}, \mathcal{T}(\mathcal{E}), [0, 1], M_e)$

is a linguistic scale representing the evaluation of alternatives and $\mathcal{U}_i \in \mathcal{T}(\mathcal{E})$ are its linguistic values.

In the FuzzME software, rule bases are defined expertly. The user defines such a rule base by assigning a linguistic evaluation to each possible combination of linguistic values of criteria.

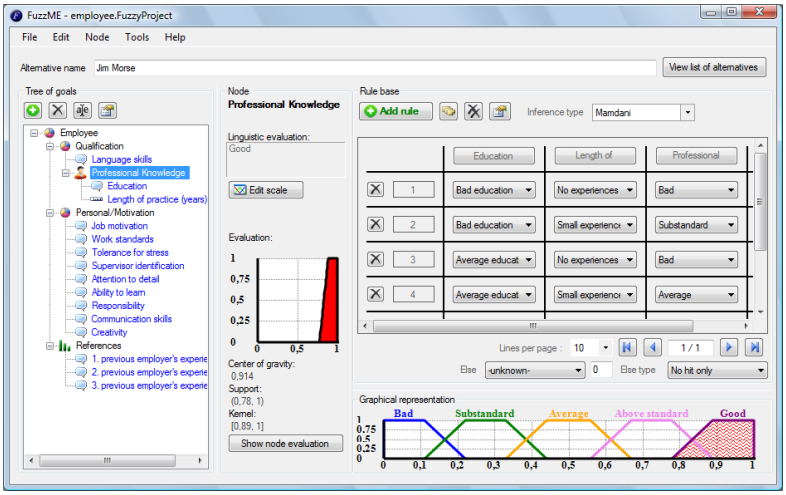

FIGURE 5. Rule base editor

For given values of criteria, a resulting fuzzy evaluation is calculated either by the Mamdani fuzzy inference algorithm, by the Sugeno-WA or the Sugeno-WOWA inference.

In the case of the Mamdani fuzzy inference, the degree h_i of correspondence between the given m-tuple of fuzzy values $(A_1^{'}, A_2^{'}, \ldots, A_m^{'})$ of criteria and the mathematical meaning of the left-hand side of the i-th rule is calculated for any $i = 1, \ldots, n$ in the following way

$$h_i = \min\{hgt(A_1^{'} \cap A_{i,1}), \ldots, hgt(A_m^{'} \cap A_{i,m})\}. \tag{7}$$

Then for each of the rules, the output fuzzy value $U_i^{'}$, $i = 1, \ldots, n$, corresponding to the given input fuzzy values, is calculated as follows

$$\forall y \in [0,1] : U_i^{'}(y) = \min\{h_i, U_i(y)\}. \tag{8}$$

The final fuzzy evaluation of the alternative is given as the union of all the fuzzy evaluations that were calculated for the particular rules in the previous step, i.e.,

$$U^{'} = \bigcup_{i=1}^{n} U_i^{'}. \tag{9}$$

Generally, the result obtained by the Mamdani inference algorithm need not be a fuzzy number. So, for further calculations within the fuzzy model, it must be approximated by a fuzzy number.

The advantage of the generalized Sugeno inference algorithm (see [8]) is that the result is always a fuzzy number. Two version of this algorithm were implemented - Sugeno-WA and, more advanced, Sugeno-WOWA.

In its first step, the degrees of correspondence h_i, $i = 1, \ldots, n$, are calculated in the same way as in the Mamdani fuzzy inference algorithm.

In Sugeno-WA algorithm, the resulting fuzzy evaluation U is then computed as a weighted average of the fuzzy evaluations U_i, $i = 1, 2, \ldots, n$, which model the mathematical meanings of linguistic evaluations on the right-hand sides of the rules, with the weights h_i. This is done by the following formula

$$U = \frac{\sum_{i=1}^{n} h_i.U_i}{\sum_{i=1}^{n} h_i}. \tag{10}$$

The expert chooses values on the right-hand sides of each rule from the linguistic fuzzy scale $(\mathcal{E}, \mathcal{T}(\mathcal{E}), [0,1], M_e)$. We can see that the result can be also obtained as a weighted average of the fuzzy numbers which model the meaning of all values of this scale. Let $E_1, ...E_k$ be those fuzzy numbers, i.e.

$$E_i = M(\mathcal{E}_i), \text{ where } \mathcal{E}_i \in \mathcal{T}(\mathcal{E}), \ i \in \{1, .., k\}. \tag{11}$$

We can assume that those fuzzy numbers are numbered according their ordering from the greatest to the lowest one, i.e., $E_i > E_{i+1}$ for $i \in \{1, ..., k-1\}$

Let $A_1, ..., A_k$ be sets of indices such that A_i contains indices of all rules which have E_i on their right-hand side, i.e.

$$A_i = \{j \in \{1, .., n\} \mid U_j = E_i\}, i = 1, ..., k \text{ where } U_j = M(\mathcal{U}_j). \tag{12}$$

The weights $w'_1, ..., w'_k \in \Re$, which correspond to the values of the linguistic scale $\mathcal{E}$, are calculated, for every i=1,..k, as follows

$$w'_i = \sum_{j \in A_i} h_j \tag{13}$$

and for the further calculations they are normalized:

$$w_i = \frac{w'_i}{\sum_{j=1}^{k} w'_j}. \tag{14}$$

The resulting evaluation of Sugeno-WA inference algorithm can be then expressed as

$$U = \sum_{i=1}^{k} w_i E_i. \tag{15}$$

Sugeno-WOWA algorithm works in the similar way but, instead of weighted average, weighted OWA operator was used. Weighted OWA operator is described in [14]. This operator uses two sets of weights. Weights w_i are the same as in the case of Sugeno-WA. The second set of weights, p_i, is defined by the expert. This gives him/her possibility to specify how important is each value of the scale for the resulting evaluation. Implementation of this inference system was motivated by the real application of this software. A risk rate was calculated by a fuzzy expert system. Expert set significantly greater weight to linguistic value "high risk" than to the value "medium risk". This causes that a single rule that estimated the risk to be high is taken much more seriously than a rule which estimated it to be just medium. So the evaluation algorithm behaves according to the expert's needs because it respects his/her preferences defined by the weights p_i.

The resulting evaluation of Sugeno-WOWA inference algorithm is calculated as follows

$$U = \sum_{i=1}^{k} \omega_i E_i, \tag{16}$$

where the weight ω_i is defined as

$$\omega_i = f(\sum_{j \le i} w_j) - f(\sum_{j<i} w_j), \tag{17}$$

the weights w_i are the same as in Sugeno-WA algorithm and f is a nondecreasing piecewise linear function that is determined by the following points

$$\{(0,0)\} \ \cup \ \{(i/k, \sum_{j \le i} p_j)\}_{i=1,\ldots,k}. \tag{18}$$

In case that the weights p_i are uniform (all scale values have the same weight), the result will be the same as the result calculated by Sugeno-WA. The fact that the values of the scale are ordered simplifies the previous formula. Definition of weighted OWA for more general cases can be found in [14].

3.4. **Overall fuzzy evaluations, the optimum alternative.** The final result of the consecutive aggregation of the partial fuzzy evaluations is an overall fuzzy evaluation of the given alternative. The obtained overall fuzzy evaluations are fuzzy numbers on $[0, 1]$. They express uncertain degrees of fulfillment of the main goal by the particular alternatives.

The FuzzME software compares alternatives according to the centers of gravity of their overall fuzzy evaluations. A center of gravity of a fuzzy number U on $[0, 1]$ that is not a real number, is defined as follows

$$t_U = \frac{\int_0^1 U(x).x \ dx}{\int_0^1 U(x) \ dx}. \tag{19}$$

If $U = u$ and $u \in \Re$, then $t_U = u$. In the FuzzME software, the optimum alternative is the one whose overall fuzzy evaluation has the largest center of gravity.

At present, the FuzzME software is aimed above all at solving multiple-criteria evaluation problems. To ensure high performance in choosing the optimum alternative, it will be necessary to include in the software other methods of ordering of the fuzzy evaluations in the future. Some approaches are proposed in [8] and further research in this area is planned.

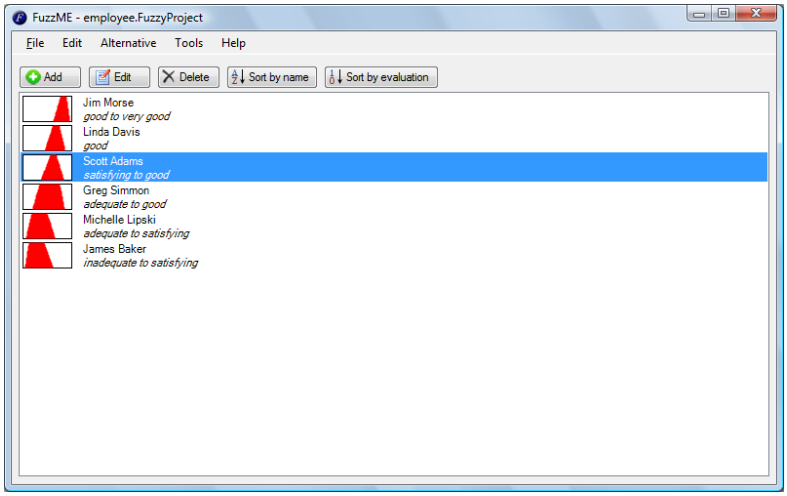

FIGURE 6. A list of alternatives ordered by centers of gravity method

3.5. **Import and export of data.** For fuzzy models of evaluation created in the frame of the program FuzzME, the criteria values of alternatives can be either set directly or imported e.g. from Excel. Similarly resulting evaluations can be exported to the Excel for their further processing.

4. EXAMPLE

The possibilities of this software can be demonstrated on a simple example. Let us consider a company which is going to hire a new employee. There are several candidates and the company naturally wants to select the best of them.

In this example, there are six candidates which are evaluated according to fifteen criteria. Both qualitative and quantitative criteria were used.

For the most of the tree nodes, the fuzzy weighted average was sufficient for the aggregation. One of the exceptions was aggregation of the candidate's references. In this example, it is assumed that the company will try to ask last three of the candidate's previous employers on their experiences with this candidate. The

company is careful and wants the worst of these three evaluations to have the greatest weight. But the other two evaluations should be also taken into account. This can be easily solved by fuzzy OWA operator.

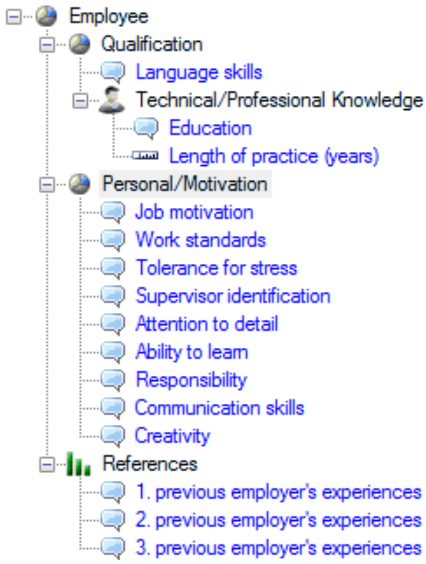

FIGURE 7. The goals tree used in this example

For the evaluation of candidate's technical/professional knowledge a fuzzy expert system was used. This evaluation is obtained from evaluation of candidate's education level and his/her length of practice. Naturally, if the candidate has lots of years of practice then the education level is irrelevant. On the other hand, if the candidate has only small or no practice, the education level should be taken into account. This relationship is too complicated for fuzzy weighted average or fuzzy OWA, but can be easily modeled by a fuzzy rule base.

This simple example shows the advantage over other software products for fuzzy evaluation and decision making. The user has freedom in choosing the aggregation method and they can be arbitrarily combined in the same goals tree.

The FuzzME demo version with this example can be downloaded at `http://FuzzME.wz.cz/`.

5. Conclusion

The FuzzME software makes it possible to create and use fuzzy models of multiple criteria evaluation in the user-friendly way. It has several positive features. The essential one is the solid theoretical basis of the methods contained in the program. The mathematical potential of the software is a result of many years of research. The implemented methods were tested on real problems.

In the FuzzME software, several new methods, algorithms and tools of fuzzy modeling were implemented, e.g.: a structure of normalized fuzzy weights, fuzzy weighted average and ordered fuzzy weighted average operations and algorithms for their calculation and Sugeno-WOWA inference algorithm.

Well-elaborated theoretical basis of the FuzzME software provides a clear interpretation of all steps of the evaluation process and brings understanding of methodology to the user.

References

[1] R.E. Bellman and L.A Zadeh. Decision-making in fuzzy environment. *Management Sci.*, 17 (4):141–164, 1970.

[2] Y. J. Lai and C. L. Hwang. *Multiple Objective Decision Making.* Springer - Verlag, Berlin, Heidelberg, 1994.

[3] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans.Systems Man Cybernet*, 3 (1):183–190, 1988.

[4] H. Rommelfanger. *Fuzzy Decision Support Systeme.* Springer - Verlag, Berlin, Heidelberg, 1988.

[5] INFORM GmbH. FuzzyTECH home page. `http://www.fuzzytech.com/`.

[6] C. Von Altrock. *Fuzzy Logic and NeuroFuzzy Applications in Business and Finance.* Prentice Hall, New Jersey, 1996.

[7] J. Talašová. NEFRIT - multicriteria decision making based on fuzzy approach. *Central European Journal of Operations Research*, 8(4):297–319, 2000.

[8] J. Talašová. *Fuzzy methods of multiple criteria evaluation and decision making (in Czech).* Publishing House of Palacký University, Olomouc, 2003.

[9] O. Pavlačka. *Fuzzy methods of decision making (in Czech).* PhD thesis, Faculty of Science, Palacký Univerzity, Olomouc, 2007.

[10] J. Talašová and I. Bebčáková. Fuzzification of aggregation operators based on Choquet integral. *Aplimat - Journal of Applied Mathematics*, 1(1):463–474, 2008. ISSN 1337-6365.

[11] O. Pavlačka and J. Talašová. Application of the fuzzy weighted average of fuzzy numbers in decision-making models. *New Dimensions in Fuzzy Logic and Related Technologies, Proceedings of the 5th EUSFLAT Conference, Ostrava, Czech Republic, September 11-14 2007, (Eds. M. Štěpnička, V. Novák, U. Bodenhofer)*, II:455–462, 2007.

[12] O. Pavlačka and J. Talašová. The fuzzy weighted average operation in decision making models. *Proceedings of the 24th International Conference Mathematical Methods in Economics, 13th - 15th September 2006, Plzeň (Ed. L. Lukáš)*, pages 419–426, 2006.

[13] B. Kosko. *Fuzzy thinking: The new science of fuzzy logic.* Hyperion, New York, 1993.

[14] V. Torra and Y. Narukawa. *Modeling Decisions.* Springer, Berlin Heidelberg, 2007.

(P. Holeček) Department of Mathematical Analysis and Applications of Mathematics, Faculty of Science, Palacký University Olomouc, Tř. 17. listopadu 1192/12, 771 46 Olomouc, Czech Republic

E-mail address: `holecekp@inf.upol.cz`

(J. Talašová) Department of Mathematical Analysis and Applications of Mathematics, Faculty of Science, Palacký University Olomouc, Tř. 17. listopadu 1192/12, 771 46 Olomouc,, Czech Republic

E-mail address: `talasova@inf.upol.cz`

Acta Universitatis Matthiae Belii ser. Mathematics, 16 (2009), 53–63.
Received: 15 July 2009, Accepted: 2 February 2010.

FISHER INFORMATION AS THE MEASURE OF SIGNAL OPTIMALITY IN OLFACTORY NEURONAL MODELS

ONDŘEJ POKORA

Abstract. Some new approximations of Fisher information are introduced and their properties are derived. These approximations are computed and applied to locate the optimal odorant concentration in two simple theoretical models for coding of odor intensity in olfactory sensory neurons. The results are compared with the deterministic criterion and with results based on Fisher information measure.

1. Introduction

Characterization of the input-output properties of sensory neurons and their models is commonly done by using the so called input-output response functions, $R(s)$, in which the response is plotted against the input s. The output is usually the spiking frequency, or rate of firing, but it can be also concentration of activated receptors as presented e.g. in [7, 8, 9] and also in this contribution. The response curves are usually monotonously increasing functions (most often of sigmoid shape) assigning a unique response to an input signal (see Fig. 1 for illustration).

The intuitive concept of "just noticeable difference", which has been deeply studied in psychophysics, is also implicitly involved in understanding of signal optimality in neurons. Having the transfer function $R(s)$ and minimum detectable increment ϵ of the response, we can calculate Δ_s which is the just noticeable difference in the signal. If the response curve is nonlinear (for example sigmoidal as in Fig. 1) we can see that Δ_s varies along D and the smallest values of the just noticeable difference in the signal are achieved where the response curve is steepest. The stimulus intensity for which the signal is optimal, that is the best detectable, is where the slope of the transfer function is highest.

2000 *Mathematics Subject Classification.* Primary 94A17; Secondary 62P10.
Key words and phrases. Approximations of Fisher information, olfactory neuron, optimal signal.

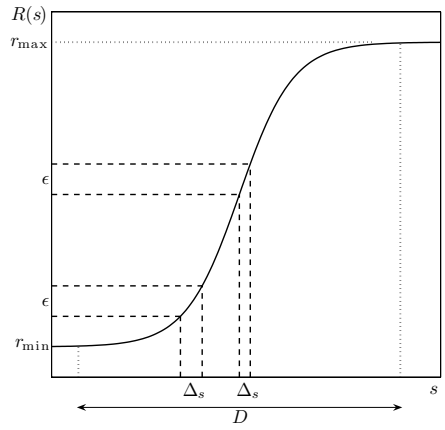

FIGURE 1. A schematic example of transfer function $R(s)$ (solid curve). The dynamic range D, threshold response $r_{\min}$, maximal discharge $r_{\max}$ and just noticeable difference Δ_s in the signal corresponding to the just noticeable difference ϵ in the response are given.

However, in practice, an identical signal does not always yield the same response. The presence of noise complicates the concept of signal optimality based on the just noticeable difference. Not only a fixed response is assigned to every level of the stimulus (as in the classical frequency coding schema), but also a probability distribution of the responses.

In [9], Fisher information was used as a general measure of signal optimality in the case of "noisy response" and applied on theoretical models. The aim of this contribution is to extend a known approximation of Fisher information to a sequence of approximations, apply the same approach on introduced approximations of Fisher information and compare these new optimality measures with known results.

2. Fisher information and its approximation

In this section, some necessary facts about Fisher information measure are recalled. Then, some approximations of Fisher information are introduced and their properties are derived.

2.1. **Fisher information and its properties.** Let us assume, it is dealt with real random variables upon the same probability space $(\Omega, \mathcal{A}, \mathsf{P})$, which have finite second moments and probability density function with respect to some countably additive measure μ. The probability density function $f(x;\theta)$ is assumed to be dependent on a scalar parameter $\theta \in \Theta$.

Regular class. Class of probability density functions $\{f(x;\theta);\theta\in\Theta\}$ is called regular if following conditions hold:

(R1) parametric space Θ is nonempty open set,
(R2) support $M=\{x\in(-\infty,\infty);f(x;\theta)>0\}$ does not depend on θ,
(R3) for almost all $x\in M$ (with respect to μ), finite derivative $\frac{\partial f(x;\theta)}{\partial\theta}$ exists,
(R4) for all $\theta\in\Theta:\int_M\frac{\partial f(x;\theta)}{\partial\theta}\,\mathrm{d}\mu(x)=0$,
(R5) $J^X(\theta)=\int_M\left(\frac{\partial\ln f(x;\theta)}{\partial\theta}\right)^2 f(x;\theta)\mathrm{d}\mu(x)$ holds $0<J^X(\theta)<\infty$.

Regular estimator. Estimator $\hat\theta=H(X)$ of parameter θ in random variable X with p.d.f. $f(x;\theta)$ is called regular if following conditions hold:

(R6) the class $\{f(x;\theta);\theta\in\Theta\}$ is regular,
(R7) $\hat\theta$ is unbiased,
(R8) for all $\theta\in\Theta:\int_M H(x)\frac{\partial f(x;\theta)}{\partial\theta}\,\mathrm{d}\mu(x)=\frac{\partial}{\partial\theta}\int_M H(x)f(x;\theta)\mathrm{d}\mu(x)$.

Fisher information. The value

$$J^X(\theta)=\mathrm{E}\left(\left(\frac{\partial\ln f(X;\theta)}{\partial\theta}\right)^2\right)=\int_M\left(\frac{\partial\ln f(x;\theta)}{\partial\theta}\right)^2 f(x;\theta)\mathrm{d}\mu(x) \tag{1}$$

is called Fisher information about parameter θ in random variable X. Fisher information is not measure of information in the sense of the theory of information (e.g. like entropy). However, it gives how much "information" is transferred into the distribution of X when the parameter θ changes. In other words, it indicates how precisely the change in parameter can be identified (estimated) from the knowledge of the changed distribution. This point of view is induced by following well-known result published in [2].

Cramér-Rao inequality. Let $\hat\theta=H(X)$ be regular estimator of parameter θ with finite second moment. Then, for all $\theta\in\Theta$ following inequality is fulfilled,

$$\frac{1}{J^X(\theta)}\le\mathrm{Var}\left(\hat\theta\right)\quad. \tag{2}$$

Hence, it gives the lower bound for variance of any regular estimator of the parameter. The proof is based on Cauchy-Schwarz inequality for variables $\hat\theta-\mathrm{E}\left(\hat\theta\right)$ and $\frac{\partial\ln f(X;\theta)}{\partial\theta}$.

Assuming we know the best estimator $\hat\theta=H(X)$ of θ in the sense of minimal variance, Cramér-Rao inequality (2) can be seen as relation which gives the quality of estimator $\hat\theta$ as a function of the true value of parameter θ. The idea of analyzing Fisher information $J^X(\theta)$ as a function of θ to find the "optimal" value of θ, i.e. the value for which the best estimator $\hat\theta$ has the lowest variance, was one of the reasons, for which the Fisher information has become a common tool in computational neuroscience (see e.g. [6, 10, 3]).

2.2. **Approximation of Fisher information.** In general, it is difficult task to compute the Fisher information analytically. Usually the integral has to be computed numerically. Moreover, having only measured data without the knowledge of their distribution (which is a typical situation), it is impossible to compute the Fisher information without estimation of the probability density function (e.g. using kernel estimators). These reasons lead to search for some approximation of the Fisher information. Following definition introduce a sequence of such approximations. It is an extension of definition of approximation $J_2^X(\theta)$, which was already used by several authors, see e.g. [6].

Approximations of Fisher information. For $k = 2, 3, \ldots$, let us define sequence of approximations

$$J_k^X(\theta) = \frac{1}{\mathrm{Var}\left(X^{k-1}\right)} \left(\frac{\partial \mathrm{E}\left(X^{k-1}\right)}{\partial \theta} \right)^2 \tag{3}$$

and sequence of conditions

(R9) $\int_M \frac{\partial}{\partial \theta} \left(x^{k-1} f(x;\theta) \right) \mathrm{d}\mu(x) = \frac{\partial}{\partial \theta} \int_M x^{k-1} f(x;\theta) \mathrm{d}\mu(x)$.

Following theorems say that, in general, approximations $J_k^X(\theta)$ are lower bounds for Fisher information $J^X(\theta)$ and that for some special distributions of X there is equality achieved.

Theorem 1. If the class $\{f(x;\theta); \theta \in \Theta\}$ satisfies regularity conditions (R1)–(R5), then, for those $k = 2, 3, \ldots$ for which condition (R9) is satisfied for all $\theta \in \Theta$, there is inequality

$$J_k^X(\theta) \leq J^X(\theta) \qquad \text{for all } \ \theta \in \Theta \ . \tag{4}$$

The principal idea of proof of this inequality uses Cauchy-Schwarz inequality for variables $X^{k-1} - \mathrm{E}\left(X^{k-1}\right)$ and $\frac{\partial \ln f(X;\theta)}{\partial \theta}$.

Theorem 2. Under the same conditions as in Theorem 1 the equality

$$J_k^X(\theta) = J^X(\theta) \qquad \text{for all } \ \theta \in \Theta \tag{5}$$

is fulfilled if and only if the probability density function $f(x;\theta)$ of random variable X has the form

$$f(x;\theta) = \exp\left\{ x^{k-1} c(\theta) - b(\theta) + a(x) \right\} \tag{6}$$

for some functions $a(x), b(\theta), c(\theta)$. The main way of proof follows the idea of sequential equivalent conditions published in [6] for the case $k = 2$. This result might be useful in further work for expressing the accuracy of the approximation in terms of a distance between the real distribution and form (6). The previous condition leads to introducing of following definition.

Exponential class with natural power. Random variable X has a distribution belonging to exponential class with respect to parameter θ and with power

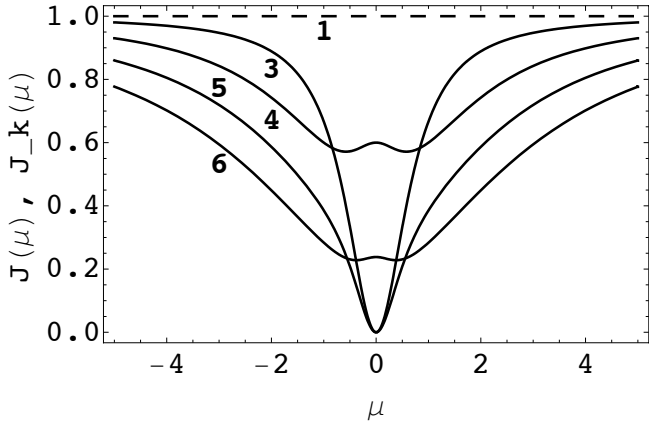

FIGURE 2. Fisher information $J^X(\mu)$ (dashed curve 1) and its approximations $J_k^X(\mu)$ for $k = 3, 4, 5, 6$ (curves 3–6) computed from random variable $X \sim \mathsf{N}(\mu, \sigma^2 = 1)$.

k, $k = 1, 2, \ldots$, if probability density function of X takes the form

$$f(x;\theta) = \exp\left\{x^k c(\theta) - b(\theta) + a(x)\right\} \tag{7}$$

for some functions $a(x), b(\theta), c(\theta)$.

Example. Let us suppose that random variable X has Gaussian distribution $X \sim \mathsf{N}(\mu, \sigma^2)$ with known variance σ^2. Fisher information about the unknown mean value μ, $J^X(\mu) = \frac{1}{\sigma^2}$ does not depend on the true value μ; it means, all values of mean are estimable with equal accuracy, which only depends on the variance. Both Fisher information $J^X(\mu)$ and its approximations $J_k^X(\mu)$ for $k = 3, 4, 5, 6$ are depicted in Fig. 2. Approximation $J_2^X(\mu) = J^X(\mu) = \frac{1}{\sigma^2}$ is accurate. This corresponds with Theorem 2, because Gaussian distribution belongs to the exponential class with respect to parameter μ with power $k = 1$, e.g. for $\sigma^2 = 1$ probability density function is $f(x;\mu) = \exp\left\{x^1\mu - \frac{\mu^2}{2} - \frac{x^2}{2} - \frac{\ln 2\pi}{2}\right\}$.

3. Theoretical models of olfactory neurons

Signal processing in olfactory systems is initialized by binding of odorant molecules to receptor molecules embedded in the membranes of sensory neurons. Binding of odorants and receptor activation trigger a sequence of biochemical events that result in the opening of ionic channels, the generation of receptor potential which triggers a train of action potentials. Studied models of the binding and activation of receptor sites are based on models proposed by [4, 7, 8].

3.1. **Methods.** Searching for "optimal odorant concentration", we aim to investigate how precisely the odorant concentration, s, can be determined from a knowledge of the response, concentration of activated receptors, $C(s)$, and which concentration levels are optimal, that means can be well determined from the knowledge of a random sample of $C(s)$. In other words, we consider an experiment in which a fixed concentration is applied and steady-state responses of the system are observed. These are independent (it is the random sample) realizations of random variable $C(s)$ from which we wish to determine s.

Deterministic approach to determine the optimal concentration is based on shape of the input-output function, $R(s)$, and it uses the optimality criterion

$$J_1(s) = \frac{\partial \mathrm{E}\left(C(s)\right)}{\partial s} \quad . \tag{8}$$

From the stochastic point of view, the determination of the concentration, s, from sampling responses of $C(s)$ corresponds to its estimation, $\hat{s}$, in chosen family of probability density functions. For reasons explained in Section 2, as measures of optimality, the Fisher information (1), $J^X(s)$, is commonly used. Here, we focus on approximations (3), $J_k^X(s)$, and on their application as another optimality criteria in search of optimal odorant concentration in investigated theoretical models.

3.2. **Models and results.** In general, the models consider interaction between odorant molecules and receptors on the surface of olfactory receptor neurons. We assume that there is only one odorant substance, that each receptor molecule possesses only one binding site and that the total number of the receptors on the surface of the membrane is fixed and equal to N. Let A denote the odorant molecules in perireceptor space, with concentration $A = \exp(s)$ which is assumed to be fixed until the olfactory system achieves the steady state. We distinguish three states in which the receptors can appear: unbound (free) state, R, bound inactive state (inactive complex of the odorant molecule and the receptor), C^*, and bound activated state (activated complex of the odorant molecule and the receptor) C. Only activated receptors trigger the response.

Optimality criteria $J_k(s)$ given by (3) are applied on two simple theoretical models of olfactory neurons. The stochastic description of both the models and results of application of $J(s)$, $J_1(s)$ and $J_2(s)$ criteria are already known. Detailed description, derivation of the steady-state (stationary) distribution of number of activated receptors $C(s)$ and results of the criteria can be found in [9]. The results of new criteria are also compared with these previous results.

Basic model.

In the simplest model each occupied receptor becomes activated instantaneously with its occupation. It is assumed that each receptor is occupied and

released independently of others in accordance with stochastic reaction schema

$$\mathrm{A} + \mathrm{R} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{C} \ , \tag{9}$$

where k_1 and k_{-1} are fixed reaction rates coefficients of association and dissociation of the odorant molecules. The ratio $K_1 = k_{-1}/k_1$ is commonly called the dissociation constant. The model can be fully described by birth and death process (see [9] for details). Using this stationary distribution to derive the mean and variance of the count of activated receptors in steady state, $C(s)$, we obtain

$$\mathrm{E}\left(C(s)\right) = \frac{N}{1 + K_1 \mathrm{e}^{-s}} \ , \tag{10}$$

$$\mathrm{Var}\left(C(s)\right) = \frac{N K_1 \mathrm{e}^{-s}}{\left(1 + K_1 \mathrm{e}^{-s}\right)^2} \ , \tag{11}$$

$$\mathrm{E}\left(C^2(s)\right) = \frac{N^2 + N K_1 \mathrm{e}^{-s}}{\left(1 + K_1 \mathrm{e}^{-s}\right)^2} \ , \tag{12}$$

$$\mathrm{E}\left(C^4(s)\right) = \frac{N\left(K_1 \mathrm{e}^{-s}\right)^{N-1}}{\left(1 + K_1 \mathrm{e}^{-s}\right)^N} \ {}_4\mathrm{F}_3\left(2, 2, 2, 1-N; 1, 1, 1; -\frac{\mathrm{e}^s}{K_1}\right) \ , \tag{13}$$

where ${}_p\mathrm{F}_q(a_1, \ldots, a_p; b_1, \ldots, b_q; x)$ stands for generalized hypergeometric function (see [1]). We have ${}_4\mathrm{F}_3(2, 2, 2, 1-N; 1, 1, 1; x) = 1 + \sum_{k=1}^{\infty} \frac{x^k}{k!}(k+1)^3 \prod_{i=1}^{k}(i-N)$.

Assuming the normal distribution of $C(s)$, criteria of optimality $J_1(s)$, $J(s)$ and $J_2(s)$ are directly derived (see[9]),

$$J_2(s) = J_1(s) = \frac{N K_1 \mathrm{e}^{-s}}{\left(1 + K_1 \mathrm{e}^{-s}\right)^2} \ , \tag{14}$$

$$J(s) = \frac{1}{2} + \frac{(N-2) K_1 \mathrm{e}^{-s}}{\left(1 + K_1 \mathrm{e}^{-s}\right)^2} = \frac{1}{2} + \frac{N-2}{N} J_2(s) \ . \tag{15}$$

The new approximations $J_k(s)$ can be computed using relation (3) via higher moments of $C(s)$ and can be expressed in terms of hypergeometric functions.

The shapes of optimality criteria are plotted in Fig. 3. The criteria $J_1(s)$ and $J_2(s)$ are equal and have unimodal shape. For $N > 2$ (which is natural in reality), the Fisher information $J(s)$ is also unimodal and it is very close to $J_1(s)$. As stated in [9], all these criteria attain maximum value $N/4$ for the odorant log--concentration

$$s_0 = \ln K_1 \ . \tag{16}$$

The approximations $J_3(s)$ and $J_4(s)$ has also unimodal shape, but their maxima are slightly shifted from s_0 to higher odorant concentrations. This shift, however, is small and depends only on N (the shift rises with increasing N). For extremely

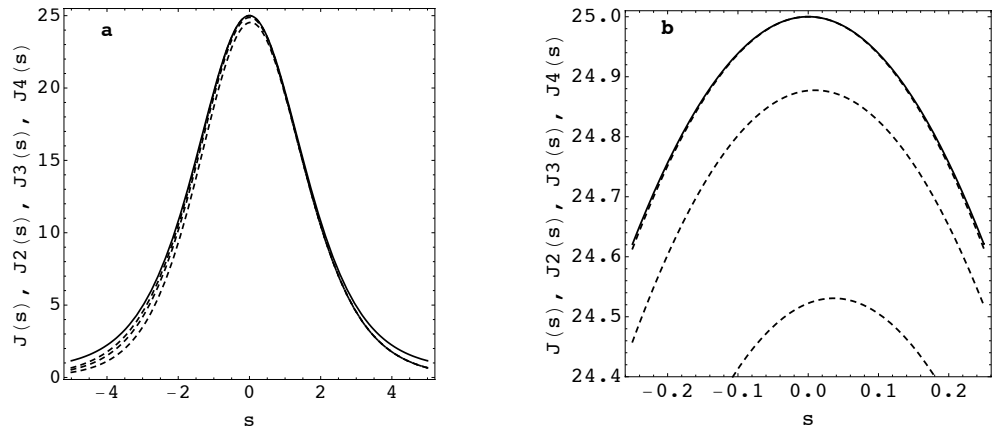

FIGURE 3. (a) Optimality criteria in the basic model: Fisher information $J(s)$, and the criteria $J_1(s) = J_2(s)$, $J_3(s)$ and $J_4(s)$ (dashed curves, up to bottom). Parameters are $K_1 = 1$ and $N = 100$. Criteria J, J_1, J_2 attain maximum value $N/4 = 25$ for the odorant log-concentration $s_0 = \ln K_1 = 0$. (b) Detail of (a); note, that the maxima of J_3 and $J_4(s)$ criteria are slightly biased.

low as well as high odorant concentrations all the criteria decrease. Both the deterministic and Fisher information criteria give the same result and locate the optimal concentration of odorant in the region around the concentration s_0 (see Fig. 3). The criteria based on approximations are slightly biased in positive direction.

Model with simple activation.

Considering the model where not every bound receptor is activated immediately, the receptors really appear in three different states: unbound, R, occupied but not activated, C*, and occupied activated, C. Model described by [5] supposes that each occupied receptor can either become activated, C, with probability $p \in (0, 1)$, or stay inactive, C*, with probability $1 - p$, independently of its past behavior and of the behavior of other receptors. Such an interaction corresponds to the following reaction schema,

$$\mathrm{C}^* \underset{k_{1\mathrm{N}}}{\overset{k_{-1}}{\rightleftharpoons}} \mathrm{A} + \mathrm{R} \underset{k_{-1}}{\overset{k_{1\mathrm{A}}}{\rightleftharpoons}} \mathrm{C} \ , \tag{17}$$

where $k_{1\mathrm{A}} = pk_1$ and $k_{1\mathrm{N}} = (1 - p)k_1$ are association rates for the activated and inactive state and k_1, k_{-1} have the same meaning as in basic model (9).

It can be proved (see [9]) that the steady-state number of activated receptors has binomial distribution $C(s) \sim \mathsf{Bi}(N, q(s))$ with $q(s) = p/(1 + K_1 \mathrm{e}^{-s})$ and its

moments are equal to

$$\mathrm{E}\left(C(s)\right) = \frac{Np}{1+K_1\mathrm{e}^{-s}} \quad , \tag{18}$$

$$\mathrm{Var}\left(C(s)\right) = \frac{NpK_1\mathrm{e}^{-s}}{\left(1+K_1\mathrm{e}^{-s}\right)^2} + \frac{Np(1-p)}{\left(1+K_1\mathrm{e}^{-s}\right)^2} \quad , \tag{19}$$

$$\mathrm{E}\left(C^2(s)\right) = \frac{Np\left(1+p(N-1)+K_1\mathrm{e}^{-s}\right)}{\left(1+K_1\mathrm{e}^{-s}\right)^2} \quad , \tag{20}$$

$$\mathrm{E}\left(C^4(s)\right) = \frac{Np\left(1-\frac{p\,\mathrm{e}^s}{K_1+\mathrm{e}^s}\right)^N {}_4\mathrm{F}_3\left(2,2,2,1-N;1,1,1;\frac{p\,\mathrm{e}^s}{\mathrm{e}^s(p-1)-K_1}\right)}{K_1\mathrm{e}^{-s}-(p-1)} \quad . \tag{21}$$

Criteria J_1 and J_2 are derived analytically,

$$J_1(s) = \frac{pNK_1\mathrm{e}^{-s}}{\left(1+K_1\mathrm{e}^{s}\right)^2} \quad , \tag{22}$$

$$J_2(s) = \frac{pNK_1^2\mathrm{e}^{-s}}{\left(1+K_1\mathrm{e}^{-s}\right)^2\left(K_1+(1-p)\mathrm{e}^{s}\right)} \quad , \tag{23}$$

Fisher information $J(s)$ and its approximations $J_k(s)$ for Gaussian distributed $C(s)$ are evaluated numerically.

As well as in basic model (9), maximum value of the criterion $J_1(s)$ is located at odorant log-concentration $s_1 = \ln K_1$, independently on the value of activation probability p. According to [9], criterion $J_2(s)$ achieves its maximum for the odorant log-concentration

$$s_2 = \ln K_1 - \ln\frac{4(1-p)}{\sqrt{9-8p}-1} \quad . \tag{24}$$

For lower activation probabilities p the location of maximum of $J_2(s)$ is shifted to lower concentrations of odorant.

As shown in Fig. 4, the shape and location of maxima of Fisher information criterion $J(s)$ and approximations $J_2(s)$, $J_3(s)$ and $J_3(s)$ are similar, but different from the maximum of deterministic criterion $J_1(s)$. The deterministic and statistical approaches can give different results, the optimum from statistical point of view is located at lower concentrations of odorant than that obtained with the approach based on the slope of the input-output function. In comparison with Fisher information, the maxima of approximations $J_k(s)$ are slightly biased in positive sense, i.e. locate the optimal signal in higher odorant concentrations than criterion $J(s)$ does. Nevertheless, these maxima are less than the deterministic optimum (as already known for maximum of $J_2(s)$).

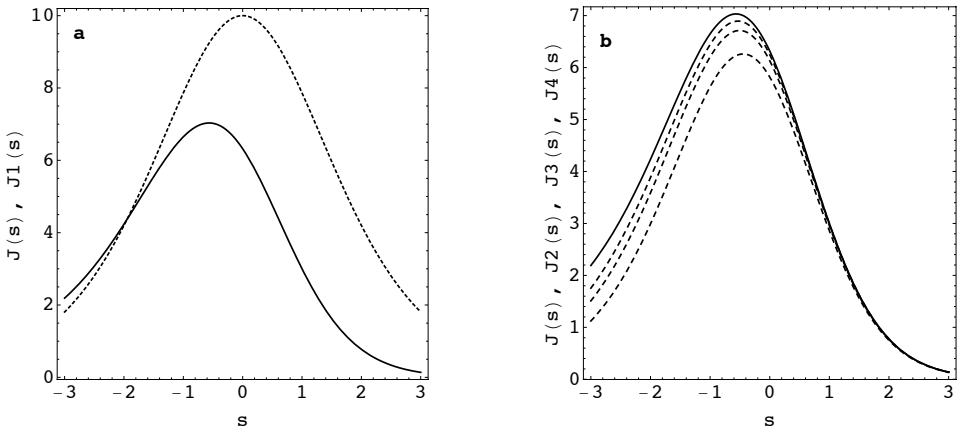

FIGURE 4. Optimality criteria in the model with simple activation: (a) first derivative of the input-output function $J_1(s)$ (dotted curve) and Fisher information $J(s)$ (solid), (b) Fisher information $J(s)$ (solid curve) and its approximations $J_k(s)$ (dashed curves, up to bottom for $k = 2, 3, 4$) as functions of the odorant log-concentration, s, in the perireceptor space. Maximum of $J_1(s)$ is located at $s_1 = 0$. Maximum of $J(s)$ is located at $s \approx -0.565$. Maxima of approximations $J_k(s)$ are shifted to higher concentrations. Parameters are $K_1 = 1$, $N = 100$ and $p = 0.4$.

4. Conclusions

Two theoretical models of theoretical models of olfactory sensory neurons were searched for the optimal signal, s, as defined by the application of approximations $J_k(s)$ of Fisher information $J(s)$. In both models, the approximations $J_k(s)$ have similar shape as Fisher information $J(s)$. In comparison with optimal concentration defined by Fisher information, the maxima of the approximations are biased in positive sense, it means the corresponding odorant concentration determined as optimal are located in higher values. In the model with simple activation, the optimal odorant concentration defined in the sense of approximations $J_k(s)$ is different (less) than the deterministically determined value. Interesting is, that in investigated models the approximations seem to be ordered, even thought there is no clear ordering of functions $J_k(s)$ in general.

Acknowledgement

This work was supported by Grant 201/05/H007 MŠMT ČR and by Center for Theoretical and Applied Statistics LC06024.

References

[1] Abramowitz, M. and Stegun, I. A., *Handbook of Mathematical Functions*, Dover Publications, New York, 1972.

[2] Cramér, H., *Mathematical Methods of Statistics*, Princeton University Press, Princeton, 1946.

[3] Johnson, D. H. and Ray, W., Optimal stimulus coding by neural populations using rate codes, *Journal of Computational Neuroscience* **16** (2004), 129–138.

[4] Kaissling, K. E., Flux detectors vs. concentration detectors: two types of chemoreceptors, *Chemical Senses* **23** (1998), 99–111.

[5] Lánský, P. and Rospars J.-P., Coding of odor intensity, *BioSystems* **31** (1993), 15–38.

[6] Lánský, P. and Greenwood, P. E., Optimal signal in sensory neurons under extended rate coding concept, *BioSystems* **89** (2007), 10–15.

[7] Mankin, R. W. and Mayer, M. S., A phenomenological model of the perceived intensity of single odorants, *Journal of Theoretical Biology* **100** (1983), 123–138.

[8] Maurin, F., A mathematical model of chemoreception for odours and taste, *Journal of Theoretical Biology* **215** (2002), 297–303.

[9] Pokora, O. and Lánský, P., Statistical approach in search for optimal signal in simple olfactory neuronal models, *Mathematical Biosciences* **214** (2008), 100–108.

[10] Sanchez-Montanes, M. A. and Pearce, T. C., Fisher information and optimal odor sensors, *Neurocomputing* **38** (2001), 335–341.

Department of Mathematics and Statistics, Faculty of Science, Masaryk University; Kotlářská 2; 611 37 Brno; Czech Republic

E-mail address: pokora@math.muni.cz

Acta Universitatis Matthiae Belii ser. Mathematics, 16 (2009), 65–79.
Received: 28 June 2009, Accepted: 2 February 2010.

THE USE OF WPF FOR DEVELOPMENT OF INTERACTIVE GEOMETRY SOFTWARE

DAVORKA RADAKOVIĆ AND ĐORĐE HERCEG

Abstract. The Windows Presentation Foundation (WPF) is a graphical subsystem in .NET Framework 3.5, that uses a markup language, called XAML, for rich user interface development. Interactive geometry software (IGS) are computer programs that allow one to create and then manipulate geometric constructions, primarily in plane geometry. Some of the free well known 2D IGS are Geogebra, Cabri and Cinderella. Besides the intended use as a means of teaching and studying geometry, IGS are often used for other purposes, such as development of mathematical games or as a part of other mathematical software (e.g. mathematical drawing viewers). Thanks to its JavaScript interface, GeoGebra is often used in that role and controlled externally by some other software. However, there are some limitations to GeoGebra's usefulness in that respect, since it wasn't developed primarily for that purpose.
Our aim is to offer a solution that can be easily used as a software component for mathematical visualization and interaction. The framework we developed, called "Geometrijica", is simple, straightforward and extensible. It is based on the WPF, which enables it to have a rich graphical appearance and interactivity.
In this paper we demonstrate how our framework, when used together with a mathematical expression evaluator, can be used as a starting point for developing interactive mathematical software.

1. Introduction

Today there are many interactive geometry software (IGS) products available [1], [2], [3]. They are used mostly in teaching and studying geometry, and some more advanced IGS can also graph functions and their derivatives, perform algebraic and symbolic manipulations and so on.

The importance of IGS in today's teaching is widely studied and recognized [9], [11]. For that reason, teachers team up with software developers in order to

2000 *Mathematics Subject Classification.* 68N19.
Key words and phrases. WPF, XAML, IGS, Geometry software, Teaching software, Component development, Mathematical games.

create interactive teaching and learning materials. In order to reach wide audiences, such as elementary school pupils and teachers, the resulting software must be affordable and able to run on various platforms.
GeoGebra is one such IGS, which has gained wide acceptance due to several factors: it is free, runs on all modern operating systems, it is constantly updated and improved, its user interface and user manual have been translated into more than 40 languages, there exist a number of examples and teaching materials freely available on the Internet, and GeoGebra applets can be embedded and used interactively in Web pages [10]. Most importantly, GeoGebra is easy and intuitive to use.

However, the situation is not so simple when it comes to developing new, stand-alone software that should use an existing IGS as a component. First of all, the licensing mode of the IGS in question may not permit such use, and even if it does, there may be some technical limitations or interoperability problems. On the other hand, there are commercial software packages which are more than suitable for such development [4], but the prices for development and runtime versions of these packages may prohibit their widespread use. One of the possible approaches to this problem is to use tools, such as Adobe Flash [5] or OpenLaszlo [6], which are not primarily intended for geometrical applications, for development of mathematical teaching materials, games and examples.

In our previous work, we used GeoGebra to develop course materials [13], [14], primarily because it is a free software and therefore accessible to our target audience. However, we encountered some of GeoGebra's limitations:

- Geometrical shapes in GeoGebra have properties, such as color, line width and shape of points, which can only be changed via the user interface. It would be much more useful if properties of geometrical shapes could get their values from the results of mathematical expressions. That way we could have visual indicators that change their appearance based on the state of the geometrical drawing. For example, an oval representing a set of even numbers could change its color or border width when all the appropriate elements (represented as points) are placed inside its bounds.
- GeoGebra can be controlled from an external program by means of its JavaScript interface. However, this interface, in its current state, provides only the basic functionality. We would like to be able to control every aspect of the geometrical drawing and to react to all the events, such as mouse clicks, keyboard pressed, or object overlapping.
- Properties of objects in GeoGebra are accessed by means of special functions, unlike properties of objects in object-oriented programming languages, which we feel is a more natural way. For example, to obtain the x-coordinate of a point, one needs to type x(A) instead of A.x. This

method is awkward when there are a large number of properties, since each property requires a special function to access it.

- For mathematical game development, we often need to create customized graphical objects, such as coins, fruits, traffic lights, houses, cars etc. While this is possible in GeoGebra, it can be awkward and time consuming. Furthermore, it is not possible to create more than one instance of a customized graphical object needed in any other way but by drawing each instance separately.

For that reason, we decided to develop a new framework, which will solve these problems, while retaining all good aspects of GeoGebra. Since we already had developed a mathematical expression parser and evaluator in C#, we decided to base our framework on it. However, our solution can easily be adopted to use another computer algebra system. Windows Presentation Foundation (WPF) was chosen as the graphical subsystem.

2. Expression evaluator and parser

We developed an expression evaluator and parser, which are based on the same principles as the ones in GeoGebra.

Expression is any simple or complex expression which can be constructed by using constants, variables, arithmetic and logic operations, properties of objects and function calls. Supported functions include common mathematical functions such as power, trigonometry and logical functions. Basically, an Expression is what we are used to seeing in most programming languages like C#. Complex expressions are built by combining simpler expressions using function composition. Besides that, expressions are used to describe geometrical notions, their properties and relations. For example, if M=Segment(A, B) represents a segment between points A and B, then Perpendicular(M, M.Midpoint) represents a line perpendicular to M, passing through its midpoint.

Parser is tasked with accepting textual input and transforming it into expressions. The syntax resembles expression syntax in C#, with arithmetic operations, function calls, and member access. Internally, arithmetic and logic operations and member access are transformed into function calls. For example, the input A=M.X + 3 is transformed into SetVar("A", Plus(MemberOf("M", "X"), 3)). Therefore, all evaluation is actually performed by executing functions.
Expressions have data types. Some common data types are: Number, String, Logical, Color, Point, Segment, Line, Circle and so on. Arguments of functions are checked for data type compatibility at execution. When an expression cannot be evaluated for any reason, it returns a value of the Error data type. Any expression depending on that value also returns a value of the Error data type. Variable is a named expression maintained by the evaluator. A variable consists of an expression and its result.

Variables can depend on other variables. For example, if M=Segment(A, B) is a segment between points A and B, and the coordinates of the point A change, then M must change accordingly. Evaluation of variables is dynamic. As soon as one variable changes its value, all dependent variables are reevaluated. Circular dependencies are not allowed. Therefore, the expressions A=f(B) and B=g(A) are not allowed at the same time. Table 2.1 shows the most important members of the Var class, which is used to keep variables in the evaluator. One important feature of the Var class is that it implements the INotifyPropertyChanged interface, which enables it to notify data consumers of changed values.

Table 1. Members of the Var class

public string **Name**	The name of the variable.
public Expression **Expr**	The expression assigned to the variable.
public Expression **Result**	Result of evaluation of the variable.
public bool **Valid**	Indicates whether the result is valid.
public event PropertyChangedEventHandler **PropertyChanged**	Event from the INotifyPropertyChanged interface, which must be implemented in order to use this class as a data source.

Evaluator is a computational engine that keeps a set of named expressions and maintains dependencies between them, ensuring that when one expression changes, all dependent expressions get reevaluated. It also maintains the expression set in a consistent state by preventing creation of circular dependencies and by deleting all dependent expressions of a deleted expression.

3. Design goals

The requirements placed before the Geometrijica framework are the following:

- Mathematical notions that can be drawn on paper, such as points, lines, circles and graphs of functions, can be shown on screen. For example, by defining a variable M=Segment(A, B) we are also creating a graphical representation of the segment M, which is drawn on screen. When the value of the variable changes, the image on the screen also changes.
- **Geometrical drawing** is a 2D image, consisting mostly of (but not limited to) geometrical shapes, such as points, lines and circles. It is kept in computer memory as a list of visual elements with their respective coordinates and other properties, such as color, size and border width.
- Any property of a visual element can be bound to any variable of the appropriate type. For example, the location of a point in the Cartesian coordinate system can depend on a variable of the type Point. When the value of the variable changes, the position of the visual element is updated on the screen.

- **Visual elements** can represent geometrical shapes, mathematical notions, but they can also be controls, such as buttons, check boxes or sliders. They are implemented by inheriting from WPF controls and user controls, or by inheriting from System.Windows.FrameworkElement. Existing functionality of inherited controls is retained.
- Dependency properties of WPF user controls can be bound to any variable of the appropriate type. Data types are converted by special converter classes. For each pair of types there exists a converter class that provides conversion between them. This enables creation of rich visual representations, which can be controlled by expressions from the evaluator. For example, one can develop a user control that displays a traffic light, with a property that specifies which light is on, and then animate the lights by binding the property to a variable in the evaluator.
- **GeoCanvas** is a WPF control that displays geometrical drawings. GeoCanvas inherits from System.Windows.Controls.Canvas. The part of the 2D plane that is shown inside the GeoCanvas is specified by the coordinates of the lower left and upper right corners. The area displayed in the GeoCanvas can be panned and zoomed.
- GeoCanvas supports both screen coordinate system and geometrical Cartesian coordinate system. Visual elements that are placed on a GeoCanvas decide which coordinate system they will use. Objects using screen coordinates do not move when the geometrical coordinate system moves. This facilitates mixing of user interface elements with the elements of a geometrical drawing.

4. Implementation

4.1. **System overview.** The structure of a program built on the Geometrijica framework is shown in Fig. 4.1. The scope of this discussion is limited to the Visuals, Conversion and Algebra packages, which correspond to appropriate namespaces in Geometrijica.

The Algebra package contains classes discussed in the section "Expression evaluator and parser". A partial list of the classes is shown in Table 4.1.

Table 2. A partial list of classes in the Algebra namespace

Number, String, Logical, EColor	Data types
Neg, Plus, Times	Arithmetic operations
Sqrt, Power, Sin, Cos	Mathematical functions
EPoint, Segment, Line, Circle	Geometrical shapes
Evaluator	Calculation engine
Var	A variable, used in the calculation engine

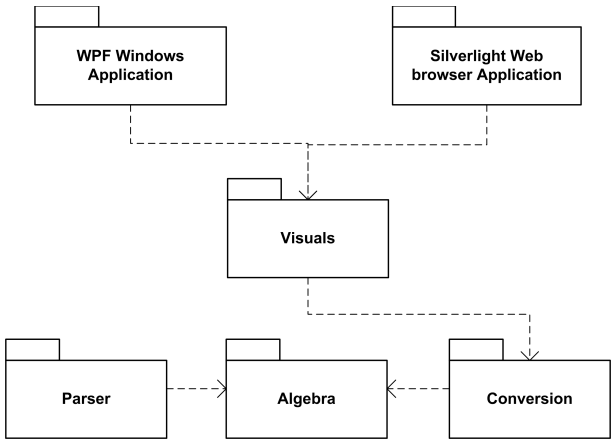

FIGURE 1. Overall structure of a system.

The Visuals package contains the GeoCanvas class, which is a special Canvas control that supports the geometrical coordinate systems, besides the usual pixel-based screen coordinate system. The package also contains classes for graphical representation of geometrical notions, as well as other graphical classes and user interface elements, such as classes derived from WPF controls.
The Conversion package contains converters which perform data type conversions necessary for data binding.

4.2. **Interfaces and enums.** Positioning mode of visual elements is determined by the LocationMode enumeration. The value Screen means that the location of an element is expressed in screen pixels, while the value Geometry means that the location is expressed in geometrical coordinates and that a conversion to screen coordinates is necessary before the element is drawn on screen.

```
public enum LocationMode
{
    Screen, Geometry
}
```

LISTING 1. *LocationMode enumeration*

All visual elements must implement the IElement interface (Table 4.2), which provides basic functionality for element positioning and visibility control. The Valid property is used to control element's visibility based on the validity of the expression it is bound to.

For example, let M=Segment(A, B) be a segment between the points A and B, and P=Perpendicular(M, M.Midpoint) a line perpendicular to M, passing through its midpoint. Suppose that both M and P have their corresponding visual elements shown on screen. Then, if the points A and B are equal, the length of the segment M is zero and the line P cannot exist. In that case, the value of

the variable P in the evaluator will be marked as invalid, and the visual element corresponding to P should not be drawn. This is accomplished by binding the Valid property of the visual element to the Valid property of the variable in the evaluator.

TABLE 3. Members of the IElement interface

GeoCanvas **GeoCanvas**	The GeoCanvas object this element belongs to. Elements use this reference to obtain information about geometrical coordinates in the GeoCanvas.
bool **Valid**	Determines whether the element is valid, i.e. whether it should be drawn.
public Expression **Result**	Result of evaluation of the variable.
bool **Visible**	Controls visibility of the element.
void **CalcScrLocation()**	Calculates and updates the location of the visual element on the screen.

```
private GeoCanvas _geoCanvas;

public GeoCanvas GeoCanvas
{
    get { return _geoCanvas;  }
    set { _geoCanvas = value; }
}

public void CalcScrLocation()
{
    if (GeoCanvas != null)
    {
        switch (LocationMode)
        {
            case LocationMode.Geometry:
                ScrLocation = GeoCanvas.Geo2Scr(Location);
                break;
            case LocationMode.Screen:
                ScrLocation = Location;
                break;
        }
    }
}
(code omitted)
```

LISTING 2. *A typical IElement implementation in a visual element class*

The ILocation interface (Table 4.3) should be implemented by visual elements which can choose the positioning mode between LocationMode.Screen and LocationMode.Geometry. Most visual elements that represent geometrical shapes do not implement this interface. On the other hand, user interface elements such as buttons, check boxes and sliders, which can be placed either on fixed location on screen or bound to geometrical coordinates, implement the ILocation interface.

TABLE 4. Members of the ILocation interface

Point **Location**	Location of the visual element, either in screen or geometry coordinates.
LocationMode **LocationMode**	Specifies how the element's location is interpreted.

4.3. **GeoCanvas.** The GeoCanvas class extends the System.Windows.Controls.Canvas class. It represents a view of a 2D plane in the Cartesian coordinate system. GeoCanvas has four properties, named X0, Y0, X1 and Y1, which determine the region of the 2D plane that is shown on the GeoCanvas.

Visual elements are added to the GeoCanvas by calling the RegisterVisual method.

The Geo2Scr method is used to convert geometrical coordinates into screen coordinates. This method is called by child visual elements, when they are requested by the GeoCanvas to determine their locations. Obviously, only the elements in the 'geometry' positioning mode use this method.

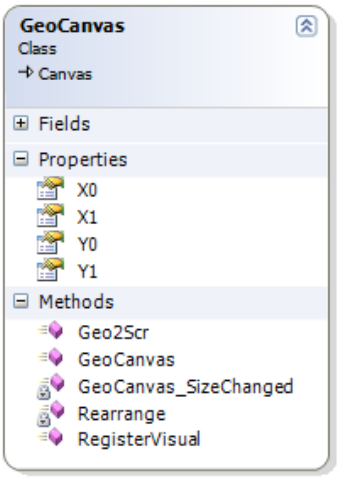

FIGURE 2. The GeoCanvas class.

4.4. **Dependency properties and data binding.** The main idea in our work is to create visual elements which react dynamically to changes in expression values in the evaluator. WPF data binding [7] provides a simple and consistent way of binding elements to data sources, such as databases, XML documents, CLR objects etc.

In our case, data sources are evaluator variables (objects of type Var, Table 2.1). These objects implement the INotifyPropertyChanged interface, which takes care of notifying the WPF infrastructure when a property of a variable

changes. Typical scenario is as follows. When a visual element is created, its properties are bound to the appropriate properties of the corresponding Var object. The visual element is then placed on a GeoCanvas and thus displayed on screen. Each subsequent change of Var object's properties causes the data binding infrastructure to change corresponding properties of the visual element, which is displayed immediately on screen. For this purpose, one-way data binding is used.

Since the data types used in the Evaluator are different from those used in the visual elements, converters must be implemented for each pair of data types for which data binding is meaningful. Listing 3 shows the EPointConverter class, which converts values of type EPoint into values of type Point.

```
[ValueConversion(typeof(EPoint), typeof(Point))]
public class EPointConverter: IValueConverter
{
    public object Convert(object value, Type targetType, object parameter,
                          System.Globalization.CultureInfo culture)
    {
        EPoint ep = value as EPoint;
        if ((ep != null) && (targetType.Equals(typeof(Point))))
        {
            double x1 = ((Number)ep.X).Value;
            double y1 = ((Number)ep.Y).Value;
            return new Point(x1, y1);
        }
        else
        {
            throw new ArgumentException("Invalid type. EPoint expected.", "value");
        }
    }

    public object ConvertBack(object value, Type targetType, object
                        parameter, System.Globalization.CultureInfo culture)
    {  (code omitted)  }
}
```

LISTING 3. *A partial listing of the DoubleConverter class*

Listing 4 shows lines of code that bind the result of the evaluator variable A to the first point of the segment sg. After this code has executed, all changes in the result of the variable A will be immediately reflected on the drawing on screen.

```
sg = new VSegment(new Point(0, 0), new Point(3, 5));
GeoPanel1.RegisterVisual(sg);

Binding bA = new Binding("Result");
bA.Source = Evaluator.Default.Variables["A"];
bA.Converter = new HMS.Geometrijica.Visuals.Conversion.EPointConverter();
sg.SetBinding(VSegment.AProperty, bA);
```

LISTING 4. *Binding of an evaluator variable to a visual element*

4.5. **Visual Elements.** Visual elements are objects that are actually drawn on GeoCanvas. They can be simple dots and lines, or complex drawings. Besides that, common WPF controls, such as buttons, check boxes and sliders can be turned into visual elements and placed on GeoCanvas, while retaining all their functionality. Even complex user controls, with graphical effects and animations can be turned into visual elements and used.

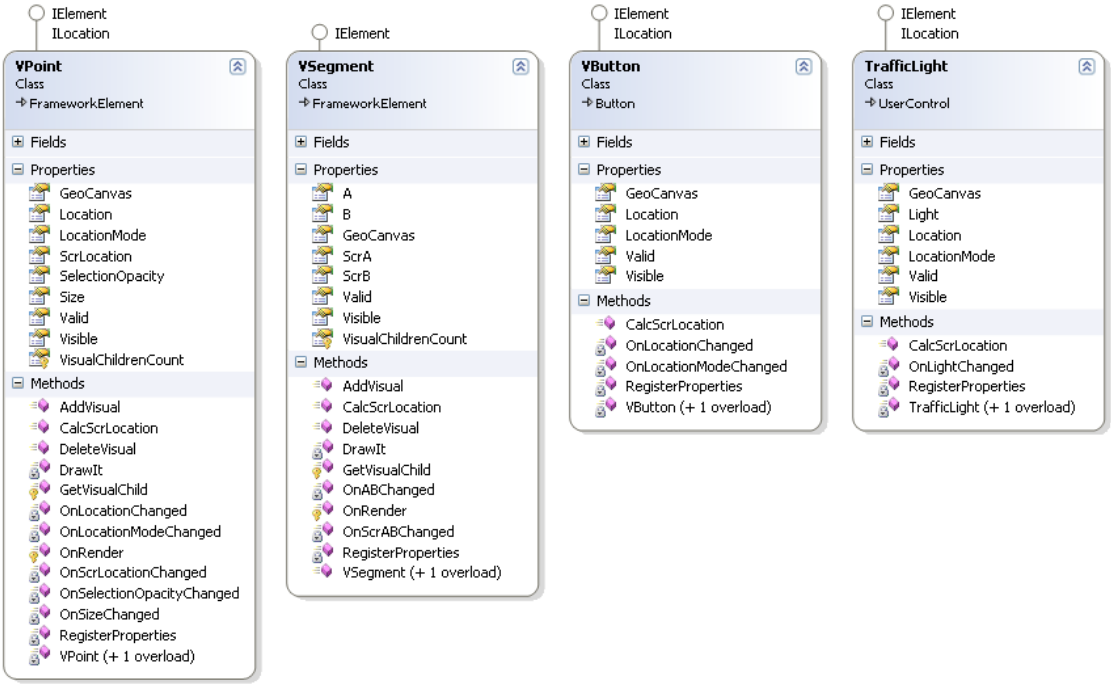

FIGURE 3. Visual elements VPoint, VSegment and VButton.

The process of making a new visual element is different depending on what class is chosen as a starting point. We can start from System.Windows.FrameworkElement and program everything by hand, or we can start from either Control, UserControl or one of the existing WPF controls and implement only a few necessary methods. In either case, the IElement interface must be implemented.

4.5.1. *Creating a new visual element from FrameworkElement.* In this section, the steps necessary to create a visual element from FrameworkElement will be explained. The VPoint class represents a geometrical point and it is drawn as a small circle on screen. The important members of the VPoint class are explained in Table 4.4.

TABLE 5. Important members of the VPoint class

public GeoCanvas **GeoCanvas**	The GeoCanvas object this VPoint belongs to.
private static void **RegisterProperties()**	Registers dependency properties and their corresponding event handlers. Called from the static constructor.
public Point **Location**	Location of the geometrical point, either in screen or geometry coordinates.
public LocationMode **LocationMode**	Specifies how the location property is interpreted.
public double **Size**	Size of this VPoint in pixels.
protected override void **OnRender**(DrawingContext drawingContext)	Called by the WPF when the VPoint needs to be drawn.
private DrawingVisual **DrawIt()**	Performs actual drawing of the VPoint in the specified DrawingContext.
public int **VisualChildrenCount** public void **AddVisual**(Visual v) public int **VisualChildrenCount** public int **VisualChildrenCount**	Required by the FrameworkElement specification. These methods must be implemented in all classes deriving from the System.Windows.FrameworkElement class.

To implement a geometrical point, we start by inheriting the FrameworkElement class. The methods VisualChildrenCount, AddVisual, DeleteVisual and GetVisualChild must be implemented as specified in [8].

We also implement the IElement interface, and the optional ILocation interface. Actual drawing of the point takes place in the DrawIt method, which is called when needed from the overridden OnRender method. The VPoint class has properties that determine its visual appearance. We will consider only the Size property, as implementation details are similar for all other such properties.

```
static VPoint()
{
    RegisterProperties();
}

private static void RegisterProperties()
{
    FrameworkPropertyMetadata mdSize =
        new FrameworkPropertyMetadata(8.0, FrameworkPropertyMetadataOptions.AffectsRender,
                                      new PropertyChangedCallback(OnSizeChanged));
    SizeProperty =
        DependencyProperty.Register("Size", typeof(double), typeof(VPoint), mdSize);

    (code omitted)
}

public static DependencyProperty SizeProperty;

public double Size
{
```

```
    get { return (double)GetValue(SizeProperty); }
    set { SetValue(SizeProperty, value); }
}

private static void OnSizeChanged(DependencyObject obj,
                                  DependencyPropertyChangedEventArgs e)
{
    VPoint t = (VPoint)obj;
    t.DrawIt();
}

private DrawingVisual DrawIt()
{
    DrawingVisual vis = (DrawingVisual)visuals[0];
    using (DrawingContext dc = vis.RenderOpen())
    {
        // selection circle
        Brush sbr = new SolidColorBrush(Colors.Orange);
        sbr.Opacity = SelectionOpacity;
        Pen sp = new Pen(sbr, 5.5);
        dc.DrawEllipse(null, sp, ScrLocation, Size + 4, Size + 4);

        // shape of the point
        Brush br = Brushes.Blue;
        Pen p = new Pen(br, 2.0);
        dc.DrawEllipse(br, p, ScrLocation, Size, Size);
    }
    return vis;
}
```

LISTING 5. *Implementation of the Size property and the DrawIt method*

Figure 4.4 shows the sequence diagram for the RegisterVisual method. When a new visual element (VPoint in this case) is added to the GeoCanvas via the RegisterVisual method, the GeoCanvas control invokes the CalcScrLocation method from the IElement interface. Since the VPoint in question is in the geometry positioning mode, it calls the Geo2Scr method of the GeoCanvas, in order to transform its geometrical coordinates into screen coordinates. After that, the OnRender method is invoked, which, in turn calls the DrawIt method. This method performs the actual drawing of the point at the screen coordinates.

4.5.2. *Creating visual elements from existing controls.* WPF controls already have full functionality, in other words, they know how to draw themselves and to react to user interaction, such as keyboard actions and mouse clicks. It is much easier to create visual elements from existing WPF controls than to code all drawing and behavior logic by hand, as is the case with visual elements based on FrameworkElement. In order to make a visual element from the Button class, we only need to implement the IElement interface in the inheriting class. If we want to be able to place the button on geometrical coordinates, as well as on screen coordinates, we should implement the ILocation interface too. Figure 4.3 shows the VButton class, which was created in the described way.

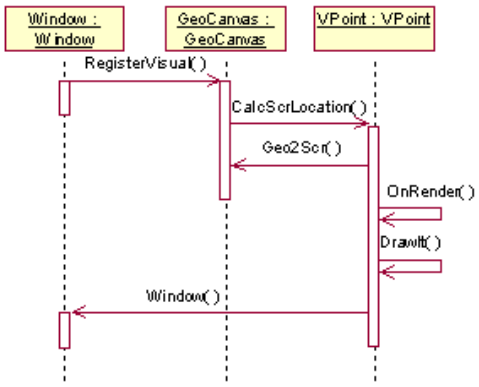

FIGURE 4. Sequence diagram for the RegisterVisual method.

4.5.3. *Creating visual elements from UserControl.* As WPF controls can be specified in XAML, it is also possible to create visual element in that way. Listing 6 shows the specification of a traffic light control with three controllable lights, which can be switched on and off by setting the TrafficLight.Light dependency property. Since the TrafficLight control also implements the IElement interface, it can be placed on GeoCanvas in the same way as all other visual elements, and its appearance can be controlled by a variable from the evaluator (Figure 4.5).

```
<UserControl x:Class="HMS.Geometrijica.Visuals.TrafficLight"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Height="90" Width="30">

    <Border BorderBrush="#FFDA1818" BorderThickness="3,3,3,3">
        <Grid x:Name="LayoutRoot" Background="#FFDDCECE">
            <Grid.RowDefinitions>
                <RowDefinition/>
                <RowDefinition Height="*"/>
                <RowDefinition/>
            </Grid.RowDefinitions>
            <Grid.ColumnDefinitions>
                <ColumnDefinition/>
            </Grid.ColumnDefinitions>
            <Ellipse Fill="#000000" Stroke="#FF000000" Grid.Row="0" />
            <Ellipse Fill="#000000" Stroke="#FF000000" Grid.Row="1" />
            <Ellipse Fill="#000000" Stroke="#FF000000" Grid.Row="2" />
            <Ellipse Fill="#FFFF3304" Stroke="#FF000000" Margin="3 3 3 3" Grid.Row="0"
                  x:Name="RedLight" Opacity="100"/>
            <Ellipse Fill="#FFDFE22A" Stroke="#FF000000" Margin="3 3 3 3" Grid.Row="1"
                  x:Name="YellowLight" Opacity="100"/>
            <Ellipse Fill="#FF1D8B35" Stroke="#FF000000" Margin="3 3 3 3" Grid.Row="2"
                  x:Name="GreenLight" Opacity="100"/>
        </Grid>
    </Border>
```

```
</UserControl>
```

LISTING 6. *Specification of the TrafficLight control in XAML*

One benefit from creating visual elements from existing controls is that these elements retain full functionality of the controls they are based on. This way, we can mix geometrical shapes with WPF controls on a GeoCanvas control. Furthermore, dependency properties of those controls can be bound to results of arbitrary expressions in the evaluator (provided that appropriate converters exist).

4.6. **Example.** Figure 4.5 shows a simple window, containing a GeoCanvas control, which in turn contains three TrafficLight controls at screen coordinates, and one point, one segment and one button at geometrical coordinates. When the GeoCanvas is resized, panned and zoomed, the traffic light controls retain their positions, while the other controls' positions move accordingly.

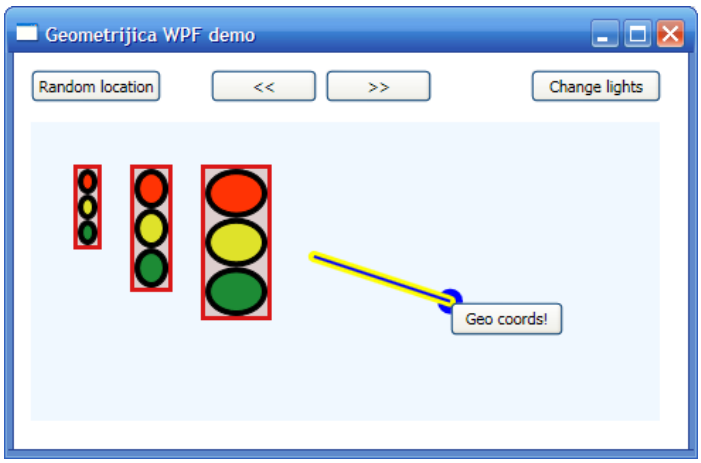

FIGURE 5. Point, segment, button and three traffic light controls on a GeoCanvas.

5. CONCLUSION

Existing interactive geometry software (IGS) are used in teaching of geometry and mathematics in general. GeoGebra is one IGS that has gained wide acceptance thanks to its intuitive use and a great range of features. However, GeoGebra cannot without difficulty be used as a component in other software products. Therefore we developed the 'Geometrijica' framework for geometry software development. A significant part of our framework is the graphical subsystem, which can display geometrical shapes, WPF controls and user controls at the same time.

By using dependency properties and data binding infrastructure in the WPF, we have managed to link the calculation engine with the graphical subsystem, so that all changes in calculation results are reflected in the geometrical drawing. We have also demonstrated that any property of a visual object can be bound to an arbitrary expression in the calculation engine, which is a step further from what GeoGebra offers in this respect. Also we have demonstrated how new visual objects can be made, either by programming them from scratch or by inheriting existing controls. By following a few simple rules, new visual objects can easily be created and used in geometrical drawings. Geometrijica can easily be used as a component in other programs.

References

[1] GeoGebra, `http://www.geogebra.org` (accessed on 8.7.2009)
[2] Cinderella, `http://www.cinderella.de` (accessed on 8.7.2009)
[3] Cabri, `http://www.cabri.com` (accessed on 8.7.2009)
[4] Mathematica, `http://www.wolfram.com` (accessed on 8.7.2009)
[5] Adobe Flash, `http://www.adobe.com/products/flash/` (accessed on 8.7.2009)
[6] OpenLaszlo, `http://www.openlaszlo.org` (accessed on 8.7.2009)
[7] WPF Data Binding - MSDN Library Online, `\emhttp://msdn.microsoft.com/en-us/library/ms750612.aspx` (accessed on 8.7.2009)
[8] MacDonald, M., Pro WPF in C# 2008, Apress, 2008.
[9] Hohenwarter, M., GeoGebra - educational materials and applications for mathematics teaching, PhD thesis, 334 pages, University of Salzburg, Austria, 2006.
[10] Hohenwarter, M. and Preiner, J., *Creating mathlets with open source tools*, Journal of Online Mathematics and its Applications. ID 1574, vol. 7. (2007).
[11] Hohenwarter, J., Hohenwarter, M. and Lavicza Z., *Introducing Dynamic Mathematics Software to Secondary School Teachers: the Case of GeoGebra*, Journal of Computers in Mathematics and Science Teaching (JCMST). 28(2) (2009), 135-146.
[12] Hohenwarter, *GeoGebra: From car design to computer fonts*, M. Informatik-Spektrum. 32(1) (2009), 18-22. Springer.
[13] Herceg, D., Herceg, Đ., *Numerical mathematics with GeoGebra in high school*, Teaching Mathematics and Computer Science, 6/2(2008), 363-378.
[14] Herceg, D., Herceg, Đ., *Scalar product and numerical integration*, Pedago?ka stvarnost, LV, 1-2(2009), 88-102.

Department of Mathematics and Informatics, Faculty of Sciences, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
E-mail address: `davorkar@dmi.uns.ac.rs`

Department of Mathematics and Informatics, Faculty of Sciences, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
E-mail address: `herceg@dmi.uns.ac.rs`

Acta Universitatis Matthiae Belii ser. Mathematics, 16 (2009), 81–89.
Received: 7 July 2009, Accepted: 2 February 2010.

DECOMPOSITION AND PROJECTIVITY OF QUANTALE MODULES

RADEK ŠLESINGER

Abstract. We prove that every quantale module join-generated by its subset of join-irreducible elements can be uniquely decomposed into a collection of further indecomposable submodules. This decomposition actually corresponds to the direct product when the module is "sufficiently distributive". After showing that regular projective indecomposable modules over a given quantale Q are isomorphic to Qd for an idempotent $d \in Q$, we characterize regular projective essential modules that admit this product decomposition as products of such cyclic modules.

Outside the original area of modules over unital rings, the concept of projectivity has also been investigated for sets endowed with an action of a semigroup or a monoid (so called *S-acts*, see [5]), and their partially-ordered variants [13]. Because of similarity of quantale modules to these structures, projectivity suggests to be studied in their categories as well. A recent article [3] presents use of projective objects in study of equivalences of consequence relations on powersets of propositional formulas or sequents, which form unital modules over quantales of sets of substitutions.

In this paper we follow a part of the article [13] where decomposability and projectivity were studied in the category of *S-posets*, i.e., partially ordered sets equipped with an action of a partially ordered monoid that is compatible with the order relation. In accordance with the article, we first develop a little theory of decomposability, and then we apply it to obtain the main result. For the extension to the non-unital case, we make use of the article [2]. For facts on categories of quantales and quantale modules, the reader can refer to [12] and [6].

1. Preliminaries

Our base environment will be the category of *sup-lattices*. Its objects are complete lattices but morphisms include all join-preserving maps. The greatest and

2000 *Mathematics Subject Classification.* 18G05, 06F99.
Key words and phrases. quantale module, projective module, projectivity, decomposition.

the least element of a sup-lattice are denoted by 1 and 0, respectively. A *quantale* stands for a sup-lattice endowed with associative binary multiplication '$\cdot$' distributing over arbitrary joins in both operands. Quantales possessing a multiplicative unit, denoted by e, are called *unital*. Quantale homomorphisms are then sup-lattice homomorphisms preserving multiplication as well.

Given a quantale Q, a *left Q-module* M means a sup-lattice with an associative left action of the quantale $\cdot\colon Q \times M \to M$ that distributes over joins in both components. Throughout this article, 'module' shall stand for a left module. When Q is a unital quantale and $e \cdot m = m$ for all $m \in M$, M is called *unital*, too. As we often consider all multiples of an element m by elements of a set $A \subseteq Q$, we denote by Am the set $\{a \cdot m \mid a \in A\}$, and, by analogy, $AN = \{a \cdot n \mid a \in A,\ n \in N\}$.

A *module homomorphism* f is a sup-lattice homomorphism satisfying $f(q \cdot m) = q \cdot f(m)$ for any $q \in Q$ and $m \in M$. A subset N of M is called a *submodule* if it is nonempty and closed under arbitrary joins and multiplication by elements of Q, while an *ideal* stands for a downward-closed sub-sup-lattice. A *submodule-ideal* will stand for an ideal that is a submodule as well. Submodule-ideals arise as principal downsets $\downarrow m$ given by elements m such that $1_Q \cdot m \leq m$.

When $A \subseteq M$ is closed under quantale action, the submodule of M join-generated by A shall be denoted by $\langle A \rangle$. A Q-module M satisfying $\langle QM \rangle = M$ is called *essential* [9]. The class of essential modules provides the setting for some of the results presented in this paper. In particular, unital modules belong to this class, as well as unital quantales and idempotent quantales, when one regards quantales as modules over themselves.

A nonzero element x of a lattice L is called *join-irreducible* if $x = a \vee b$ implies $x = a$ or $x = b$. To derive our results, we shall deal with modules that are join-generated by their sets of join-irreducible elements. Such lattices have been called *finitely spatial* by F. Wehrung [16]. This class includes, for instance, supercontinuous modules (see [4], Theorem I-3.16, an equivalent statement for completely distributive complete lattices and co-prime, i.e. join-prime elements).

An object P is *regular projective* when for a given regular epimorphism $g\colon A \to B$ any morphism $f\colon P \to B$ can be lifted to $h\colon P \to A$ satisfying $g \circ h = f$. In categories of quantale modules, regular epimorphisms are exactly surjective homomorphisms. As only regular projectivity is discussed in this article, it shall be called projectivity for short.

Two following propositions present well-known properties of projective modules [1, section 4.6].

Proposition 1. *For a Q-module P, the following conditions are equivalent:*

(1) *P is projective.*
(2) *Every epimorphism $f\colon R \to P$ splits, that is, a monomorphism $g\colon P \to R$ exists and satisfies $f \circ g = \mathrm{id}_P$.*

(3) *P is a retract of a free module.*

Proposition 2. *Let $P = \coprod_{i\in I} P_i$ be a coproduct of modules. Then P is projective iff each P_i is projective.*

In categories of sup-lattices and quantale modules, the notions of products and coproducts coincide, so we do not have to distinguish between them.

There exists a free Q-module over any set. Provided that Q is unital, the free module over a set X is Q^X with standard product ordering and componentwise action. For a non-unital quantale Q, the module $(\mathbf{2} \times Q)^X$ plays the role of the free object ($\mathbf{2}$ stands for the two-element quantale in which $1 \cdot 1 = 1$). Action of Q on such a module is given as follows:

$$q \cdot (b, m) = \begin{cases} (0, q \cdot m) & \text{if } b = 0, \\ (0, q \vee q \cdot m) & \text{if } b = 1. \end{cases}$$

This construction was presented in [8], an alternative formulation can be found in [6].

2. Decomposition of Modules

A module M is called *decomposable* if there exist two nontrivial submodule-ideals of M, A and B, $A \cap B = \{0\}$ that generate M as a sup-lattice by joins. Expressed using elements, there exist $a, b \in M$ satisfying $a \wedge b = 0$, $1_Q \cdot a \leq a$, $1_Q \cdot b \leq b$, and for any $m \in M$ it holds that $m = (m \wedge a) \vee (m \wedge b)$. If this does not happen, we say that M is *indecomposable*.

Lemma 3. *Let M be a Q-module and $m \in M$ be a join-irreducible element. Then $N = {\downarrow}\langle Qm \cup \{m\}\rangle = {\downarrow}((1_Q \cdot m) \vee m)$ is an indecomposable submodule-ideal.*

Proof. Obviously, N is an ideal because it is a lower set, and it is also a submodule since the quantale action is order-preserving. Suppose that N is decomposable, that is, there exist A and B, submodule-ideals of N, $A \cap B = \{0\}$ such that $m = a \vee b$ for some $a \in A$, $b \in B$. As m is join-irreducible, we have $m = a$ or $m = b$. Without loss of generality we can suppose $m = a$, thus $m \in A$, $Qm \subseteq A$, and $b = 0$. □

Lemma 4. *Let a module $M = \langle \bigcup_{i\in I} A_i\rangle = \langle \bigcup_{j\in J} B_j\rangle$ for two families $(A_i)_{i\in I}$, $(B_j)_{j\in J}$ of its submodule-ideals. Then for each $i \in I$, the submodule-ideal A_i equals $\langle \bigcup_{j\in J}(A_i \cap B_j)\rangle$.*

Proof. It is evident that $A_i \supseteq \langle \bigcup_{j\in J}(A_i \cap B_j)\rangle$. The converse inclusion also holds: if $a \in A_i$, it is a join of elements $b_k \in A_i$ where each b_k is contained in some B_j since $\bigcup_{j\in J} B_j$ generates the whole M. □

Lemma 5. *Let M_i, $i \in I$, be a family of indecomposable submodule-ideals of a finitely spatial module M satisfying $\bigcap_{i\in I} M_i \neq \{0\}$. Then the submodule $\downarrow\langle\bigcup_{i\in I} M_i\rangle$ is also an indecomposable submodule-ideal.*

Proof. Suppose $\downarrow\langle\bigcup_{i\in I} M_i\rangle$ is decomposable into A and B. Since a non-zero element, which is a supremum of join-irreducibles, belongs to $\bigcap_{i\in I} M_i$, and all M_i are lower sets, there certainly exists a join-irreducible element $m \in \bigcap_{i\in I} M_i$. If m could be written as $a \vee b$ for some $a \in A$ and $b \in B$, it would equal either a, or b. Again, without losing generality, if $m \in A$, $M_i \cap A \neq \{0\}$ for any i. By Lemma 4, $M_i = \langle(M_i \cap A) \cup (M_i \cap B)\rangle$. As all M_i are indecomposable, $M_i \cap B = \{0\}$, $M_i = \langle M_i \cap A\rangle \subseteq A$, therefore $\downarrow\langle\bigcup_{i\in I} M_i\rangle = A$. □

Theorem 6. *Every finitely spatial Q-module can be uniquely decomposed into a collection of its Q-submodule-ideals that are indecomposable and pairwise meeting in 0 only.*

Proof. We already know that $\downarrow\langle Qm \cup \{m\}\rangle$ is indecomposable when m is join-irreducible. Therefore, considering a join-irreducible element x, the set $D_x = \{N \mid x \in N,\ N$ is an indecomposable submodule-ideal$\}$ is nonempty, and $\bigcap_{N\in D_x} N \neq \{0\}$ because it contains x. From Lemma 5 it follows that the set $A_x = \downarrow\langle\bigcup_{N\in D_x} N\rangle$ is an indecomposable submodule-ideal.

Consider two join-irreducible elements $x \neq y$. Then either $A_x \cap A_y = \{0\}$, or $A_x = A_y$. This holds because if there exists $m \in A_x \cap A_y$, $m \neq 0$, also a join-irreducible element $n \leq m$ belongs to the intersection. Obviously $A_x \subseteq \downarrow\langle A_x \cup A_y\rangle$. Using Lemma 5 again, $\downarrow\langle A_x \cup A_y\rangle$ is downward-closed, indecomposable (since $A_x \cap A_y \neq \{0\}$), and containing y, so $\downarrow\langle A_x \cup A_y\rangle \subseteq A_y$ (as A_y includes all indecomposable submodule-ideals containing y). The converse inclusion can be shown in the same way.

We can therefore set an equivalence θ on join-irreducible elements of M as $x\theta y$ iff $A_x = A_y$. Since every element of M is a supremum of join-irreducibles, $M = \langle\bigcup_{x\in C} A_x\rangle$ where C is a suitable set of representatives of classes of θ.

Suppose there exist two such decompositions, so $M = \langle\bigcup_{i\in I} A_i\rangle = \langle\bigcup_{j\in J} B_j\rangle$, and consider one of the B_j. By Lemma 4, $B_j = \langle\bigcup_{i\in I}(A_i \cap B_j)\rangle$. For any $i \in I$ we then obtain a decomposition of B_j as $\langle(A_i \cap B_j) \cup \langle\bigcup_{l\neq i}(A_l \cap B_j)\rangle\rangle$. Let $b \in B_j$ be nonzero. It equals to the supremum of join-irreducibles a_k such that $a_k \leq b$ for all k. Pick any a_k of them and the submodule A_{i_k} that contains a_k. As B_j is indecomposable and $A_{i_k} \cap B_j$ is nontrivial, $\langle\bigcup_{l\neq i_k}(A_l \cap B_j)\rangle = \{0\}$, $B_j = \langle A_{i_k} \cap B_j\rangle$, and thus $B_j \subseteq A_{i_k}$. And vice versa, inclusion of A_{i_k} in some B_m (which is necessarily the considered B_j) can be shown. Identity of the collections A_i, $i \in I$, and B_j, $j \in J$, follows. □

The result can be further improved if we strengthen our assumptions on the order structure of the module. Extending the notion of 0-distributivity [15, section

3.4], we shall call a complete lattice L *infinitely 0-distributive* if L is distributive, and $a \wedge b = 0$ for all $b \in B$ implies $a \wedge \bigvee B = 0$ for any $a \in L$, $B \subseteq L$. As was pointed out by the referee, the class of infinitely 0-distributive modules includes some structures introduced by P. Resende [11]: so-called *quantal frames*, quantales in which binary meets also distribute over arbitrary joins (hence they possess the structure of a frame as well), and *stably supported quantales* where the support ςQ of such a quantale Q is a frame and becomes a left Q-module.

Theorem 7. *Every finitely spatial, infinitely 0-distributive Q-module is isomorphic to the direct product of its Q-submodule-ideals that are indecomposable and pairwise meeting in* 0 *only.*

Proof. We need to show that representation of any element x of M by a join of elements that belong to the compositing submodules is unique. Let M have a decomposition $M = \langle \bigcup_{i\in I} A_i \rangle$ as shown in the previous theorem, and let $x \in M$ satisfy $x = \bigvee\{a_i \mid a_i \in A_i\} = \bigvee\{b_i \mid b_i \in A_i\}$. Then, since the only pairwise-common element of the compositing downward-closed submodules is 0, for each $j \in I$ we have $a_j = a_j \wedge x = a_j \wedge \bigvee_{i\in I} b_i = a_j \wedge \left(\left(\bigvee_{i\neq j} b_i\right) \vee b_j\right) = \left(a_j \wedge \bigvee_{i\neq j} b_i\right) \vee (a_j \wedge b_j) = 0 \vee (a_j \wedge b_j)$, hence $a_j \leq b_j$. Using this argument we can see that the collections of a_i and b_i are equal.

The join map $f\colon \prod_{i\in I} A_i \to M$ given as $f((a_i)) = \bigvee_{i\in I} a_i$ is then a module isomorphism as it is surjective by the assumptions, injective according to the previous paragraph, and it can be verified that it is a homomorphism. □

Example. Consider the lattice $L = Idl(\mathbb{Z}_{60})$ of ideals of the ring $\mathbb{Z}_{60}$. Its subset $\mathrm{JI}(L)$ of join-irreducible elements consists of $a = ([12]_{60})$, $b = ([15]_{60})$, $c = ([20]_{60})$, and $d = ([30]_{60})$. With multiplication of ideals it becomes also a unital quantale, hence a module as well, and the cyclic submodules then look as follows: $La = \{(0), (12)\}$, $Lb = \{(0), (15), (30)\}$, $Lc = \{(0), (20)\}$, $Ld = \{(0), (30)\}$. All of them are subchains connecting 0 and the respective elements, and they are identical to their down-sets. The resulting decomposition then consists of $A_a = La$, $A_b = A_d = Lb$, and $A_c = Lc$.

A different-looking module results from the construction of the endomorphism quantale $\mathcal{Q}(L)$. Multiplication by a quantale element is then performed by endomorphism application. All the elements of L can be achieved from any join-irreducible element this way, hence all cyclic submodules $\mathcal{Q}(L)m$ generated by elements $m \in \mathrm{JI}(L)$ are equal to L, and L is therefore indecomposable as $\mathcal{Q}(L)$-module by Proposition 3.

Now look at the quantale $\mathcal{Q}(L)$. As shown more generally in [14, Proposition 1.7], it is join-generated by its join-irreducible elements of the form

$$f_{ij}(x) = \begin{cases} c_j & \text{if } x \geq c_i, \\ 0 & \text{otherwise,} \end{cases}$$

for c_i and c_j ranging over $\mathrm{JI}(L)$, and so it allows application of our results. Any endomorphism $f \in \mathcal{Q}(L)$ is uniquely determined by setting images of join-irreducible elements. This prescription can be almost arbitrary, it just has to preserve the partial order on $\mathrm{JI}(L)$. As a sup-lattice, $\mathcal{Q}(L)$ is thus isomorphic to $C = \{g\colon \mathrm{JI}(L) \to L \mid g \text{ is monotone}\}$.

The poset $\mathrm{JI}(L)$ can be viewed as a union of disjoint components (with respect to the ordering relation) $L_1 = \{a\}$, $L_2 = \{b, d\}$, $L_3 = \{c\}$. Then each $C_i = \{g\colon \mathrm{JI}(L) \to L | g \text{ is monotone, } g(x) = 0 \text{ for all } x \notin L_i\}$ is also a submodule of $\mathcal{Q}(L)$, and it can be seen that $C_1 \cup C_2 \cup C_3$ join-generates $\mathcal{Q}(L)$. Therefore, $\mathcal{Q}(Idl(\mathbb{Z}_{60}))$ can be decomposed as $Idl(\mathbb{Z}_{60}) \oplus Idl(\mathbb{Z}_{60}) \oplus Idl(\mathbb{Z}_{60})^{\mathbf{2}}$ where $S^{\mathbf{2}}$ means the poset of all monotone maps from $\mathbf{2}$ to a sup-lattice S.

3. Projective Essential Modules

Lemma 8. *Let Q be a quantale and $d \in Q$ be idempotent. Then the module Qd is projective.*

Proof. Let $f\colon Qd \to N$ be a homomorphism and $g\colon M \to N$ be a surjective homomorphism. If $f(d) = n \in N$, there exists $m \in M$ such that $g(m) = n$. Define $h\colon Qd \to M$ as $h(q \cdot d) = (q \cdot d) \cdot m$. This map is a module homomorphism because $h(r \cdot (q \cdot d)) = (r \cdot (q \cdot d)) \cdot m = r \cdot h(q \cdot d)$ and $h\left(\bigvee_{i\in I}(q_i \cdot d)\right) = \left(\bigvee_{i\in I}(q_i \cdot d)\right) \cdot m = \bigvee_{i\in I}(q_i \cdot d \cdot m) = \bigvee_{i\in I}(h(q_i \cdot d))$. The fact of d being idempotent makes the homomorphisms commute: $(g \circ h)(q \cdot d) = g(q \cdot d \cdot m) = q \cdot d \cdot g(m) = q \cdot d \cdot n = q \cdot d \cdot f(d) = f(q \cdot d \cdot d) = f(q \cdot d)$. □

Note that the above Lemma implies that unital quantales are projective when they are regarded as modules.

Proposition 9. *Let M be a Q-module and $m \in M$ belong to Qm. Then the following conditions are equivalent:*

1. *Qm is projective.*
2. *There exists an idempotent $d \in Q$ such that $m = d \cdot m$ and $q \cdot m \mapsto q \cdot d$ is a homomorphism.*
3. *$Qm \cong Qd$ for some idempotent $d \in Q$.*

Proof. 1. $\Rightarrow$ 2. Let Qm be projective. Since $\psi\colon Q \to Qm$ taking q to $q \cdot m$ is onto, by Proposition 1 it is a retraction and there exists a homomorphism $g\colon Qm \to Q$ such that $\psi \circ g = \mathrm{id}_{Qm}$. Let $d \in Q$ denote the g-image of m. Then $m = \psi(g(m)) =$

$\psi(d) = d \cdot m$, and d is idempotent: $d = g(m) = g(d \cdot m) = d \cdot g(m) = d^2$. We can see that g is the desired homomorphism: $g(q \cdot m) = q \cdot g(m) = q \cdot d$.

2. $\Rightarrow$ 3. The image of g which was obtained in the previous step is Qd, and we know that g is injective.

3. $\Rightarrow$ 1. See the previous proposition. □

Proposition 10. *An indecomposable essential Q-module is projective if and only if it is isomorphic to Qd for an idempotent $d \in Q$.*

Proof. The sufficient condition for projectivity is implied by Lemma 8, so suppose that an essential Q-module P is projective and indecomposable.

Let $A = QP$ and for each $a \in A$ fix a pair q_a, r_a such that $a = q_a \cdot r_a$. As P is essential, $P = \langle A \rangle$. For each $a \in A$ we can set a map $f_a \colon Q \to P$ as $f_a(q) = q \cdot r_a$. It can be easily seen that f_a is a module homomorphism containing a in its image.

Using the homomorphisms f_a we can define a map $f \colon \prod_{a \in A} Q \to P$ as $f(x) = f((x_a)) = \bigvee_{a \in A} f_a(x_a)$. Note that $\prod_{a \in A} Q$ is also a coproduct equipped with natural injections ι_a from Q. As for every $a \in A$ $f \circ \iota_a = f_a$, universal property of the coproduct yields that f is a homomorphism. Moreover, f is a surjection — let $p \in P$ be arbitrary, then $p = \bigvee B$ for some $B \subseteq A$. Take the element $y \in \prod_{a \in A} Q$ given as

$$y_a = \begin{cases} q_a & \text{if } a \in B, \\ 0 & \text{if } a \notin B. \end{cases}$$

Then $f(y) = \bigvee_{a \in A} f_a(y_a) = \bigvee_{a \in B} (q_a \cdot r_a) = p$. By Proposition 1 there is an injective homomorphism $g \colon P \to \prod_{a \in A} Q$ satisfying $(f \circ g)(P) = P$. This gives us a submodule $g(P) \subseteq \prod_{a \in A} Q$ isomorphic to P.

Suppose there is an element $0 \neq x \in g(P)$ with x_a and x_b different from 0 for distinct $a, b \in A$. Consider the submodules $R = \{z \in g(P) \mid z_b = 0 \text{ for all } b \neq a\}$ and $S = \{z \in g(P) \mid z_a = 0\}$. By the assumption, these two submodules are nontrivial and downward-closed in $g(P)$, and they join-generate $g(P)$. However, this contradicts indecomposability of $g(P)$. Therefore all non-zero elements of $g(P)$ must be contained in a copy of Q for some $b \in A$.

Hence we obtain that $P = f(g(P)) \subseteq f\left(\prod_{a \in A} Q\right) = P$, thus $f(g(P)) = f_b(Q) = Q \cdot r_b$ for $r_b \in P$. By part 3. of Proposition 9, $P \cong Qd$ for an idempotent $d \in Q$. □

Lemma 11. *The product $M = \prod_{i \in I} M_i$ is essential iff each M_i is essential.*

Proof. If $m = \bigvee_{j \in J} (q_j \cdot n_j)$ for an index set J with $q_j \in Q$ and $n_j \in M$, then obviously every m_i, the i-th component of m, equals $\bigvee_{j \in J} (q_j \cdot (n_j)_i)$.

For the converse, let $m \in \prod_{i \in I} M_i$. For every $i \in I$ there is a set J_i such that $m_i = \bigvee_{j \in J_i} (q_j \cdot n_j)$ for $q_j \in Q$ and $n_j \in M_i$. Then $m = \bigvee_{i \in I, j \in J_i} (q_j \cdot \iota_i(n_j))$ where ι_i denotes the injection into the i-th component of the coproduct. □

Theorem 12. *An infinitely 0-distributive finitely spatial essential Q-module is projective if and only if it is isomorphic to $\prod_{i\in I} Qd_i$ where each d_i is an idempotent element of Q.*

Proof. The 'if' direction follows from the previous proposition and from the fact that products of Q-modules coincide with their coproducts.

For the converse, we have seen that every infinitely 0-distributive finitely spatial Q-module M has a unique decomposition into a set of its indecomposable submodules M_i, $i \in I$, making M isomorphic to their coproduct. As a coproduct of modules is projective iff each one is projective and the same holds true for essentiality, by Proposition 10 each M_i has to be isomorphic to Qd_i for an idempotent $d_i \in Q$. □

The example of decomposition of a quantale/module of sup-lattice endomorphisms on page 85 also illustrates the above result. Since $\mathcal{Q}(L)$ is a unital quantale, all summands can be expressed as cyclic submodules of the quantale which are generated by idempotent homomorphisms. In this case, these homomorphisms are of the form (when prescribed on join-irreducible elements) $f_i(x) = x$ if $x \in L_i$, and 0 otherwise.

Note that if the only idempotents of a unital quantale Q are 0 and the neutral element e, then every projective infinitely 0-distributive finitely spatial unital module over Q is free since its decomposition consists only of copies of Q.

Obviously, finite spatiality and infinite 0-distributivity are not necessary in the 'if' part of the main theorem. In certain cases, they may not be required in the other direction either. An instance of quantales which allow these assumptions on a projective module to be omitted is provided by supercontinuous quantales. Recall that a complete lattice is called supercontinuous if every its element x is a join of elements that lie completely below x, that is, such elements y satisfying $x \leq \bigvee A \implies (\exists a \in A)(y \leq a)$. G. N. Raney [10] proved that supercontinuity equals to complete distributivity, and from [4, Theorem I-3.16] it then follows that supercontinuous lattices are finitely spatial. Complete distributivity also implies infinite 0-distributivity. As products of supercontinuous complete lattices are supercontinuous as well [17] and supercontinuity is preserved by retraction of modules (shown in [7]), projective modules over supercontinuous quantales are supercontinuous, too. The above also implies that finite spatiality and infinite 0-distributivity are satisfied for all projective modules over finite quantales which are distributive as lattices.

Example. Consider the lattice $\mathcal{O}(X)$ of open sets of a topological space $X = [0,1] \cup [2,3]$ with standard topology on reals. With intersection as the binary operation, it becomes a unital idempotent commutative quantale (i.e., a frame), hence a module over itself. Although it is not finitely spatial since it lacks enough join-irreducible elements, it is a decomposable projective module because it is

a unital quantale and it is join-generated by its two submodules $\mathcal{O}([0,1])$ and $\mathcal{O}([2,3])$.

Acknowledgement

The author is grateful to Jan Paseka for providing many valuable comments and directions during preparation of this paper.

References

[1] F. Borceux, *Handbook of Categorical Algebra 1: Basic Category Theory*, Cambridge University Press, Cambridge, 1994.

[2] Y. Q. Chen and K. P. Shum, *Projective and Indecomposable S-acts*, Science in China Series A: Mathematics 42 (1999), pp. 593–599.

[3] N. Galatos and C. Tsinakis, *Equivalence of consequence relations: an order-theoretic and categorical perspective*, Journal of Symbolic Logic 74(3) (2009), pp. 780–810.

[4] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott, *Continuous Lattices and Domains*, Cambridge University Press, Cambridge, 2003, pp. 323–362.

[5] M. Kilp, U. Knauer, and A. V. Mikhalev, *Monoids, Acts and Categories*, Walter de Gruyter, Berlin, 2000.

[6] D. Kruml and J. Paseka, *Algebraic and categorical aspects of quantales*, in Handbook of Algebra, vol. 5, North-Holland, 2008, pp. 323–362.

[7] Y. Li, M. Zhou, and Z. Li, *Projective and injective objects in the category of quantales*, Journal of Pure and Applied Algebra 176 (2002), pp. 249–258.

[8] M. Ordelt, *Moduly v kvantových svazech* (in Czech), Master's thesis, Masaryk University, Brno, 2004.

[9] J. Paseka, *Morita equivalence in the context of Hilbert modules*, in Proceedings of the Ninth Prague Topological Symposium, Charles University and Topology Atlas, Toronto, 2002, pp. 231–258.

[10] G. N. Raney, *A Subdirect-Union Representation for Completely Distributive Complete Lattices*, Proceedings of the American Mathematical Society 4(4) (1953), pp. 518–522.

[11] P. Resende, *Étale groupoids and their quantales*, Advances in Mathematics 208 (2007), pp. 147–209.

[12] K. I. Rosenthal, *Quantales and their applications*, Pitman Research Notes in Mathematics, Longman Scientific & Technical, New York, 1990.

[13] X. Shi, Z. Liu, F. Wang, and S. Bulman-Fleming, *Indecomposable, projective and flat S-posets*, Communications in Algebra, 33 (2005), pp. 235–251.

[14] R. Šlesinger, *Projective Quantales and Quantale Modules*, Master's thesis, Masaryk University, Brno, 2008, available online at `http://is.muni.cz/th/106321/prif_m/`

[15] M. Stern, *Semimodular Lattices: Theory and Applications*, Cambridge University Press, Cambridge, 1999.

[16] F. Wehrung, *Direct decompositions of non-algebraic complete lattices*, Discrete Mathematics 263 (2003), pp. 311–321.

[17] B. Zhao and Y. Zhou, *The category of supercontinuous posets*, Journal of Mathematical Analysis and Applications 320 (2006), pp. 632–641.

Department of Mathematics and Statistics, Masaryk University, Kotlářská 2, 611 37 Brno, Czech Republic

E-mail address: `xslesing@math.muni.cz`