

# DATA SCALING BY DIFFERENTIAL EVOLUTION FOR FCA OVER DATA FROM LMS eLOGIKA

Jakub Kermaschek, Pavla Dráždilová, Marek Menšík

VŠB - Technical University of Ostrava  
Department of Computer Science  
17. listopadu 15, 708 33 Ostrava  
Czech Republic

[jakub.kermaschek@vsb.cz](mailto:jakub.kermaschek@vsb.cz), [pavla.drazdilova@vsb.cz](mailto:pavla.drazdilova@vsb.cz), [marek.mensik@vsb.cz](mailto:marek.mensik@vsb.cz)

*Abstract: The e-learning system eLogika serves for teaching logic. The system collects data about users who are logged in, e.g. time spent on a particular activity, the number of activities performed by particular students, what data is a student interested in, etc. The goal of this paper is to describe the application of many-valued formal concept analysis (FCA) in order to discover typical patterns of students behavior. Since the data stored in the eLogika system are numerical we need to categorize them in order to be used in the many-valued FCA method. In the paper we describe a way of data categorization by differential evolution that proved to be applicable in the FCA method with promising results.*

*Keywords: E-learning, log files, many-valued FCA, data scaling, differential evolution, behavior pattern.*

## 1 Introduction

The amount of data that is obtained by using the logs in the LMS allows us to find behavior patterns of students. Found behavior patterns can be used to improve the quality of learning materials, LMS reorganization, alerting students to potential problems and recommending appropriate approaches within the LMS. One of many possibilities processing of logs in the LMS method is FCA. In the paper [1] authors given a brief example showing the application of the FCA to the domain of computer aided learning, and they discussed comprehensive categorization and classification of the domain knowledge. They indicated their principal ways for utilizing the FCA method to facilitate the exam creation and to integrate it into e-learning systems. Other approach to use FCA in e-learning domain is used in [2]. The online messages are used in the work [3] where the proposed mechanism can automatically construct concept maps based on the messages posted to online discussion forums. By browsing the concept maps, instructors can quickly identify the progress of their students and adjust the pedagogical sequence on the fly. Authors applied unsupervised mapping techniques such as Singular Value Decomposition (SVD) to reduce the dimensionality of the concept space. The objective of the paper [4] is to investigate the knowledge reduction in FCA and propose a method based on Non-Negative Matrix Factorization (NMF) for addressing the issue. The article [5] propose a way of scaling of general attributes to fuzzy attributes. In the paper [12], authors propose a multi-scaled concept lattice. The presented method can effectively reduce the number of concepts, while conserving the main formal structure. A formal context can be converted into an induced context.

We looked for significant patterns of behavior and we did not need to know and visualize the whole structure of conceptual lattice. For this reason, we used just different data scaling by a differential evolution (DE) and looked for an interesting concepts. We used the algorithm described in [11] for fast FCA which are calculated via DE.

## 2 Basic Concepts and Definitions

In this section, we introduce some basic concepts which are used in our approach for data scaling in multi-valued FCA. Figure 2 captures the stages that were required to find concepts - patterns of behavior: preprocessing of log files and creation of multi-valued formal context, categorization of data (subsection 2.2) via a differential evolution. An algorithm In-close is an algorithm for fast computing of formal concepts, whose source code made available [11] which we used for the calculation of the concepts of our categorical data.

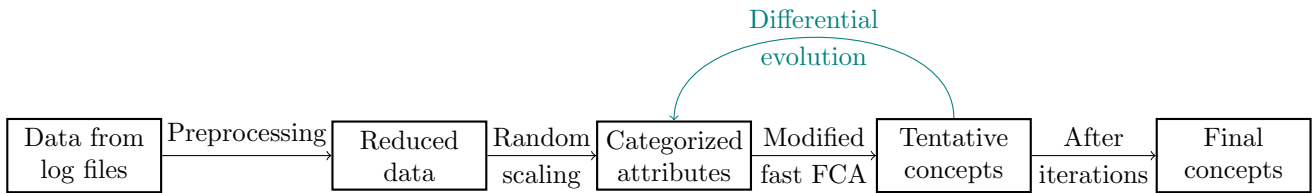


Figure 1: Schema of a proposed approach.

## 2.1 The FCA in Brief

The Formal Concept Analysis (FCA) project originated around 1980 in Darmstadt, Germany. Research group led by R. Wille began to systematically develop a framework for knowledge representation using lattice theory applications. The first publication by R. Wille was presented in 1982 [6], and in 1996 Ganter and Wille published a textbook on the mathematical foundations of FCA, which has been translated into English [7]. The method has been originally proposed as a mathematical model of data analysis. Much of the mathematics required for the application has been borrowed from lattice theory [8]. However, it achieved results that are of much a broader interest than originally intended, and the method is useful in particular as a tool for structured knowledge representation.

Formal concept analysis starts with the primitive notion of a (formal) context.

**Definition.** A formal context is a triple  $[G, M, I]$ , where  $G$  and  $M$  are non-empty sets and is a binary relation  $I$  called incidence relation between  $G$  and  $M$ ,  $I \subseteq G \times M$ . The elements of  $G$  are called objects and the elements of  $M$  are attributes of the context.

We write  $(g, m) \in I$  and say "the object  $g$  has an attribute  $m$ ". Fundamental operators defined at the formal context characterise particular associations (connections) between objects and attributes given by the incidence relation. They are defined as follows:

**Definition.** Given a formal context  $[G, M, I]$  and a subset  $A \subseteq G$  of objects, we define  $A^\uparrow = \{m \in M \mid \forall g \in A (g, m) \in I\}$  and similarly for a subset  $B \subseteq M$  of attributes we define  $B^\downarrow = \{g \in G \mid \forall m \in B (g, m) \in I\}$ .

Operators  $\uparrow, \downarrow$  form the so-called Galois connection between power sets of  $G$  and  $M$ ,  $P(G)$  and  $P(M)$ , respectively, characterized by conditions indicating a natural duality between objects and attributes (see [8]). We can see that  $A^\uparrow$  is the set of all the attributes common to objects of  $A$ , and  $B^\downarrow$  is the set of all the objects possessing attributes from  $B$ . Hence two operators were defined as  $\uparrow: P(G) \rightarrow P(M)$  from objects to the attributes and  $\downarrow: P(M) \rightarrow P(G)$  from attributes to the objects.

**Definition.** A formal concept of the context  $[G, M, I]$  is a pair  $(A, B)$  such that  $A \subseteq G$ ,  $B \subseteq M$ , and  $A^\uparrow = B$ ,  $B^\downarrow = A$ . We say that  $A$  is the extent and  $B$  is the content of the concept  $(A, B)$ . The set of all concepts is denoted by  $\mathcal{B}[G, M, I]$ .

Traditionally, it is taken for granted that the law of inverse proportion between the content and extent holds: increasing the content causes decreasing the extent and vice versa; (extending the demands decreases the number of objects meeting them the more attributes the less objects falling under them, and vice versa). This intuition makes it possible to define a partial ordering on the set of formal concepts following the information content of a formal concept.

## 2.2 Log Data Description

System eLogika, briefly described in [8], stores a lot of data about using the system. The whole system is a web application which means that it is easy to log our students' way of using the system. We log all the data corresponding to particular web pages that the students can see. This means that we have sequences of the accesses to particular URL addresses at our disposal. In this article we deal with the frequency of the access. Our log file consists of the set of n-tuples where each n-tuple contains user (student's) ID, URL address and data-time attributes. Before we could apply data mining methods on our logs, we had to preprocess the data. In our log file we had 269 unique accounts (user IDs) of all the system roles (Administrator, Tutor, Guarantor and Student) and 55 activities (tests and other activities like seminary theses, etc.) in the academic years 2013-2014 and 2014-2015. As a result of data preprocessing we obtained 142 unique students ID and 21 relevant attributes (web pages).

We stored them into the file of vectors consisting of user ID and the frequency of particular web pages access in one academic year. The last attribute (22 nd) in our vectors is attribute of the students evaluation (credits)

in the particular course. This attribute is used for construction of fitness function in the differential evolution and for the evaluation of the results of the analysis.

### 2.3 Data Categorization in Many-valued Formal Context

To use the FCA, it is necessary to convert the attributes in the shape of a formal context where all the values of all attributes are binary, i.e. 1 if the object has the attribute (property) and 0 if not. Values, which become our attributes, are in the range from 0 to 477 and therefore it was necessary to categorize the data. Categorization subsequently determines the creation of formal concepts. The total number of different values which our attributes acquire is 570. In our research, we chose at first a simple categorization One to One (OTO), where each value is categorized into its own category. That means, if the attribute acquires ten different values than there will be ten new attributes (properties). The second type of scaling is realized by DE which constructs categories - new properties of objects. Then we construct by FCA less amount of concepts with smaller variation of students credits in concept.

### 2.4 Differential Evolution

Differential evolution (DE) is used for a global system optimization. DE is a direct parallel optimization method which uses the vectors and objective functions to work on. The vectors represent the particular system settings. An objective function mostly defines the optimization problem as a problem of minimalization. This function is mostly called cost-function or fitness-function. The main problem of all of the direct searching methods is to determine the strategy which generates a variation of the vectors parameters and which goes well through the system. At that moment, the variation is done it is necessary to decide whether to accept the new vector or not. The best method is the greedy method which accepts such as vectors which have the lowest fitness value. Parallelism helps the methods to avoid a local extreme.

For each generation  $G$  which has a population with  $NP$  (population size) vectors, DE processes a population of  $NP$   $D$ -dimensional vectors  $x_{i,G}$ , ( $i = 1, 2, \dots, NP$ ). During the process of minimalization the  $NP$  value stays unchanged. The original vector population is generated by the random way and should cover whole system where the global extreme is looked for. DE generates new parameters by adding the weighted difference of two vectors of the population to the third vector. The mentioned operation is called mutation. This mutated vector is then mixed according to the exact rules with another, preselected fourth (target) vector. We call this operation a crossover, and the newly created vector is called a trial vector. The fitness value of the trial vector is calculated and compared to the target vector. A vector that has a lower fitness value is then chosen to be in a new population. This last operation is called a selection. Each vector of the population serves just once as the target vector, so there are  $NP$  comparisons in one generation and the new population contains  $NP$  vectors.

Process Selection decides whether a target vector  $x_{i,G}$  or trial vector  $u_i$  will be assigned to the new generation  $G+1$ . The target vector is compared to the trial vector with respect to its fitness value. We use greedy selections and a vector with a lower fitness value is selected into a new population.

#### 2.4.1 Differential Evolution - Fitness Function Construction

DE was used to find the best categorization of data with respect to student results. A vector of population represents how properties are to be categorized. Elements of vector determine the categorization of individual attributes. Each element is again a vector that represents the interval of particular category. The element  $v_{i,G} = (x_1, x_2, \dots, x_8)$ , where  $v_{i,G}$  contains min value and  $\max+1$  value. Min represents the least value of an attribute and  $\max$  represents the greatest value of an attribute. A new property of binary formal context (one category from multi-valued attribute) is given by every pair  $x_j, x_{j+1}$  for  $j \in \{1, 2, \dots, 7\}$ . Element's values are ascending in the vector and gain real numbers values. The difference of the elements in the mutation is then performed by standard deduction [10] between the categorization vector elements. Control parameter values are  $CR = 0.8$  (cross rate),  $F = 0.5$  (mutation factor),  $D = 21$  (dimensions),  $NP = 210$  (size of population) and algorithm ends after 5 000 generations.

Due to the different behavior of students who received the 0-point score, these objects were removed from the many-valued formal context. Their very different (border) behaviors cause that the DE algorithm to run badly.

The fitness function determines which individual (i.e. data categorization) is better than the other. To calculate the fitness value, FCA is counted for each individual. A set of all formal concepts is evaluated according to the students' results so that the result is one particular number that DE minimizes. In our paper we used two ways to calculate this figure, called fitness  $\Delta$  ( $\phi_\Delta$ ) and fitness  $\sigma$  ( $\phi_\sigma$ ). For each object of the formal concept, there exists a function  $f$  such that  $f(o_i) = e_i$ .  $O_i$  represents the object (student) of the formal concept and  $e_i$  says what the credit of the course has. A  $\sigma$  value or a  $\Delta$  value is calculated for each formal concept.

$$\sigma = \sqrt{\frac{1}{N} \left( \sum_{i=1}^N e_i^2 \right) - \bar{e}^2}, \quad (1)$$

where  $N$  is the number of all objects in the concept and  $\bar{e}$  is the average evaluation of students (objects) in a such formal concept.

$$\Delta = \frac{\max E_c - \min E_c}{\max E - \min E}, \quad (2)$$

where  $\max E$  and  $\min E$  are constants which are the same for all calculations.  $\max E$  represents the maximum student credit in the data and  $\min E$  represents the minimum student credit in the data.  $\max E_c$  represents the maximum credit in the concept and  $\min E_c$  represents the minimum value of  $e_i$  in the concept. The  $\Delta$  value then represents the coverage ( $\Delta \in \langle 0, 1 \rangle$ ) of the credits in the concept.

Once these values are calculated for all formal concepts, the fitness value of the set of all formal concepts is calculated as

$$\phi_\Delta = \frac{\sum_{i=1}^K \Delta_i}{M}, \phi_\sigma = \frac{\sum_{i=1}^K \sigma_i}{M}, \quad (3)$$

where  $K$  is a number of formal concepts and  $M$  is the number of formal concepts whose  $\Delta$  (or  $\sigma$ ) value is not equal to zero. By this condition of  $M$ , we favor creating concepts having non-zero  $\Delta$  or  $\sigma$ . This is because the zero values usually have concepts containing 1, 2 objects. So we do not want to favor making small concepts. DE is then forced to create concepts with a small value of  $\Delta$  or  $\sigma$ , but not zero.

### 3 Experiments with Data Scaling

The above proposed approach for data scaling allows us to significantly reduce the number of concepts in the formal context created from data in LMS eLogika or the obtained concepts have a required properties of  $\Delta$  and  $\sigma$  which ensure that the concepts are compact.

#### 3.1 Experiments with Fitness Functions

Figure 2 demonstrates a different between fitness function  $\phi_\Delta$  and  $\phi_\sigma$ . Selected attribute  $A_3$  was divided to different categories which then represent a new properties in binary formal context.

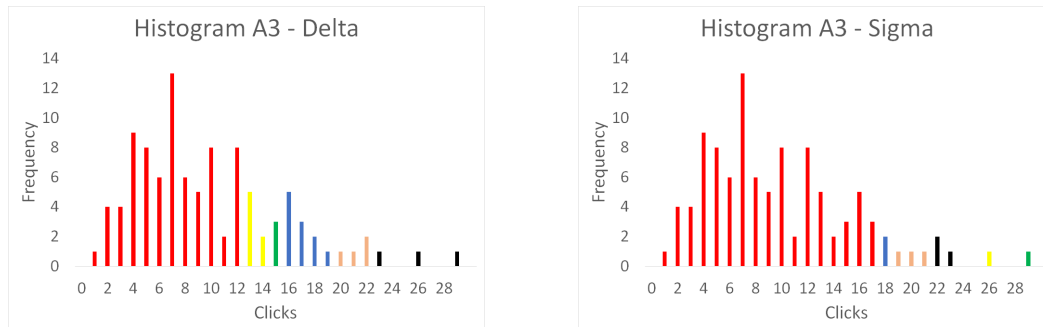


Figure 2: Different scaling of attribute  $A_3$ .

Table 1 contain a short characteristic for obtained formal concepts. The simplest scaling OTO has a lot of concepts and the biggest average value for  $\Delta$  and  $\sigma$ . The scaling by method with a name DER  $\sigma$  has more concepts than OTO but better values of  $\Delta$  and  $\sigma$ . The scaling by method with a name DER  $\Delta$  has the least number of concepts and the smallest values of  $\Delta$  and  $\sigma$ . Table 2 is concentrated on number of formal concepts with a selected value of  $\sigma$ . The simplest method OTO has a lot of concepts with larger  $\sigma$ .

The meaning of DER in the particular methods stands for differential evolution, where the the values of elements in population vectors are in the set of real numbers. As we can see, the mutation creates difference between two vectors of population. Then the result is multiplied by the real number. We can see that there is a possibility to generate non-integer values. The reason why such a feature is enabled is to enable DE to rid of intervals that do not need to create an optimal categorization. For example the optimal categorization of some attribute can be in form  $\langle \min, \max + 1 \rangle$ . So the attribute is not mandatory and should not be split into more than one new property of Binary formal context. If we force DE to generate only integer values that belong to set  $A = \{ \min, \dots, \max + 1 \}$  then a vector that describes the categorization of the given attribute by only

Table 1: Concept’s characterization with different data scaling in multi-valued formal context.

	OTO	DER $\Delta$	DER $\sigma$
Number of formal concepts	4 469	1 120	6 014
Average of $\Delta_i$	0.47	0.26	0.40
Average of $\sigma_i$	14.04	7.73	7.90

one new property should be in form:  $(min, min, \dots, min, max + 1)$  or  $(min, max + 1, max + 1, \dots, max + 1)$ . In any other case, more than one property would be created. We can see that the generation of such a vector (according to the rules we mentioned earlier) is almost impossible. At the beginning of the research, we tried to work with integer values on a subset of natural numbers, but the results were generally worse than using the whole set of real numbers without any other conditions.

Table 2: Number of concepts with different data scaling in multi-valued formal context.

	OTO	DER $\Delta$	DER $\sigma$
Number of formal concepts with $\sigma \leq 10$	1 573 (35%)	879 (78%)	4 753 (79%)
Number of formal concepts with $\sigma \leq 1$	217 (4%)	147 (13%)	108 (2%)
Number of formal concepts with $\sigma = 0$	104 (2%)	96 (8%)	84 (1%)

Table 3 contains all attributes which are divided to some new properties (intervals). New attribute’s border interval values are represented by values in the particular rows in table 3. Other attributes ( $A_4, A_7, \dots, A_{21}$ ) are not splitted to more parts. They stay as one attribute.

Table 3: Scaling of selecting attributes in multi-valued formal context.

Name of attribute								
$A_1$	0	18	20	22	33	53	119	126
$A_2$	0	14	19	27	35	42	374	375
$A_3$	0	13	15	16	20	26	30	
$A_5$	0	2	6	12	20	33	173	174
$A_6$	0	15	28	39	51	78	222	
$A_{10}$	0	3	4	5	6	8	18	
$A_{12}$	0	11	17	33	36	58	72	130
$A_{14}$	0	3	4	12	13	16	40	117

Next part of article contains a short description of the selected results (the biggest concepts with specific parameters) obtained by different data scaling methods and for parameters  $\sigma \leq 10$  and  $\Delta \leq 0.2$ .

- OTO - first method for data scaling in many-valued formal context
  - Concept:  $(\{32, 57, 61, 75, 77, 80, 81\}, \{A_{11}(2, 2), A_{18}(1, 1)\})$ ,
  - $\sigma$ : 4.85,  $\Delta$ : 0.18, average of credits in selected concept: 58.46,
  - extends of direct subcontents:  $\{32, 61, 75, 77, 80, 81\}, \{32, 57, 75, 77, 80, 81\}, \{57, 61, 77, 80\}$ ,
  - extends of direct supercontents:  $\{1, 9, 12, 20, 28, 31, 32, 34, 40, 57, 61, 72, 74, 75, 77, 80, 81\}, \{0, 2, 5, 7, 8, 10, 11, 15, 18, 21, 22, 24, 25, 26, 27, 29, 30, 32, 33, 36, 38, 42, 43, 45, 46, 47, 48, 49, 51, 53, 55, 57, 58, 59, 60, 61, 62, 66, 67, 68, 69, 70, 71, 75, 76, 77, 79, 80, 81, 82, 83, 84, 85, 86, 87, 89, 90, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101\}$ .
- Second method for data scaling - DER  $\sigma$ 
  - Concept:  $(\{21, 23, 31, 35, 37, 40, 41, 44, 45, 52, 55, 60, 62, 63, 64\}, \{A_3(1, 17), A_4(0, 17), A_5(18, 98), A_6(4, 51), A_8(0, 41), A_9(0, 30), A_{10}(0, 17), A_{11}(0, 16), A_{12}(36, 55), A_{13}(0, 26), A_{14}(4, 11), A_{15}(0, 11), A_{16}(0, 24), A_{17}(0, 12), A_{18}(0, 3), A_{19}(0, 10), A_{20}(0, 19), A_{21}(0, 4)\})$ ,
  - $\sigma$ : 4.73,  $\Delta$ : 0.19, average of credits in selected concept: 65.23,

- extends of direct subcontents: {21, 23, 31, 35, 37, 40, 41, 44, 45, 52, 55, 60, 62, 63}, {21, 23, 31, 35, 37, 40, 41, 44, 45, 55, 60, 62, 63, 64}, {21, 23, 35, 37, 40, 41, 44, 45, 52, 55, 60, 62, 63, 64}, {21, 23, 31, 35, 37, 40, 41, 45, 52, 55, 60, 62, 63, 64}, {21, 23, 31, 37, 40, 41, 44, 45, 52, 55, 60, 62, 64}, {21, 23, 31, 35, 37, 41, 44, 45, 52, 55, 60, 63, 64}, {23, 31, 35, 37, 40, 41, 44, 45, 52, 60, 62, 63, 64}, {21, 23, 31, 35, 40, 41, 44, 45, 52, 55, 62, 63, 64}, {31, 35, 40, 44, 52, 55, 60, 64},
- extends of direct supercontents: {0, 3, 7, 13, 21, 23, 31, 35, 37, 40, 41, 44, 45, 52, 55, 60, 62, 63, 64}, {9, 14, 17, 18, 19, 21, 23, 31, 35, 37, 40, 41, 42, 44, 45, 52, 55, 56, 57, 60, 62, 63, 64, 65, 73, 77}, {1, 10, 12, 16, 20, 21, 23, 25, 30, 31, 32, 35, 36, 37, 39, 40, 41, 44, 45, 46, 48, 52, 55, 60, 62, 63, 64, 67, 68, 69, 71, 74, 75, 78, 79, 82, 84, 85}.
- Last method for data scaling - DER  $\Delta$ 
  - Concept: {23, 31, 35, 45, 58, 62, 63, 64, 65}, {A4(0, 17), A5(20, 32), A7(0, 477), A8(0, 41), A9(0, 30), A10(8, 17), A11(0, 16), A13(0, 26), A14(4, 11), A15(0, 43), A16(0, 28), A17(0, 12), A18(0, 7.0), A19(0, 70), A20(0, 22), A21(0, 4)},
  - $\sigma$ : 5.17,  $\Delta$ : 0.18, average of credits in selected concept: 63.71,
  - extends of direct subcontents: {23, 31, 35, 58, 63}, {23, 45, 62, 64, 65}, {23, 45, 58, 63}, {23, 31, 62, 64}, {31, 58, 63, 64}, {23, 35, 45, 62}, {31, 35, 64}, {35, 63, 65},
  - extends of direct supercontents: {23, 31, 35, 39, 45, 58, 62, 63, 64, 65, 67, 82}, {9, 23, 31, 35, 40, 45, 58, 60, 62, 63, 64, 65}, {0, 3, 4, 6, 13, 17, 18, 19, 23, 31, 35, 44, 45, 56, 58, 62, 63, 64, 65}.

## 4 Conclusion

In this paper we showed other possibility how to process the data to get some relevant information. We processed the logs from this LMS and we can analyze students' behavior on the base of frequency of student activities in eLogika. The proposed method of data categorization by a differential evolution allow us to find significant patterns. Obtained patterns are represented as FCA concepts. Our approach construct a smaller amount of formal concepts which contain objects (students) with very similar behavior (similar attributes) in LMS eLogika or the obtained concepts have a smaller values of  $\Delta$  and  $\sigma$  than the results of the OTO method.

**Acknowledgement:** This research support the GACR, project No. GA15-13277S, Hyperintensional logic for natural language analysis, by the grant of VSB–TU Ostrava, project SGS No. SP2017/133, Knowledge modeling and its applications in software engineering III and by SGS No. SP2017/100 Parallel processing of Big Data 4.

## References

- [1] Škopljanač-Mačina, F., Blašković, B.: Formal Concept Analysis–Overview and Applications, *Procedia Engineering*, pp. 1258–1267 (2014).
- [2] Beydoun, G.: Formal concept analysis for an e-learning semantic web. *Expert Systems with Applications*, pp. 10952–10961 (2009).
- [3] Lau, R. Y., Song, D., Li, Y., Cheung, T. C., Hao, J. X.: Toward a fuzzy domain ontology extraction method for adaptive e-learning. *IEEE transactions on knowledge and data engineering*, pp. 800–813 (2009).
- [4] CH, A. K., Dias S. M., Viera, N. J.: Knowledge reduction in formal contexts using non-negative matrix factorization. *Mathematics and Computers in Simulation*, pp. 46–63 (2015).
- [5] Bělohlávek, R., Konečný, J.: Scaling, granulation, and fuzzy attributes in formal concept analysis. *Fuzzy Systems Conference, IEEE International* (2007).
- [6] Wille R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival, editor, *Ordered sets*, Reidel, Dordrecht-Boston, pp. 445–470 (1982).
- [7] Ganter B., Wille R.: *Formal Concept Analysis*. Springer Verlag, (1999).
- [8] Birkhoff G.: *Lattice theory*. American Mathematical Society Colloquium Publications, Providence, RI. (1967).
- [9] Menšík, M., Dráždilová, P.: Analysis of students behavior in LMS eLogika, *SGEM2015 Conference Proceedings, Book 2, Vol. 1*, pp. 27–34 (2015).
- [10] Storn, R., Price, K.: Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11.4 pp. 341–359 (1997).
- [11] Andrews, S.: In-close, a fast algorithm for computing formal concepts. In: *International Conference on Conceptual Structures (ICCS)*, Moscow, (2009).
- [12] Ma, L., Mi, J.S., Xie, B.: Multi-scaled concept lattices based on neighborhood systems. *International Journal of Machine Learning and Cybernetics* (2016).